INSTITUT DE FINANCEMENT DU DÉVELOPPEMENT DU MAGHREB ARABE

معهد تمويل التنمية للمغرب العربي

Mémoire de fin d'Etudes

Thème :

Etude du risque de rachat à l'aide des algorithmes de Machine Learning appliqués à un contrat d'épargne individuel

Présenté et soutenu par :

Amal DAKHLI

Encadré par :

M. Ghazi BEL MUFTI

Etudiant(e) parrainé(e) par :

Assurances MAGHREBIA VIE

November 7, 2022

dédié à mes parents,

dédié à mon mari,

REMERCIEMENTS

Je remercie Monsieur WALID REBHI, responsable de la direction conformité, gestion des risques et actuariat, pour sa disponibilité et pour l'aide qu'il m'a apporté et les indications qu'il m'a fournies pour me guider au mieux dans mon stage.

Je remercie également Monsieur GHAZI BEL MUFTI pour l'aide qu'il m'a apportée lors des différents suivis effectués, et les conseils concernant les missions évoquées dans ce rapport.

Je remercie Madame RAOUDHA BAHRI, responsable de la direction bancassurance pour l'aide qu'elle m'a apporté et pour sa disponiblité durant mon stage.
Je voudrais enfin exprimer ma reconnaissance envers toutes les personnes qui m'ont soutenu et qui m'ont conseillé lors de la rédaction de ce mémoire.

ABSTRACT

Un contrat d'assurance vie est un engagement mutuel de deux parties : l'assuré et l'assureur. Bien que la décision de l'assureur de couvrir le risque soit généralement irrévocable, l'assuré jouit des options qui lui permettent de suspendre, réduire ou résilier la police en récupérant totalement ou partiellement le capital assuré. Ces options légales et contractuelles sont appelées options de rachat et présentent un risque pour l'assureur.
Le risque de rachat est un risque de souscription lié au comportement de l'assuré. Nous distinguons deux types de rachat. Le premier, appelé rachat structurel, est lié au besoin immédiat de liquidité par l'assuré. Le deuxième est le rachat conjoncturel ou dynamique. Ce dernier est induit par le comportement des assurés envers les variations des taux servis par l'assureur par rapport aux taux offerts sur le marché. En assurance vie, le risque de rachat est l'un des risques de souscription les plus conséquents. En effet, en cas d'un flux de rachat massif non prévu ou une mauvaise estimation de ce risque, l'assureur sera confronté à une sortie de liquidité importante.
Dans ce mémoire, nous étudions le rachat structurel par l'approche de machine learning. Nous utilisons dans notre étude des méthodes de machine learning à savoir la régression logistique, l'arbre de décision, la forêt aléatoire et l'algorithme XGBoost.

Mots clés— assurance vie, rachat, machine learning, régression logistique, arbre de décision, forêt aléatoire , XGBoost

Table des matières

Table des figures

Liste des tableaux

INTRODUCTION

L'activité de l'assurance repose sur la notion de risque, du fait de l'inversion du cycle de production. L'assureur ne connaît pas avec précision le montant des sinistres et les frais de gestions occasionnés par ceux-ci. Ainsi, la prestation représente toujours un coût financier aléatoire au début de la période d'assurance (alors que la prime a déjà été payée).
On entend par risque en matière d'assurance tout évènement incertain qui ne dépend pas exclusivement de la volonté des parties et de la survenance de laquelle est subordonnée l'obligation de l'assureur à exécuter la prestation convenue.

En assurance vie, les assureurs font face à divers types de risque lié aux spécificités de leurs produits. Les produits d'assurance vie comme les produits d'épargne, de prévoyance ou les produits décès sont liés à des risques spécifiques à l'assurance vie comme le risque de longévité, le risque de mortalité ou encore le risque de rachat. Nous nous intéressons au risque de rachat d'un contrat d'épargne individuel qui peut trop peser sur la solvabilité des assureurs vie.
Dans un contexte de transformation digitale basée sur l'intelligence artificielle, de nouvelles méthodes statistiques sont apparues offrant de meilleures performances sur plusieurs niveaux ; il s'agit de l'apprentissage automatique (en anglais *machine learning*) ou apprentissage statistique.

Notre objectif est de mettre en place la technique de *machine learning* afin de prédir le comportement des assurés en terme de rachat et maitrisser les variables qui infuencent

ce comportement.

Nous présentons d'abord, dans le premier chapitre, quelques généralités sur l'assurance vie et le risque de rachat en particulier. Ensuite nous exposons dans le deuxième chapitre les algorithmes d'apprentissage automatique qui seront utilisés ultérieurement. Le troisième chapitre traitera l'étape de la préparation et l'analyse de la base de données. Le dernier chapitre est une présentation et analyse des résultats obtenus.

Chapitre 1

ASSURANCE VIE : DÉFINITIONS ET CONTEXTE

Ce chapitre présente les différents concepts clés de l'assurance vie en Tunisie avec un accent particulier sur le rachat. [1]
Il s'attarde sur les généralités de l'assurance vie d'une part et le risque de rachat d'autre part.

1.1 Le contrat d'assurance vie

Le contrat d'assurance vie est régit en Tunisie, par les règles du droit commun des obligations et des contrats, par les dispositions du code des assurances, ainsi que par les arrêtés du ministère des finances et les circulaires du Comité Général des Assurances (CGA).
Le législateur tunisien opte pour une organisation extrêmement réglementée du marché de l'assurance vie, présentant la typologie et les mécanismes de fonctionnement des différents contrats d'assurance vie, définissant les parties prenantes, leurs droits et leurs obligations contractuelles, et dotant le CGA de toutes les prérogatives nécessaires pour mener à bien ses missions de surveillance et de contrôle.
De plus, souhaitant encourager la constitution de l'épargne nationale, il accorde plusieurs avantages fiscaux aux contractants des produits de l'assurance vie.

1.1.1 Les principaux types des contrats

Les assurances en cas de vie

Ce sont des produits visant à constituer une épargne. Ce type de contrat prévoit le versement d'un capital à l'assuré s'il était en vie au terme du contrat. Ils peuvent être assortis d'une contre-assurance permettant aux ayants droit de récupérer les primes payées en cas de décès de l'assuré avant le terme. Cette catégorie regroupe les formules d'assurances suivantes :

— L'assurance de capital différé :

Ce contrat garantit le versement d'un capital à son terme si l'assuré est en vie à cette date. Selon les contrats, les primes pourront donner lieu à des versements périodiques ou un versement unique.

— La rente viagère différée :

En vertu de ces contrats, l'assuré bénéficie de rentes à partir d'une date fixée au contrat et qui cessent avec le décès du rentier contre le paiement d'une prime unique ou de primes périodiques. L'assuré peut également bénéficier de rentes viagères immédiates et auquel cas, elles seront versées dès la prise d'effet du contrat et contre le paiement d'une prime unique.

Les assurances en cas de décès

Ces contrats sont souscrits dans un but de prévoyance. L'assurance décès couvre les conséquences financières du risque de disparition prématurée de l'assuré, elle permet ainsi de couvrir les crédits bancaires, protéger la famille ou encore fournir des rentes scolaires au profit des enfants. L'assureur s'engage alors à verser un capital ou une rente aux bénéficiaires désignés en cas de décès de l'assuré pendant la durée du contrat, ou ses ayants droit. Parmi les contrats d'assurance en cas de décès, on distingue les variantes suivantes :

— L'assurance temporaire décès : Son objet est le versement au bénéficiaire d'un capital ou d'une rente si l'assuré décède pendant la durée du contrat, et ce contre le paiement

d'une prime unique ou de primes périodiques.

— L'assurance vie entière : Elle prévoit le versement d'un capital ou d'une rente prédéterminés au bénéficiaire quelque soit la date du décès de l'assuré contre le paiement d'une prime unique ou de primes périodiques.

Les assurances mixtes

Ces contrats sont en quelque sorte une combinaison des contrats précédemment évoqués. Ils allient à la fois l'objectif épargne et l'objectif prévoyance, c'est-à-dire qu'ils assurent la couverture du risque décès et le versement d'un capital en cas de vie. Cependant, ces deux opérations ne sont pas cumulatives et la compagnie d'assurance ne sera appelée à honorer ses engagements qu'une seule fois. Cette catégorie regroupe :

— L'assurance mixte ordinaire : Un capital sera versé soit au décès de l'assuré si celui-ci intervient avant le terme du contrat, soit à l'échéance si l'assuré est vivant à cette date.

— L'assurance mixte à terme fixe : Dans ce type de contrat, l'assureur devra verser un capital à une date fixée au contrat. Ce capital sera perçu par l'assuré s'il est en vie, ou par les bénéficiaires désignés en cas de décès.

— L'assurance combinée : Il s'agit d'une assurance mixte à terme fixe pour laquelle, les garanties vie et décès sont inégales.

— La vie universelle : Elle consiste à proposer au sein d'un même contrat des prestations financières de type épargne et une garantie décès optionnelle.

Les contrats d'assurance capitalisation

Ce sont de pures produits financiers semblables au mécanisme de l'épargne bancaire dans la mesure où, techniquement, ils ne se basent pas sur les probabilités de décès ou de survie. Ainsi, les primes versées déterminent le capital final. Ces produits peuvent être :

— Libellés en unités monétaires : c'est-à-dire que les garanties du contrat sont des montants monétaires et auquel cas le seul risque supporté par l'assureur sera le risque de taux.

— Libellés en unités de compte : c'est-à-dire que les garanties du contrat sont des valeurs

mobilières (actions, obligations ...) sachant que le risque de perte sera supporté par l'assuré.

1.1.2 Les parties prenantes au contrat d'assurance vie

Quatre parties prenantes au contrat d'assurance vie sont évoquées dans le code des assurances, à savoir :

— **La compagnie d'assurance ou l'assureur**

L'assureur doit être agrée au préalable par l'autorité de tutelle à savoir le Comité Général des Assurances pour pratiquer l'assurance vie.

— **Le souscripteur**

C'est la personne morale ou physique qui s'engage en son nom personnel envers l'assureur, notamment au paiement des primes .

— **L'assuré**

C'est la personne (physique) sur la tête de laquelle l'assurance repose. L'assurance sur la vie peut être contractée sur la tête d'autrui. S'il s'agit d'une assurance en cas de décès, le souscripteur du contrat, aussi bien que le bénéficiaire, peuvent avoir intérêt à la disparition de l'assuré. C'est pour faire face à un tel danger que le législateur a imposé (article 36 du Code des assurances) que l'assuré donne, avant la souscription du contrat, son consentement écrit à l'assurance. Ce consentement doit intervenir avant la souscription du contrat. Il est ainsi admis que l'assuré qui signe le Formulaire de Déclaration du Risque (prévu par l'article 7 alinéas 2 du Code) donne ainsi son consentement à l'assurance. L'absence du consentement requis entraîne la nullité de l'assurance.

— **Le bénéficiaire**

C'est la personne qui, si elle est en vie à cette époque, reçoit les prestations prévues au contrat lors de la réalisation du risque garanti. Le bénéficiaire est désigné par le souscripteur. Cette désignation peut être directe et nominative, mais elle peut tout

aussi bien être indirecte : Le Code prévoit ainsi que sont considérées des personnes désignées : Le conjoint, les descendants nés ou à naître et les héritiers sans indication de leurs noms

1.1.3 Les particularités techniques de l'assurance vie

— **Les provisions mathématiques**

Les provisions mathématiques résultent principalement du phénomène "inversion du cycle de production". Ainsi, le client a en permanence une créance vis-vis de l'assureur qui pour être en mesure d'honorer ses engagements doit constituer des provisions en mettant de côté une bonne partie des primes encaissées. Afin de pouvoir verser à l'échéance le capital promis, l'assureur est tenu de constituer peu à peu des provisions. Ces provisions sont dites mathématiques car elles sont calculées selon des techniques de mathématiques actuarielles Selon le code des assurances, c'est la différence entre les valeurs actuelles des engagements respectivement pris par l'assureur et les souscripteurs de contrats d'assurance vie.

— **La valeur de rachat**

Le rachat est la faculté offerte au souscripteur de mettre fin à son contrat en demandant à l'assureur de lui verser la provision mathématique correspondante. En pratique, tous les contrats ne peuvent être rachetés. C'est le cas par exemple des assurances en cas de décès. Le droit au rachat dont dispose le souscripteur existe car il existe une provision mathématique sur laquelle il a un droit de créance. Parfois, seulement une partie de la provision mathématique est versée comme valeur de rachat ; l'assureur retient en effet une somme, à titre de pénalité de rachat. Les modalités de calcul de la valeur de rachat sont déterminées dans une note technique établie par l'assureur et soumise à l'approbation de l'autorité de contrôle, à savoir le Comité Général des Assurances.

— **La valeur de réduction**

Le contrat d'assurance vie peut faire l'objet d'une réduction en cas de non paiement des primes. Le contrat n'est alors réduit que s'il existe une valeur de rachat suffisante. La réduction consiste en une réduction des garanties de l'assureur. Il est en effet logique que si le souscripteur ne respecte pas ses engagements, l'assureur diminue les siens. Mais le contrat ne peut disparaître totalement puisqu'il existe une provision mathématique. Le souscripteur peut donc prétendre à une assurance équivalente aux droits acquis lors de la cessation des versements.

— **L'avance**

L'avance consiste en une remise par l'assureur au souscripteur d'une partie de sa provision mathématique. Tant que celle-ci n'est pas remboursée, les prestations sont réduites. Ainsi, si l'assureur doit intervenir (réalisation du risque garanti), il déduira des prestations dues le montant de l'avance accordée et non encore remboursée. Par contre, dès que le souscripteur reverse à l'assureur la totalité de l'avance, il retrouve ses droits intacts L'avantage d'une telle formule est évident ; il permet au souscripteur d'obtenir des disponibilités financières sans mettre fin à son contrat. Le montant total de l'avance ne peut excéder la valeur de rachat. Le souscripteur verse à la compagnie un intérêt, contrepartie normale de la perte de revenus supportée par l'assureur du fait de la diminution des actifs de placement.

1.2 La fiscalité en assurance vie

1.2.1 Fiscalité des primes

Les primes d'assurance vie sont déduites du revenu imposable dans la limite de 100 000 dt par an et un minimum d'impôt de 45% et ce, lorsque le contrat obéit à ces conditions :

— L'exécution du contrat dépend de la durée de la vie humaine
— Garantie d'un capital en cas de vie à l'assuré ou à ses descendants pour une durée minimale de 8 ans
— Garantie d'une rente viagère à l'assuré ou à ses descendants avec jouissance différée d'au moins 8 ans
— Garantie d'un capital en cas de décès au profit du conjoint, ascendants ou descendants de l'assuré

1.2.2 Fiscalité des prestations

Toutes les prestations (capital ou rente) sont exonérées de l'IRPP. Toutefois, en cas de rachat pendant les 8 premières années du contrat, les réductions d'impôts doivent être restituées à l'administration fiscale.
Les prestations réglées en cas de décès sont exonérées des droits d'enregistrement sur les successions.

1.3 Le risque de rachat

Un contrat d'assurance vie offre aux assurés une garantie de taux d'intérêt minimum et une participation au bénéfice qui sont liés à la performance de la compagnie d'assurance. Habituellement, des options supplémentaires sont intégrées dans les polices afin de les rendre plus attrayantes pour les assurés. Parmi celles-ci, la plus populaire est l'option de rachat.

Le risque de rachat est avant tout un risque de comportement humain : c'est l'aboutissement d'un processus de décision de l'assuré. Il convient de distinguer deux grandes catégories de rachat : les rachats conjoncturels et les rachats structurels.

1.3.1 Le rachat conjoncturel

Le rachat conjoncturel est lié aux conditions de marché et à la volonté de l'assuré d'investir dans des placements plus rémunérateurs. Au sein des facteurs de risque de rachat d'origine conjoncturelle nous pouvons distinguer les catégories suivantes :

— Le contexte économique et financier : état des marchés financiers, taux de chômage, inflation, croissance etc.
— L'évolution de la législation : apparition de nouvelles taxes
— La concurrence : lancement de nouveaux produits,..
— L'image et le rating de la compagnie

1.3.2 Le rachat structurel

le rachat structurel, correspond à un désir ou besoin de liquidité immédiat de la part de l'assuré. Lors d'un rachat structurel l'assuré rachète pour des raisons personnelles généralement inconnues de l'assureur (achat immobilier, etc.) et cela même si les conditions de marché lui sont favorables. De même, nous pouvons classifier les potentiels facteurs de risques structurels de la façon suivante :

— Les caractéristiques de l'assuré : l'âge, la richesse de l'assuré, son état de santé etc.
— Le type de contrat : contrat d'épargne, de rentes..
— Les caractéristiques du contrat : le montant et le nivellement des primes, la fréquence de versement, les options proposées par le contrat, les pénalités de rachat.

Chapitre 2

LES MÉTHODES DE MACHINE LEARNING : CONCEPTS THÉORIQUES

2.1 Motivations

Né dans les années 1950, le machine learning ou ”apprentissage automatisé ” s'impose de plus en plus dans le monde des données comme un ensemble de méthodes permettant de trouver des solutions à des problèmes statistiques donnés. Cette tendance n'échappe pas à la modélisation du risque de rachat en assurance vie. En effet, nombreux sont les articles de recherche parus récemment qui utilisent le machine learning pour étudier le risque de rachat.

2.1.1 Définition

Le terme "apprentissage automatique" ou "machine Learning " en anglais, désigne essentiellement le processus par lequel les ordinateurs apprennent à partir de données. Sans données, les ordinateurs ne seront pas en mesure d'apprendre quoi que ce soit. Par conséquent, si nous voulons apprendre l'apprentissage automatique, nous devons absolument continuer à interagir avec les données. Toutes les connaissances en matière d'apprentissage automatique impliqueront certainement des données. Les données peuvent être les mêmes, mais les algorithmes et les approches sont différents pour obtenir des résultats optimaux.

2.1.2 Principe du machine learning

De la même manière que le cerveau humain acquiert des connaissances et de la compréhension, le machine learning s'appuie sur des données d'entrée, telles que des données d'entraînement ou des graphes de connaissances, pour comprendre les entités, les domaines et les connexions entre eux. Une fois les entités définies, l'apprentissage profond peut commencer.

Le processus de machine learning commence par des observations ou des données, telles que des exemples, une expérience directe ou des instructions. Il recherche des modèles dans les données afin de pouvoir ensuite faire des déductions sur la base des exemples fournis. L'objectif principal de l'apprentissage automatique est de permettre aux ordinateurs d'apprendre de manière autonome, sans intervention ni assistance humaine, et d'adapter leurs actions en conséquence.

2.1.3 Les différents types d'algorithmes

Il existe de nombreuses méthodes de formation à l'apprentissage automatique parmi lesquelles nous pouvons choisir :

L'apprentissage supervisé

Les algorithmes d'apprentissage automatique supervisé appliquent ce qui a été appris dans le passé à de nouvelles données en utilisant des exemples étiquetés pour prédire des événements futurs. En analysant un ensemble de données d'apprentissage connues, l'algorithme d'apprentissage produit une fonction déduite pour prédire les valeurs de sortie. Le système peut fournir des cibles pour toute nouvelle entrée après un entraînement suffisant. Il peut également comparer sa sortie avec la sortie correcte et prévue pour trouver des erreurs et modifier le modèle en conséquence.

L'apprentissage supervisé peut être divisé en deux sous-catégories : **la classification** et **la régression**.

— **La classification**

Au cours de la formation, un algorithme de classification reçoit des points de données auxquels est attribuée une catégorie. Le travail d'un algorithme de classification consiste alors à prendre une valeur d'entrée et à lui attribuer une classe, ou catégorie, dans laquelle elle s'inscrit sur la base des données d'entraînement fournies. Un exemple classique de classification consiste à déterminer si un courriel est un pourriel ou non. Avec deux classes au choix (spam ou non spam), ce problème est appelé problème de classification binaire. L'algorithme reçoit des données d'entraînement avec des e-mails qui sont à la fois des spams et des non spams. Le modèle trouvera les caractéristiques des données qui correspondent à l'une ou l'autre classe et créera la fonction de correspondance mentionnée précédemment : Y=f(x). Ensuite, lorsqu'il reçoit un e-mail non vu, le modèle utilise cette fonction pour déterminer si l'e-mail est un spam ou non.

Les problèmes de classification peuvent être résolus à l'aide d'un grand nombre d'algorithmes. Le choix de l'algorithme à utiliser dépend des données et de la situation. Voici quelques algorithmes de classification populaires :

- Méthode des K plus proches voisins (on K-nearest neighbors, KNN)
- Arbre de décision
- Random forest ou Forêt aléatoire
- Support Vector Machine

— **La régression**

La régression est un processus statistique prédictif dans lequel le modèle tente de trouver la relation importante entre les variables dépendantes et indépendantes. L'objectif d'un algorithme de régression est de prédire un nombre continu tel que les ventes, les revenus et les résultats de tests.

Les algorithmes cités précedemment peuvent être utilisés en cas de la régression.

L'apprentissage non supervisé

L'apprentissage non supervisé ne nécessite pas que le créateur "supervise" le modèle pendant la formation. L'apprentissage non supervisé permet au modèle de découvrir des modèles indépendamment d'un ensemble de données bien organisées et étiquetées. C'est pourquoi les ensembles de données non étiquetées sont souvent utilisés pour l'apprentissage non supervisé afin de générer des modèles d'apprentissage automatique.

Alors que les modèles d'apprentissage supervisé reçoivent les entrées et doivent prédire les sorties, les modèles d'apprentissage non supervisé prédisent à la fois les entrées et les sorties. Pour ce faire, ils détectent les schémas dominants entre les entrées et les associent aux sorties potentielles.
Les algorithmes spécifiques aux modèles d'apprentissage automatique non supervisés comprennent l'algorithme K-means, l'algorithme Apriori et le clustering hiérarchique.

L'apprentissage semi-supervisé

L'apprentissage semi-supervisé se situe à mi-chemin entre l'apprentissage non supervisé et l'apprentissage supervisé. L'apprentissage semi-supervisé nécessite généralement une combinaison d'une petite quantité de données étiquetées et d'une quantité relativement importante de données non étiquetées pour l'entraînement de ses modèles d'apprentissage automatique.

Comme pour l'apprentissage supervisé, les exemples de problèmes d'apprentissage automatique qui peuvent être résolus avec l'apprentissage semi-supervisé incluent la classification et les régressions. L'apprentissage semi-supervisé est utile pour la

classification d'images par exemple.

L'apprentissage par renforcement

Les algorithmes d'apprentissage automatique par renforcement sont une méthode d'apprentissage qui interagit avec son environnement en produisant des actions et en découvrant des erreurs ou des récompenses. Les caractéristiques les plus pertinentes de l'apprentissage par renforcement sont la recherche par essais et erreurs et la récompense différée. Cette méthode permet aux machines et aux agents logiciels de déterminer automatiquement le comportement idéal dans un contexte spécifique afin de maximiser ses performances. Un simple retour de récompense - appelé signal de renforcement - est nécessaire pour que l'agent apprenne quelle action est la meilleure.

2.2 Les algorithmes de machine learning

Nous détaillons dans cette section les algorithmes que nous allons utiliser ultérieurement.

2.2.1 Régression logistique

La régression logistique est la méthode la plus pratiquée pour modéliser une variable dépendante binaire. Elle a été utilisée avec réussite dans plusieurs domaines, principalement en médecine, par exemple pour détecter les facteurs liés à une maladie, dans le domaine bancaire pour détecter les profils à risque lors de la souscription d'un crédit et en marketing pour le scoring. À présent, cette méthode est la plus fréquente dans le secteur des assurances pour traiter des problèmes de classification binaire.

Dans cette étude, l'objectif de la régression logistique est d'expliquer la variable catégorielle Y à deux modalités : rachat ou non, à partir d'un vecteur de p variables explicatives X qui peuvent être quantitatives (continues) ou catégorielles (discrètes ou qualitatives).

Principes et propriétés mathématiques de la régression logistique

Le principe fondamental de tout modèle de régression logistique est d'exprimer l'espérance conditionnelle de la variable à expliquer Y sous forme d'une combinaison linéaire des variables explicatives X_i . Étant donné que Y suit une loi de Bernoulli, modéliser l'espérance conditionnelle $E(Y|X = x)$ revient à admettre la probabilité que $Y = 1$, car :

$$E(Y|X = x) = 1 * P(Y = 1|X = x) + 0 * P(Y = 0|X = x) = P(Y = 1|X = x)$$

Par la suite nous notons : $\pi(x) = P(Y = 1|X = x)$
Pour revenir à un modèle linéaire classique, nous utilisons la fonction lien "logit" définie par :

$$\text{logit(v)} = \ln(\frac{v}{1-v})$$

La courbe logistique est la suivante :

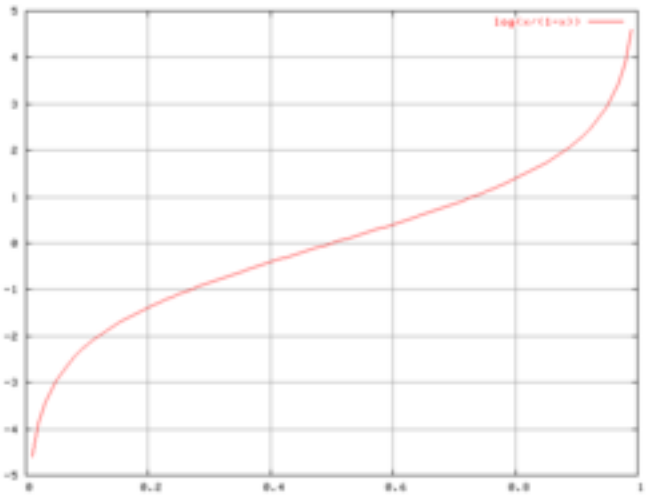

FIGURE 2.1 – *La courbe de la fonction logit*

L'application de la transformation logit permet de passer de l'intervalle $[\ 0;1]$ à l'intervalle $[\ -\infty;+\infty]$.

Ainsi, la formule du modèle logistique est définie comme suit :

$$\text{logit}(\ \pi(x)) \ = \ln(\frac{\pi(x)}{1-\pi(x)}) \ = X\beta = \beta_0 + \sum_{i=1}^{p} X_i\beta_i$$

Finalement, nous pouvons écrire $\pi(x)$ en fonction des coefficients β_i :

$$\pi(x) = \frac{e^{\beta_0+\sum_{i=1}^{p} X_i\beta_i}}{1+e^{\beta_0+\sum_{i=1}^{p} X_i\beta_i}}$$

Odds et odds-ratios

Le rapport $\pi(x)/[\,1-\pi(x)]$ est appelé odds (cote lors d'un pari), il exprime la chance de la réalisation de l'événement $Y = 1|X = x$ par rapport à $Y = 0|X = x$. Prenons l'exemple suivant : si la probabilité qu'un assuré effectue le rachat de son contrat est de 0.8 alors celle de ne pas racheter le contrat est de 0.2, l'odds vaut $0.8/0.2 = 4$ et donc cet individu a 4 fois plus de chance de racheter son contrat.

L'odds-ratio d'une variable explicative X_i mesure l'évolution du rapport des probabilités d'apparition de l'événement $Y = 1$ contre $Y = 0$. Il est défini comme suit :

$$OR_i = \exp(\beta_i)$$

Estimation des paramètres

Les paramètres β_i permettent de mesurer l'influence de chaque variable sur le modèle et donc d'identifier celles les plus discriminantes.

La méthode la plus utilisée pour estimer le paramètre inconnu $\beta = (\beta_0, \beta_1, ..., \beta_p)$ est celle du maximum de vraisemblance. D'après l'hypothèse d'Indépendance entre les n observations, le principe de cette méthode est de maximiser la fonction de vraisemblance suivante :

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} f(y_i|X = x_i)$$

Ce qui est équivalent à :

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} P(y_i|X = x_i)$$

$y_i \in \{0, 1\}$ est la valeur prise par la variable à expliquer Y et x_i désigne le vecteur de variables explicatives pour la $i^{\text{ème}}$ observation. Si $y_i = 1$, alors $P(y_i|X = x_i) = \pi(x_i)$. La fonction de vraisemblance s'écrit donc :

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} \pi(x_i)^{y_i}(1 - \pi(x_i))^{1-y_i}$$

La probabilité $\pi(x_i)$ est déjà calculée en fonction du vecteur de coefficients $\beta = (\beta_0, \beta_1, ..., \beta_p)$, donc en nous pouvons écrire $\pi(x_i)$ sous la forme :

$$\pi(x_i) = \frac{e^{\beta_0 + \sum_{j=1}^{p} X_i^j \beta_j}}{1 + e^{\beta_0 + \sum_{j=1}^{p} X_i^j \beta_j}}$$

Finalement, en remplaçant $\pi(x_i)$ dans l'expression de $\mathcal{L}(\beta)$, la fonction de vraisemblance permettant d'estimer les paramètres β_j est décrite comme suit :

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} \left(\frac{e^{\beta_0 + \sum_{j=1}^{p} X_i^j \beta_j}}{1 + e^{\beta_0 + \sum_{j=1}^{p} X_i^j \beta_j}}\right)^{y_i} \left(1 - \frac{e^{\beta_0 + \sum_{j=1}^{p} X_i^j \beta_j}}{1 + e^{\beta_0 + \sum_{j=1}^{p} X_i^j \beta_j}}\right)^{1-y_i}$$

En général, la solution analytique ne permet pas d'avoir la valeur de β qui maximise la fonction $\mathcal{L}(\beta)$. Toutefois il est possible de calculer la valeur de β par d'autres méthodes comme l'utilisation des algorithmes d'optimisation standards spécifiques à chaque logiciel.

Les paramètres de la régression logistique

L'algorithme de la régression logistique sous *python* dispose d'un paramètre principal qui sera modifié par la suite à l'aide d'un processus de paramétrage. Ce paramètre est :

C **:** c'est l'inverse du paramètre de régularisation λ (par défaut $C = 1$) .

Le but de la régularisation est de limiter le sur-apprentissage et ainsi d'accroître les performances du modèle sur de nouvelles données.

Les faibles valeurs de C permettent une régularisation plus importante.

Avantages et inconvénients

Avantages

— La régression logistique permet la sélection pas à pas de variables. Il existe trois méthodes de sélection pas à pas : *Forward*, *Backward* et *Stepwise*.

  — La méthode *forward* étudie d'abord un modèle comportant que le terme constant, puis intègre pas à pas les variables exogènes les plus influentes. Le processus s'arrête lorsqu'il n'y a plus de variables significatives.

  — La méthode *backward* retient d'abord le modèle complet, et puis élimine la variable la moins significative à chaque fois.

  — la méthode *stepwise* peut être considérée comme le mélange des deux méthodes précédentes. En cours de processus, les variables supplémentaires peuvent être acceptées et en même temps les variables déjà présentes peuvent être enlevées.

— La régression logistique est une approche paramétrique. En conséquence, elle fournit un coefficient pour chaque variable explicative et des intervalles de confiance à la sortie du modèle.

— La régression logistique détecte des optimums globaux. Contrairement à certaines méthodes d'apprentissage, la régression logistique renvoie des résultats d'un point de vue globale. Par exemple, à l'étape de la sélection de variables, nous regardons l'impact de la variable testée sur la qualité du modelé courant. Nous décidons de prendre une variable comme variable explicative si et seulement si elle améliore globalement le modèle courant.

Inconvénients :

— Comparée à d'autres algorithmes d'apprentissage comme l'arbre de décision, la régression logistique nécessite des traitements supplémentaires sur la base. Par exemple, la régression logistique ne traite pas les valeurs manquantes. De plus, les variables explicatives doivent être linéairement indépendantes dans la régression logistique afin de mieux sélectionner les variables significatives.

— Elle ne converge pas toujours vers une solution optimale. En pratique, les raisons de la non convergence sont variées. Par exemple, quand le terme de régularisation λ est mal réglé ou bien certaines variables sont mal découpées selon les expériences de l'utilisateur.

2.2.2 Arbre de décision

L'arbre de décision est un modèle prédictif dont l'objectif principal est d'expliquer une variable d'intérêt en fonction d'un ensemble de variables exogènes.

Il existe deux catégories d'arbres de décision :

— les arbres de classification dont la valeur à expliquer est catégorielle.

— les arbres de régression dont la valeur à prédire est continue.

L'idée de base est de partager un échantillon de données de manière ascendante en se basant sur des règles binaires et de visualiser les résultats par un arbre. L'ensemble des nœuds de l'arbre ainsi obtenu se divise en trois catégories [Figure 2.2] :

— Nœud racine : permet l'accès à l'arbre.

— Nœuds internes (ou noeuds fils) : les nœuds ayant des descendants.

— Nœuds terminaux (ou feuilles) : les nœuds qui n'ont pas des descendants.

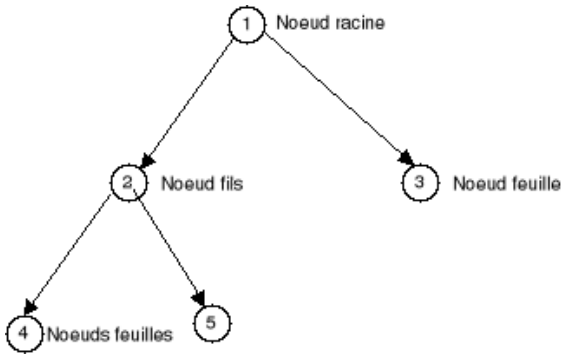

Figure 2.2 – *Structure de l'arbre de décision*

Construction de l'arbre de décision

Démarrant d'une base d'apprentissage avec p variables candidates. La totalité des individus sont initialement placés dans le nœud racine. L'algorithme va diviser alors la population successivement en deux sous-ensembles à chaque nœud de telle sorte que les deux nœuds descendants soient plus homogènes que le nœud parent et les plus différents possibles entre eux vis-à-vis de la variable d'intérêt.

Pour ce faire, l'algorithme choisit la variable X_i et la valeur de cette variable permettant la meilleure discrimination des données en maximisant un critère de séparation. Une fois cette première séparation est effectuée, l'algorithme réitère la même procédure sur chaque nœud obtenu jusqu'à une des conditions d'arrêt soit détectée.
Finalement, chaque branche de l'arbre se termine par une feuille à laquelle nous affectons la classe 0 ou 1 de la variable cible.
L'algorithme de l'arbre de décision est le suivant :

Algorithme 1 : Arbre de décision

Entrée : Échantillon d'apprentissage.
Initialiser l'arbre : Racine = noeud courant.
Répéter

Si noeud terminal alors lui affecter une classe et calculer les probabilités associées aux effectifs du noeud.

Sinon choisir la meilleure variable explicative qui fait le mieux progresser la discrimination des données et diviser en deux noeuds fils en maximisant le gain en information.

Fin Si

Passer au noeud suivant non-exploré (s'il existe).

Jusqu'à plus de noeud sans classe ou condition d'arrêt imposée.
Sortie : Arbre binaire de décision.
Lors de l'implémentation de l'arbre de décision, trois critères sont utilisés :

— Le critère d'arrêt permet d'arrêter la croissance de l'arbre. Si la croissance est arrêtée trop tôt, il y a un risque de sous-apprentissage. Dans le cas ou la croissance n'est pas arrêtée, l'apprentissage devient trop spécifique et nous sommes face à un problème de sur-apprentissage.

La solution consiste à construire l'arbre en entier puis réduire sa taille en utilisant

un critère (une mesure de performance) qui permet de comparer les performances de cette dernière à des arbres de tailles inférieures. Cette méthode se base sur une procédure de paramétrage afin d'obtenir un arbre plus performant en classification.

— Le critère de partitionnement permet de choisir parmi les p variables candidates celles qui maximisent le gain en information. Parmi les mesures de qualité de partitionnement, l'indice de Gini et l'Entropie de Shannon sont définis comme suit :

Soit K le nombre de classes, m le nœud courant et P_i la proportion d'individus de la $i^{\text{ème}}$ classe.

— Indice de Gini :

$$i_g(m) = \sum_{k=1}^{K} P_i(1 - P_i) = 1 - \sum_{k=1}^{K} P_i^2$$

— Entropie de Shannon :

$$i_e(m) = \sum_{k=1}^{K} -P_i \log(P_i)$$

— Le critère d'affectation permet d'attribuer une classe à chaque nœud terminal. Un nœud est considéré terminal quand il est homogène : la plupart des individus de ce nœud sont déjà dans la même classe ou bien il n'y a plus de variables exogènes à tester. Ainsi, la classe prédite d'un nouveau candidat est celle affectée à la feuille où il arrive. En général, la classe affectée est la classe la plus présente dans la feuille.

Dans le cas où il y a une classe majoritaire par rapport aux autres classes, la classe affectée coïncide le plus souvent avec la classe majoritaire.

Les paramètres de l'arbre de décision

Dans cette étude, la librairie *sklearn* de *python* propose l'algorithme "*DecisionTree-Classifier*" pour l'implémentation des arbres de décision. Cet algorithme utilise l'indice de Gini et construit l'arbre maximal, puis à l'aide d'une modification de ses paramètres, l'algorithme sélectionne le sous arbre optimal pour la classification.
les principaux paramètres de l'algorithme de l'arbre de décision sont :

— **max-depth :** c'est la profondeur maximale de l'arbre.

— **min-samples-leaf :** c'est le nombre minimal d'individus pouvant constituer une feuille.

— **min-samples-split :** c'est le nombre minimal d'individus présents dans un noeud interne pour effectuer sa division.

Avantages et inconvénients

Avantages

— Les arbres de décision savent traiter directement les valeurs manquantes selon l'ordre imposé. Une fois que l'arbre détecte une variable qui contient des valeurs manquantes, il est possible de créer automatiquement une modalité qui regroupe toutes ces valeurs manquantes. Nous pouvons aussi paramétrer l'arbre pour que l'algorithme supprime directement les individus qui contiennent des valeurs vides.

— Contrairement à la régression logistique qui est une méthode paramétriques, l'arbre est capable d'étudier des variables colinéaires sans produire un problème de convergence. De plus, la variable à expliquer peut être non linéaire en fonction des variables explicatives sélectionnées à la sortie de l'arbre. Autrement dit, l'étude de corrélation est optionnelle quand nous voulons modéliser par l'arbre binaire de décision.

— Une classification supervisée à la sortie de l'arbre qui fait parler les données nous permet de prédire la variable à expliquer. Les résultats sont clairement représentés

sous la forme graphique d'un arbre. Il en découle une grande compréhensibilité des résultats.

— Les arbres sont très faciles à généraliser. Les méthodes existantes qui sont basées sur l'arbre de décision sont nombreuses. Ces méthodes ont été développées dans le but d'améliorer la qualité de l'arbre. Parmi ces méthodes, l'algorithme de la forêt aléatoire et le *boosting* d'arbres.

Inconvénients

— L'arbre manque de robustesse. Les modèles obtenus dépendent fortement de l'échantillon. La modification de l'échantillon peut entièrement changer tous les paramètres de l'arbre.

— L'arbre détecte des optimums locaux et non globaux [**st**]. A l'opposé du modèle logistique, l'arbre ne peut pas fournir une segmentation globale pour chaque variable sélectionnée. L'arbre segmente la variable choisie pour chaque noeud. Dans une situation complexe, une seule variable peut être présente plusieurs fois dans des chemins différents. Les segmentations de cette variable multi-utilisée sont souvent variées selon les chemins associés. L'arbre est incapable de centraliser toutes les informations locales afin de renvoyer une segmentation globale pour chaque variable retenue.

— La technique des arbres de décision est non-paramétrique, ce qui signifie qu'il n'y a pas de coefficient estimé à la sortie du modèle. L'effet de chaque modalité n'est pas évident sans l'aide d'un coefficient correspondant. Nous n'avons donc aucun accès aux relations relatives entre les modalités.

2.2.3 Bagging et forêt aléatoire

Le *bagging* regroupe un ensemble de méthodes introduit par Lèo Breiman en 1996 [2] . Le terme *bagging* vient de la contraction de *bootstrap Aggregating*.

L'approche *bagging* consiste à agréger des classifieurs en les construisant sur des échantillons *bootstrap*. Un échantillon *bootstrap* est obtenu en tirant aléatoirement n individus avec remise de la base d'apprentissage. Le classifieur *bagging* affecte aux observations la classe majoritaire parmi les classifieurs *bootstrap*. La classification finale s'obtient par un vote à la majorité pour les variables catégorielles ou une moyenne pour les variables continues.

Le *bootstrap* est beaucoup moins sensible à l'échantillon d'apprentissage, ainsi, il permet de réduire la variance du modèle.

L'algorithme du *bagging* est le suivant :

Algorithme 2 : Bagging

Entrée : M le nombre de classifieurs.

Faire pour $m = 1$ à M.

Tirer un échantillon *bootstrap*.

Construire un classifieur f_m sur chaque échantillon *bootstrap*.

Estimer f_m sur l'échantillon *bootstrap*.

Fin pour

Sorties : * Une estimation moyenne $f(x) = \frac{\sum_{m=1}^{M} f_m(x)}{M}$ pour des variables continues.

* Un vote majoritaire parmi les $f_m(x)$ pour des variables catégorielles.

Forêt aléatoire

Les forêts d'arbres décisionnels (ou forêts aléatoires de l'anglais *random forest*) ont été proposées en 2001 par Leo Breiman[3]. Elles font partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de *bagging*. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles des données légèrement différents.

En fin de l'algorithme, nous possédons des arbres que nous moyennons et la classification d'un individu est prise par vote à la majorité.

Algorithme 3 : Forêts aléatoires

Entrée : * M : le nombre d'arbres.

* q : le nombre de variables aléatoires.

Faire pour $m = 1$ à M.

Tirer un échantillon *bootstrap*.

Tirer un nombre q de variables aléatoires.

Construire un arbre de décision f_m sur l'échantillon *bootstrap* et les q variables.

Fin pour

Sorties : * Une estimation moyenne $f(x) = \frac{\sum_{m=1}^{M} f_m(x)}{M}$ pour des variables continues.

* Un vote majoritaire parmi les $f_m(x)$ pour des variables catégorielles.

Le choix aléatoire des q variables explicatives pour effectuer la division des noeuds augmente la variabilité du modèle. Ainsi, chaque modèle est moins performant, mais l'agrégation de tous les modèles est plus performante.

La figure suivante représente le fonctionnement de la forêt aléatoire :

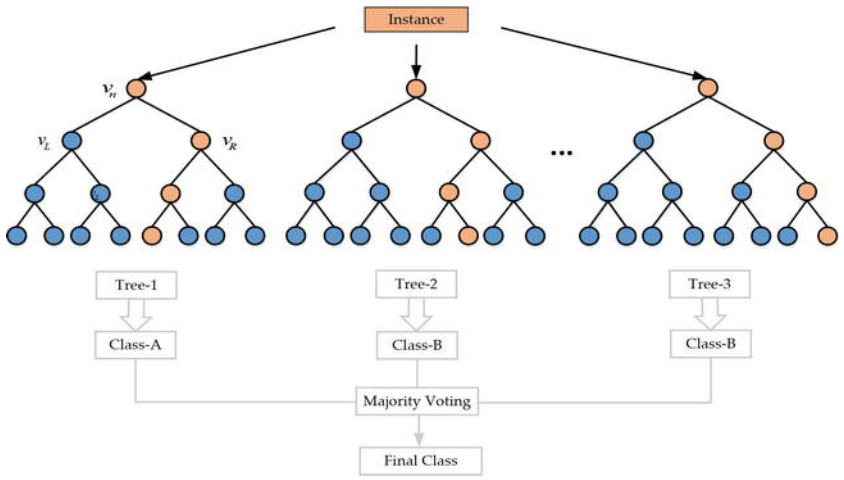

Figure 2.3 – *Fonctionnement de la forêt aléatoire*

Les paramètres de la forêt aléatoire

L'algorithme proposé par la librairie *sklearn* de *python* pour implémenter les forêts aléatoires est "*RandomForestClassifier*". Afin de tenter d'optimiser le paramétrage, nous avons choisi de jouer sur trois paramètres de cet algorithme :

— **n-estimators :** le nombre d'arbres créés par l'algorithme.

— **max-depth :** la profondeur maximale de chaque arbre.

— **max-features :** le nombre de variables utilisées pour diviser un noeud. L'algorithme sélectionnera alors q (*max-features*) variables parmi les p variables explicatives.

Avantages et inconvénients

Avantages

— Elles sont très simples à mettre en œuvre et très fructueuses en grande dimension.

— D'après leur méthode de construction, les forêts aléatoires offrent des modèles de

prédiction particulièrement robustes et fiables.

Inconvénients :

Comme il s'agit d'une méthode d'agrégation des classifieurs, la forêt aléatoire ne permet pas une interprétation directe des résultats. Le modèle, en contrepartie, permet une présentation graphique de l'importance de chaque variable explicative dans le modèle agrégé. L'importance des variables dépend de la fréquence d'apparition et de la place qu'elles occupent dans chaque arbre.

2.2.4 Boosting et Extreme Gradient Boosting

Le *boosting* repose sur le même principe que le *bagging* (forêt aléatoire). Au lieu d'utiliser un seul modèle, il combine plusieurs classifieurs à faible pouvoir prédictif pour obtenir un seul résultat. Dans la construction du modèle, contrairement au *bagging* qui construit tous les modèles en un seul temps, le *boosting* travaille d'une manière séquentielle. À tout instant t, les résultats du modèle sont pondérés en fonction des résultats de l'instant précédent $t-1$. Les résultats prévus correctement recevront un poids plus faible et ceux qui ont été mal classés auront un poids plus important.
L'exemple ci-dessous illustre bien le fonctionnement de la méthode *boosting* :

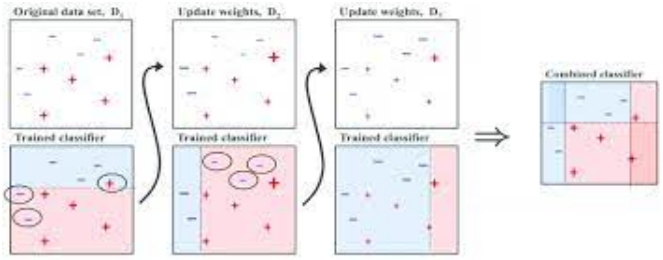

Figure 2.4 – *Principe du boosting*

Boite1 : Résultat du premier classifieur :
Initialement, tous les points ont le même poids (illustré par leur taille).
Le premier modèle prédit 2 points (+) et 5 points (-) correctement.

Boite 2 : Résultat du deuxième classifieur :
Les points bien classés à la sortie de la première boite ont un poid plus faible et vis-vers-ça. Le modèle se concentre sur les points ayant des poids élevés et les classe correctement. Mais, d'autres sont mal classés maintenant.
La même procédure peut également être observée dans la boite 3. Cette opération sera réitérée plusieurs fois. Finalement, tous les modèles reçoivent un poids en fonction de leur précision et un résultat consolidé est généré.

Gradient boosting

Les algorithmes de *boosting* existants diffèrent principalement par leurs manières de pondérer les individus mal classés.

Principe du Gradient Boosting

Nous avons choisi la méthode de "*Gradient Boosting*" qui utilise le principe du *boosting* sur les arbres de décision. L'idée est de construire des arbres de décision et de les ajouter d'une manière séquentielle au modèle. De nouveaux arbres sont créés pour corriger les erreurs de classification dans les prédictions de la séquence d'arbres existante.
Cette méthode présente un risque de sur-apprentissage sur la base d'apprentissage puisque le modèle peut apprendre rapidement. Afin de contrôler l'apprentissage dans le modèle de *gradient boosting*, nous introduisons un paramètre de pondération pour les corrections apportées par les nouveaux arbres lorsqu'ils sont ajoutés au modèle.
Ce paramètre, ayant l'objectif de "lisser" les mises à jour, est compris entre 0 et 1 ($0 < \nu \leq 1$). Empiriquement, on constate que ν faible ($\nu < 0.1$) améliore les performances prédictives, mais au prix d'une convergence plus lente (nombre d'itérations plus élevé).

l'Extreme Gradient boosting

L'extreme Gradient Boost est une version particulière de l'algorithme de Gradient Boost. En effet, il s'agit d'un assemblage de "weak learners" qui prédisent les résidus, et corrigent les erreurs des "weak learners" précédents.

La particularité de cet algorithme réside dans le type de "weak learner" utilisé. Les "weak learners" sont des arbres décisionnels. Les arbres qui ne sont pas assez bons sont "élagués", c'est à dire qu'on leur coupe des branches, jusqu'à ce qu'ils soient suffisamment performant. Sinon ils sont complètement supprimés.

Ainsi, l'extreme gradient boost s'assure de ne conserver que de bons weak learners.

Avantages et inconvénients

Avantages

— En plus de réduire la variance comme le *bagging*, le *boosting* permet aussi de réduire le biais de prévision.

— En présence de classifieurs faibles (classifieurs juste un peu meilleurs que le hasard), le *boosting* a de meilleures performances que le *bagging*.

Inconvénients

De même que pour les forêts aléatoires, le *boosting* ne permet pas l'interprétation directe des résultats. Mais, il permet une illustration graphique de l'importance des variables explicatives dans le modèle.

2.2.5 Les mesures de performance

Après avoir implémenter les algorithmes d'apprentissage, l'étape suivante consiste à mesurer leurs performances.

Une règle de classification consiste en une fonction qui attribue un score s à un individu selon les variables explicatives. Les individus sont assignés à la classe 1 si leur score est supérieur à un seuil t et ils sont assignés à la classe 0 si $s < t$. Les règles de classification ne sont pas parfaites, elles affectent certains individus à la mauvaise classe. L'objectif des mesures de performance est de mesurer l'importance des erreurs de classification.

Afin de mesurer les performances d'un modèle, le modèle est appliqué à des individus dont la classe est connue. La base d'apprentissage sera utilisée pour construire les règles de classification et la base de test servira à mesurer les performances. Par la suite, nous utiliserons une validation croisée pour séparer l'apprentissage de la validation. Le principe de la validation croisée sera détaillé plus tard.

À la suite de la comparaison des scores au seuil t, une proportion des individus de la classe 0 et une proportion des individus de la classe 1 sont mal classées. Ces mauvais classements, ainsi que les bons classements, peuvent être illustrés par une matrice de confusion.

Matrice de confusion

La matrice de confusion croise la classe réelle des individus avec la classe prédite par le modèle.

		Classe réelle	
		Negative	Positive
Classe prédite	Negative	Vrai négatif (TN)	Faux négatif (FN)
	Positive	Faux positif (FP)	Vrai positif (TP)

Table 2.1 – *Matrice de confusion*

La matrice de confusion en elle-même n'est pas une mesure de performance, mais la plupart des métriques de performances sont basées sur les valeurs de cette matrice qui sont :

- **Vrais positifs(TP) :** ce sont les individus dont la classe réelle est 1 (Positif) et leur classe prédite est également 1 (Positif).
- **Vrais négatifs (TN) :** ce sont les individus dont la classe réelle est 0 (Négatif) et leur classe prédite est également 0 (Négatif).
- **Faux positifs (FP) :** ce sont les individus dont la classe réelle est 0 (Négatif) et leur classe prédite est 1 (Positif). Dans ce cas, le modèle prédit incorrectement la classe de ces individus.
- **Faux négatifs (FN) :** ce sont les individus dont la classe réelle est 1 (Positif) et leur classe prédite est 0 (Négatif).

A partir de cette matrice de confusion, plusieurs mesures de performance peuvent être définies comme l'accuracy, la sensibilité, la spécificité, le rappel et la précision.

La majorité des algorithmes d'apprentissage statistique considèrent des classes équilibrées. Ces algorithmes performent beaucoup mieux lorsque les classes sont équilibrées, mais leur performance baisse en cas de déséquilibre des classes. Généralement, ces algorithmes ne classent pas correctement les classes minoritaires.

Accuarcy

En général, l'accuracy est la mesure de performance la plus utilisée. Elle présente le rapport entre le nombre de bonnes prédictions faites par le modèle et le nombre total de prédictions.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

Cependant, l'accuracy ne devrait jamais être adaptée pour évaluer un modèle lorsque les classes de la variable cible sont en situation de déséquilibre. Dans ce cas, l'accuracy ne distingue pas les mauvaises classifications selon les classes. Par exemple, avec 3% d'individus positifs, si le classifieur prédit tous les individus en négatifs, le taux d'accuracy est de 97%. C'est un taux élevé mais ne reflète pas le fait que tous les individus positifs sont mal classés (TP=0).

Dans le cas où les classes sont déséquilibrées, nous devons choisir des mesures de performance spécifiques qui prennent en considération la distribution des classes.

Les mesures de performance suivantes sont définies :

$$Sensibilité = \frac{TP}{TP + FN}$$

$$Spécificité = \frac{TN}{TN + FP}$$

Courbe ROC

La courbe ROC (*Receiver Operating Characteristics*) permet de visualiser les performances d'un classifieur. Pour chaque individu, la probabilité d'appartenance à la classe d'intérêt est calculée. Ensuite les individus sont représentés dans l'espace : sensibilité/(1-spécificité). Pour quantifier cette approche graphique, l'aire sous la courbe ROC est calculé. Ce critère de performance s'appelle l'AUC (*Area Under the Curve*). Plus l'AUC est élevé, plus le modèle est meilleur.

La précision et le rappel

D'autres mesures de performance comme le rappel et la précision peuvent être définies :

— La précision mesure le nombre de vrais positifs par rapport au nombre de positifs prédits.

$$Précision = \frac{TP}{TP + FP}$$

— Le rappel mesure le nombre de vrais positifs par rapport au nombre de positifs réels.

$$Rappel = \frac{TP}{TP + FN}$$

La précision et le rappel évaluent les performances d'un modèle sur une modalité spécifique. Ces deux mesures sont plus intéressantes dans le cadre de classes déséquilibrées que le taux d'erreur, que la spécificité et la sensibilité car elles favorisent moins la classe majoritaire.

— La F-mesure correspond à un compromis entre la précision et le rappel. La F-mesure est une moyenne harmonique pondérée par un coefficient β

$$F\text{-}mesure = \frac{(1 + \beta^2) Précision.Rappel}{\beta^2 Précision + Rappel}$$

Le coefficient β permet d'équilibrer le poids entre le rappel et la précision. En général, $\beta = 1$.

Chapitre 3

ANALYSE ET PRÉPARATION DES DONNÉES

La phase de préparation des données d'étude est indispensable pour le processus de modélisation. En effet, le pouvoir prédictif de nos modèles dépend étroitement de la qualité des données.
De ce fait, ce chapitre sera consacré à la présentation, le traitement et l'étude de nos données.

3.1 Description de la base de données

Notre portefeuille se compose des produits d'épargne rachetables de MAGHREBIA VIE. Les polices étudiés sont souscrites entre 1985 et 2016 (en cours , échu et racheté).

3.1.1 Le fichier Base 17-18-19

Ce fichier contient l'état de portefeuille de 2017 à 2019 incluant les informations sur les contrats ainsi que les assurés. Il est composé de 20 942 lignes et de 9 colonnes :

— **ID_Contrat**
— **Prime :** la prime payée à la souscription.
— **Fréquence :** la fréquence des versements : mensuelle, trimestrielle, semestrielle et annuelle .
— **Date_Effet :** la date de souscription du contrat.

— **Date_Echéance :** la date d'échéance du contrat.

— **Réseau_Distribution :** Producteur vie ou Agent.

— **Date_de_naissance**

— **Sexe :** M pour homme et F pour femme.

3.1.2 Le fichier Rachat

Il contient une description des contrats rachetés (ID_contrat,type de prestations et date de survenance) sur la période 2017-2019. Elle est composée de 3 colonnes et 1303 lignes.

Ce fichier est uniquement utilisé pour déterminer la variable d'intérêt.

3.2 Préparation des données

L'implémentation d'un modèle statistique nécessite un traitement préalable des variables présentes dans la base de données. En effet, la présence des valeurs manquantes, de valeurs extrêmes, de valeurs non cohérentes pose certains problèmes que nous allons détailler dans cette étude.

3.2.1 Fusion des fichiers

En premier lieu, nous avons opté pour la fusion des fichiers "Base 17-18-19" et "Rachat" en utilisant comme critère principal le "ID_Contrat". Il est à noter que chaque référence du contrat est unique.

Nous avons utilisé la table "Rachat" afin de créer notre variable d'intérêt "Rachat" qui détermine s'il y a eu une prestation de rachat ou non :

$$Y=\begin{cases} 1 & \text{si ID\_Contrat existe dans le fichier Rachat .} \\ 0 & \text{sinon.} \end{cases}$$

3.2.2 Transformation des variables

Regroupement des variables continues

Pour le but de faciliter l'interprétation des résultats ultérieurement, nous avons procédé à des regroupement des variables continues dans des classes comme suit :

Nom de la variable	**Modalités**
Prime	]0,1000[, [1000,2000[, [2000,5000[, [5000,10000[, Sup à 10000
Age_Survenance	moins de 18, [18,40[, [40,50[, [50,60[, plus de 60
Age_Souscription	moins de 18, [18,40[, [40,50[, [50,60[, plus de 60
Ancienneté_Contrat_Survenance	[0,10[, [10,15[, [15,20[, Sup à 20

Table 3.1 – *Liste des variables et leurs modalités*

Conversion des variables dates

Généralement, les dates dans une base de données sont difficiles à exploiter, c'est pour cette raison que nous avons dégagé à partir de celle-ci des variables susceptibles d'expliquer le phénoméne de rachat.

A partir de la date de naissance, la date d'effet, la date d'échéance et la date de survenance, nous avons pu calculer l'âge à la souscription, l'âge à la date de survenance et l'ancienneté du contrat à la date de survenance ,ce qui a rendu la manipulation des données plus facile.

3.2.3 Nettoyage des données

Nous avons procédé à la suppression des valeurs manquantes pour les variables "date de naissance" et "fréquence" ainsi que les valeurs incohérentes dans toute la base. Nous avons supprimé les variables inutiles pour la modélisation ("ID_contrat", "date

de naissance", "date d'effet", "date d'échéance" et "la date de survenance").

3.3 La base finale

A partir des traitements développés dans ce chapitre, nous présentons la base de données finale qui sera utilisée pour paramétrer les modèles statistiques. Ainsi, la taille de la base est ramenée à 17 859 observations et 7 variables.

La table ci-dessous résume les variables explicatives de la base finale :

Nom de la variable	**Type**	**description**
Prime	présent	la prime payée la premiére année
Fréquence	présent	la fréquence des versements de la prime
Réseau de distribution	présent	la canal de distribution de la police
Sexe	présent	le genre du détenteur de la police
Age_Souscription	calculé	l'age de l'assuré à la souscription de la police
Age_Survenance	calculé	l'age de l'assuré à la survenance du rachat
Ancienneté_Contrat_Survenance	calculé	l'ancienneté du contrat à la date du rachat

Table 3.2 – *Description des variables explicatives du modèle*

Chapitre 4

ÉVALUATION DES RÉSULTATS

Dans ce chapitre, nous présentons puis analysons les résultats de l'étude.
Le logiciel utilisé pour implémenter les algorithmes d'apprentissage est *python*, essentiellement à travers la librairie *sklearn*.
Par la suite, pour chaque méthode nous présentons la librairie utilisée.[4]

Dans ce chapitre, la première partie présente les traitements préliminaires effectués sur la base de données. Les méthodes de classification sont ensuite évaluées sur notre base de données. En conclusion, le modèle final retenu est exposé.

4.1 Travail préliminaire

Nous suivons dans cette partie la démarche classique de tout projet de machine learning. Nous appliquons d'abord les étapes de data engineering qui consistent à "nettoyer" et à "réparer" les données erronées, à transformer les variables qualitatives et sélectionner les varibles les plus pertinentes pour chaque algorithme. Après ce traitement, nous appliquons un découpage de la base. Les modèles d'apprentissage seront appliqués sur la base d'apprentissage et le modèle qui ressort sera testé sur l'échantillon appelé test. La dernière étape consiste à comparer et à tester la robustesse des modèles.

4.1.1 Traitement des variables qualitatives

La librairie *sklearn* ne supporte pas les variables qualitatives. Donc, avant de lancer les algorithmes, nous procédons à une transformation de chaque variable qualitative en un vecteur de variables *dummy* (*dummy* variables) de la manière suivante :

— Chaque modalité de la variable devient elle-même une variable *dummy* qui prend 1 si cette modalité est présente pour une instance fixée, et 0 si la modalité n'est pas présente pour cette instance.

— La variable qualitative initiale est éliminée de la base, et seules les variables représentant les modalités sont retenues.

Notons qu'on a La figure suivante illustre un exemple de ce traitement effectué sur la variable "Fréquence" :

ID_Contrat	Fréquence
1	M
2	T
3	S
4	A

ID_Contrat	Fréquence_M	Fréquence_T	Fréquence_S	Fréquence_A
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Figure 4.1 – *Transformation de la variable Fréquence*

A la suite de la transformation des 7 variables qualitatives présentes dans la base, le nombre de variables explicatives passe de 7 à 27 variables explicatives.

4.1.2 Sélection de variables

Il s'agit de sélectionner les variables explicatives les plus pertinentes pour chaque algorithme.

La fonction RFECV (*Recursive Feature Elimination with cross validation*) permet la sélection de variables. La méthode de sélection consiste à construire un modèle sur toutes les variables explicatives et choisir la variable la moins importante ; l'importance de chaque variable est mesurée par son coefficient. Cette variable est éliminée par la suite, et le processus est répété avec le reste des variables jusqu'à avoir la

meilleure performances.

4.1.3 Base apprentissage-Base test

Une fois la base de données est prête à être exploitée, pour commencer à mettre en place des modèles prédictifs de la sinistralité, nous avons dû procéder à la création d'une base d'apprentissage et d'une base de test. Nous avons choisi un découpage selon une répartition de données de 70% pour la base d'apprentissage et de 30% pour la base de test.

La première base essentielle à la réalisation de nos modèles est la base d'apprentissage. C'est à partir de cette base là que seront réalisés tous les modèles qui suivront.

Ces modèles ainsi créés devront être exécutés sur une seconde base, que l'on appelle base de test. C'est sur ce portefeuille que seront estimées les prédictions.

Dans tout ce qui suit, nous notons :
— **RL :** régression logistique.
— **AD :** arbre de décision.
— **FA :** forêt aléatoire.
— **XGB :** extreme gradient boosting.

4.2 Résultats des différents algorithmes

En premier lieu, nous avons implémenté les algorithmes d'apprentissage avec leurs paramètres par défaut ; sans effectuer aucun paramétrage.
Ensuite, nous avons eu recours à la méthode“*GridSearchCV* " pour avoir les meilleurs paramétres qui nous donne les meilleurs performances.

4.2.1 Régression logistique

Le paramètre de régularisation optimal obtenu aprés l'application de GridSearchCV est : **C**=0.001.

Les variables retenues pour la régression logistique après sélection des variables par la méthode RFECV sont : **"Age-Survenance-18-40", "Age-Survenance-40-50", "Ancienneté-Contrat-<10"et "Ancienneté-Contrat->=20".**

Les coefficients estimés β_i et les odds-ratio OR_i de chaque variable X_i retenue ainsi que la constante (intercept) du modèle β_0 sont présentés dans la table suivante :

Nom de la variable	**Coefficient estimé** $\hat{\beta}_i$	**Odds-ratio** OR_i
constante	$\hat{\beta}_0 = 0$	
Age-Survenance-18-40	$\hat{\beta}_1 = 0.048$	$OR_1 = 1.049$
Age-Survenance-40-50	$\hat{\beta}_2 = 0.115$	$OR_2 = 1.129$
Ancienneté-Contrat-<10	$\hat{\beta}_3 = 0.031$	$OR_3 = 1.031$
Ancienneté-Contrat->=20	$\hat{\beta}_4 = -1.613$	$OR_4 = 0.199$

Table 4.1 – *Coefficients et odds-ratio du modèle retenu*

La variable "Age-Survenance-40-50" est la variable la plus pertinente dans le modèle vu qu'elle admet le coefficient le plus élevé.

Interprétation des odds-ratio

— $OR_1 = 1.049 > 1$: si l'age de l'assuré à la survenance est compris entre 18 et 40 ans, alors il a une forte probabilité de racheté son contrat. et $OR_2 = 1.129 > 1$: si l'age de l'assuré à la survenance est compris entre 40 et 50 ans, alors il a une forte probabilité de racheté son contrat. et $OR_3 = 1.031 > 1$: Si l'ancienneté du contrat à la survenance est inférieur à 10 ans, il est susceptible de racheté son

contrat.

— $OR_3 = 0.199 < 1$: Si l'ancienneté du contrat à la Survenance est supérieur à 20 ans, il est moins problable que l'assuré rachéte son contrat.

4.2.2 Arbre de décision

Les meilleurs paramétres qui correspondent à la meilleur performance, obtenues avec GridSearchCV qui sont :

— **max-depth :** la profondeur maximale de l'arbre=5

— **min-samples-split :**le nombre minimal d'individus présents dans un noeud interne pour effectuer sa division = 30

— **min-samples-leaf :** le nombre minimal d'individus pouvant constituer une feuille = 20

Graphe de l'arbre de décision

Le graphe de l'arbre de décision obtenu[Annexe] comporte 7 feuilles qui classent les individus dans la classe positive "1".

Ici l'arbre est expliqué de la manière suivante :

Si la modalité cité dans le noeud est présente pour l'observation, prendre le chemin à droite sinon prendre le chemin à gauche.

Ce processus est répété pour chaque noeud jusqu'à la fin de l'arbre (arriver à une feuille).

Finalement, la classe affectée est la classe majoritaire des individus qui ont suivi le même chemin.

Importance des variables :

Le graphe ci-dessous représente l'importance des variables explicatives pour l'arbre de décision obtenu :

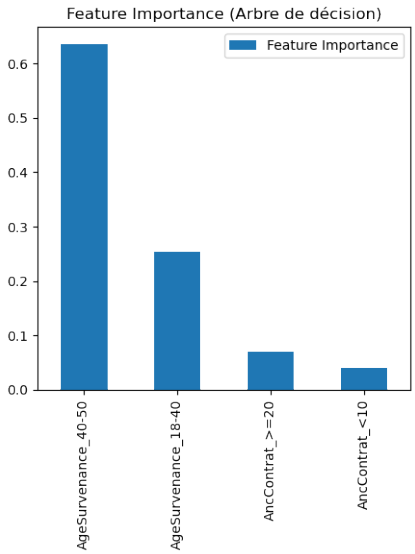

Figure 4.2 – *Importance des variables pour l'arbre de décision*

La variable "Age-Survenance-40-50" est la variable la plus pertinante. Sa contribution est de 63.5% suivie par la variable "Age_Survenance_18-40" dont l'importance est de 25,4%.

4.2.3 Forêt aléatoire

Une optimisation des paramétres par GridSearchCV nous a permis d'avoir la forêt aléatoire la plus performante. Les paramètres optimaux de l'algorithme dans ce cas sont :

— **n-estimators :** le nombre d'arbres créés par l'algorithme = 100

— **max-depth :** la profondeur maximale de chaque arbre = 4

— **max-features :**le nombre de variables utilisées pour diviser un noeud= 2

Les variables retenues pour cet algorithme après sélection des variables par la méthode RFECV sont : "Age-Souscription-18-40", "Age-Survenance-18-40", "Age-Survenance-40-50", "Age-Survenance-50-60", "Age-Survenance->60", "Ancienneté-Contrat-0-10", "Ancienneté-Contrat-<10 et "Ancienneté-Contrat->=20".

Importance des variables : Le graphe suivant montre la contribution de chaque variable explicative dans le modèle :

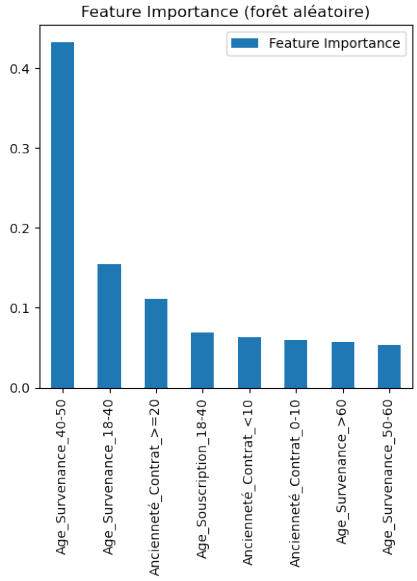

Figure 4.3 – *Importance des variables pour la forêt aléatoire*

Les variables "Age-Survenance-40-50","Age-Survenance-18-40","Ancienneté-Contrat->=20" sont les les plus discriminantes, leur contribution ensemble dans le modèle est de 69%.

4.2.4 Extreme Gradient Boosting

Les paramètres optimaux de l'algorithme obtenus suite à la méthode *Grid Search CV* sont :

— **learning-rate**= 0.01

— **n-estimators**= 100

— **max-depth**= 4

Les variables retenues pour cet algorithme après sélection des variables par la méthode RFECV sont :

"Age-Souscription-18-40", "Age-Survenance-18-40","Age-Survenance-40-50" et "Age-Survenance-50-60"

L'importance des variables dans le modèle ainsi obtenu est présentée dans la figure ci-dessous :

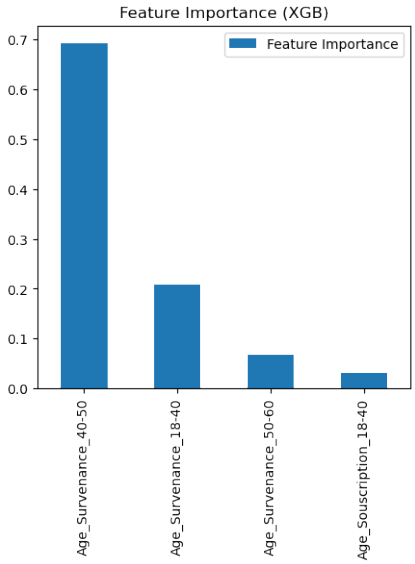

Figure 4.4 – *Importance des variables pour l'extreme gradient boosting*

Les trois variables les plus pertinentes dans le modèle sont : "Age-Survenance-40-50" avec une contribution de 69%, "Age-Survenance-18-40" qui contribue de 21% et "Age-Survenance-50-60" avec une contribution de 7%.

4.2.5 Comparaison entre les modèles

Le tableau et la figure ci-dessous récapitulent les performances des algorithmes d'apprentissage :

	Rappel	*Précision*	*F-mesure*
RL	0.69	0.92	0.79
AD	0.67	0.90	0.77
FA	0.67	0.90	0.77
XGB	0.64	0.84	0.73

Table 4.2 – *Performances des algorithmes d'apprentissage*

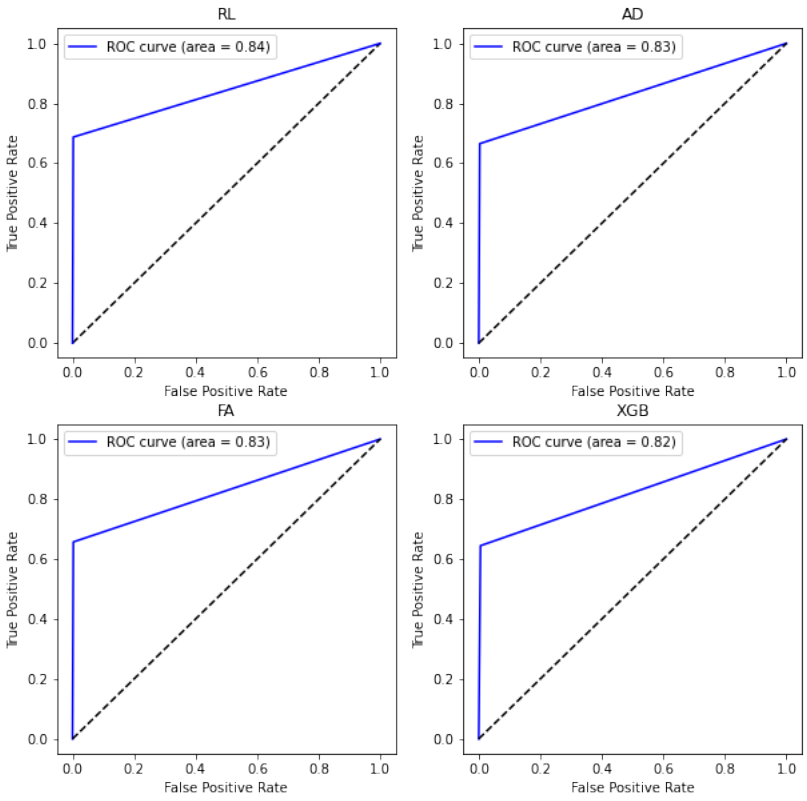

FIGURE 4.5 – *Courbes de ROC*

Discussion des résultats

Les deux variables “Age-Survenance" et “Anciennete-Contrat-Survenance" sont les plus discriminantes pour nos modéles.
Les résultats sont trés proches en terme des différents mesures de performance.

Le modéle le plus performant en terme de rappel, précision et F-mesure est la régression logistique.

CONCLUSION

Le rachat est un risque qui peut compromettre la solvabilité des compagnies d'assurance. Dans un marché très concurrentiel tel que le marché de l'assurance vie, il est important de bien gérer le risque de rachat.

Notre étude propose des méthodes de machine learning dont le but est de détecter le profil de l'assuré risqué et les paramétres qui influencent son comportement.

Les algorithmes utilisés sont la régression logistique et des algorithmes basés sur les arbres de décision. Il aurait été intéressant de comparer les différentes méthodes utilisées avec un modèle de réseau de neurones ou un modèle SVM (*Support Vector Machine*), ce qui n'a pu être fait faute de temps.

Il serait intéressant aussi de tester les différentes méthodes sur plusieurs jeux de données.

L'étude du rachat par les algorithmes de machine learning peut être exploitée afin d'améliorer les stratégies de commercialisation des produits ou la tarification des produits d'assurance.

BIBLIOGRAPHIE

[1] Skander LAHRIZI. *Cours assurance vie.* (2021).

[2] Leo Breiman. "Bagging predictors". In : *Machine learning* 24.2 (1996), p. 123-140.

[3] Leo Breiman. "Random forests". In : *Machine learning* 45.1 (2001), p. 5-32.

[4] *Python spécial machine learning.* `https://www.youtube.com/playlist?list=PLO_fdPEVlfKqMDNmCFzQISI2H_nJcEDJq`.

[5] Houweida JEMLI. *: Étude du risque de rachat de produits d'épargne italiens par des données agrégées et individuelles.* Université Paris Dauphine.

[6] Naoufal RAKAH. *: Modélisations des rachats dans les contrats d'épargne.* Centres d'études actuarielles, (2012).

ANNEXE

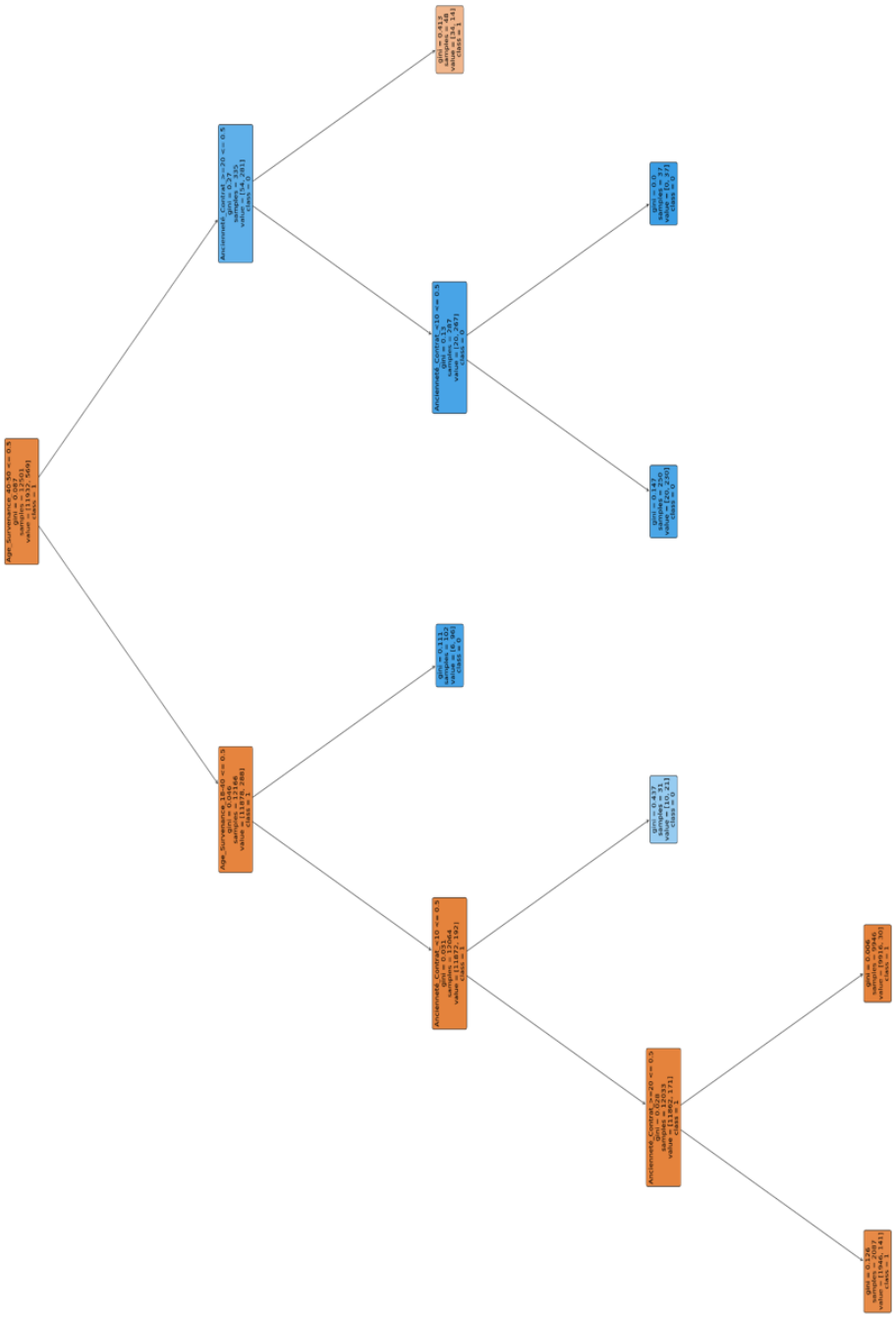

Figure 4.6 – *Arbre de décision*