République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaia

Faculté des Sciences Exactes

Département de Recherche Opérationnelle

Mémoire de fin de cycle

Domaine : Mathématiques Informathiques

Filière : Mathématiques Appliquées

Spécialité : Mathématiques Financières

Thème

Machine Learning et Application en finance

Réalisé par :

— M^{lle}.Alimrina karima

Proposée par :

M^{lle}.Zohra Aoudia

Soutenu septembre 2022 devant le jury composé de :

Dr B. Barahmi	Président
Dr Y. Ziane	Examinatrice
Dr L. Djerroud	Examinatrice

Promotion 2021-2022

- Dédicace -

Je dédie ce modeste travail aux personnes chères à mon coeur.

À *mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,*

À *mes chères sœurs Lynda, Sarah, Lydia et Farah pour leurs encouragements permanents, et leur soutien moral,*

À *mon cher frère Said pour son appui et son encouragement,*

À *toute ma famille pour leur soutien tout au long de mon parcours universitaire,*

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

karima

- Remerciements -

Au terme de notre travail, on remercie Dieu tout puissant de nous avoir donné le courage et la patience pour ce modeste travail.

La réalisation de ce mémoire a été un parcours jalonné de nombreuses rencontres, sans lesquelles ce travail n'aurait pas pu aboutir. On n'aurait pas éprouvé autant de plaisir de réaliser ce travail sans ces personnes, qui par leur générosité, leur disponibilité, leur bonne humeur et l'intérét manifesté à l'égard de notre recherche, ont grandement coutribué à l'amélioration de notre travail.

Je remercie d'abord mon encadrante mademoiselle ***Zohra Aoudia*** *de m'avoir orienté durant l'étaboration de ce travail, elle a toujours été disponible et à l'écoute de mes nombreuses questions.*

Nos sincères remerciements vont également à tous les enseignants du département Rechercher Opérationnelle de l'université ***ABDERRAHMANE MIRA de Bejaia****.*

Enfin, les remerciements s'adressent à toutes nos familles qui sont toujours là pour nous soutenir.Et les personnes qui ont contribué de près ou de loi à la réalisation de ce modesate travail.

karima

Table des matières

Table des figures

INTRODUCTION GÉNÉRALE

Le Machine Learning (ML) est né dans les années 80, et peut être considéré comme une branche de l'intelligence artificielle (IA, dont les débuts remontent à l'après guerre). Le ML (ou apprentissage automatique) est intimement lié à l'analyse de données et aux algorithmes de décision. L'analyse de données tire son origine des statistiques. Jusqu'au 18eme siècle, la statistique, alors « science de l'état » était uniquement descriptive (Desrosières, 2010). Ce n'est qu'un siècle plus tard que les probabilités seront liées aux statistiques, avec entre autre la notion d'extrapolation entre l'observation d'un échantillon et les caractéristiques d'une population. A partir du début du 20ème siècle, les statistiques s'organisent comme une science à part et deux disciplines se distinguent : les statistiques descriptives et les statistiques inférentielles [4]. La notion d'apprentissage automatique fait d'abord référence à la compréhension de la pensée humaine, étudiée par Descartes puis par G. Leibniz dans son ouvrage « De Arte combinatoria » en 1666. Le philosophe tente alors de définir les raisonnements les plus simples de la pensée (à l'image d'un alphabet) qui permettront, une fois combinés de formuler des pensées très complexes [5].Ces travaux seront formalisés par G.Boole en 1854 dans son ouvrage « An investigation of the laws of thought, on which are founded the mathematical of logic and Probability ».

Il est impossible d'évoquer l'origine du machine learning sans parler de celle de (IA). On attribue généralement ses débuts à la création du test de Turing, en 1950. C'est le

mathématicien britannique Alan Turing qui imagine cette épreuve, censée déterminer si une machine peut simuler la pensée humaine. Pour cela, un examinateur est confronté à deux interlocuteurs, l'un étant un ordinateur, l'autre humain. À l'aide d'échanges textuels, il doit alors identifier lequel des deux est une machine. S'il échoue, l'ordinateur a alors passé le test avec succès.

Dans ce contexte, de premiers programmes « intelligents » voient le jour. En 1959, c'est l'informaticien américain Arthur Samuel qui utilise pour la première fois le terme « machine learning », pour son programme créé en 1952. Celui-ci est capable de jouer aux dames et d'apprendre au fur et à mesure de ses parties. Jusqu'à finir par battre le quatrième joueur des États-Unis.

En parallèle, en 1957, un autre informaticien américain, Frank Rosenblatt, crée le « perceptron ». Il s'agit alors d'un classifieur binaire, c'est-à-dire un algorithme capable de classer des éléments (notamment des images) en deux catégories. À cet effet, le programme exploite ce qui constitue le premier réseau de neurones artificiels.

L'objectif du machine learning est de reconnaître parmi des données des structures souvent trop difficiles à détecter ou à mesurer manuellement.

Ce mémoire est organisé de la manière suivante.

Le but de chapitre 1 est d'établir plus clairement ce qui relève ou non du machine learning, ainsi que des branches de ce domaine dont cet mémoire traitera.

Dans le chapitre 2, nous présentons quelques méthode du machine learning, la méthode k plus proches voisins se base sur le principe de « qui se ressemble s'assemble », et utilise les étiquettes des exemples les plus proches pour prendre une décision, Les réseaux de neurones artificiels ne sont au fond rien d'autre que des modèles paramétriques, potentiellement complexes : contrairement à la régression linéaire, ils permettent de construire facilement des modèles très flexibles, Un arbre de décision est une structure arborescente qui représente différents choix possibles et un résultat pour chaque cheminement et machines à vecteurs supports (SVM) sont un algorithme dont le but est de résoudre les problèmes de discrimination à deux classes.

Dans le chapitre 3, nous présentons le Machine Learning appliqué à la finance qu'est

un sujet très important. Dans la mesure où le secteur de la finance collecte un volume élevé de données (big data) recueillies auprès de ses clients, il est parfaitement adapté aux avantages de l'exploration de données.

Dans le chapitre 4, nous présentons les fraudes bancaires ont aujourd'hui un impact financier important et nécessitent d'être détectées au plus vite.

Aujourd'hui, nous utilisons le machine learning dans tous les domaines. Lorsque nous interagissons avec les banques, achetons en ligne ou utilisons les médias sociaux, des algorithmes de machine learning entrent en jeu pour optimiser, fluidifier et sécuriser notre expérience. Le machine learning et la technologie qui l'entoure se développent rapidement, et nous commençons seulement à entrevoir ses capacités.

CHAPITRE 1

PRÉSENTATION DU MACHINE LEARNING

Introduction

Le machine learning est un domaine captivant. Issu de nombreuses disciplines comme les statistiques, l'optimisation, l'algorithmique ou le traitement du signal, c'est un champ d'études en mutation constante qui s'est maintenant imposé dans notre société. Déjà utilisé depuis des décennies dans la reconnaissance automatique de caractères ou les filtres anti-spam, il sert maintenant à protéger contre la fraude bancaire, recommander des livres, films, ou autres produits adaptés à nos goûts, identifier les visages dans le viseur de notre appareil photo, ou traduire automatiquement des textes d'une langue vers une autre. Dans les années à venir, le machine learning nous permettra vraisemblablement d'améliorer la sécurité routière (y compris grâce aux véhicules autonomes), la réponse d'urgence aux catastrophes naturelles, le développement de nouveaux médicaments, ou l'efficacité énergétique de nos bâtiments et industries.

1.1 Machine Learning

Le machine Learning a vu son apparition en 1959 par Le mathématicien américain Arthur Samuel qui d'evloppé un programme qui rèussi à apprendre à joueur au dame sans aides humain. Ce derniera d´efini le Machine Learning suis « Machine Learning is the science of gesting computers to learn Without being explicitly programmed »[18]. En 1998, L'Américain Tom Mitchell donne une autre d´efinition plus avancée en énonçant qu'une machine apprend quand sa performance à faire une certaine tache s'améliore avec de nouvelles expériences. Donc, Le Machine Learning c'est la capacité d'une machine à apprendre quel calcul effectuer pour résoudure un probléme donné.

Définition du Machine Learning dans le dictionnaire français

L'apprentissage automatique est le processus par lequel un algorithme évlue et améliore ses performances sans l'intervention d'un programmeure, en répétant son exécution sur des jeux de donnés jusqu'à obtenir, de manière, déguliére, des résultats pertinents.

♣Exemple

Supposons qu'une entreprise veuille connaître le montant total dépensé par un client ou une cliente à partir de ses factures. Il suffit d'appliquer un algorithme classique, à savoir une simple addition : un algorithme d'apprentissage n'est pas nécessaire. Supposons maintenant que l'on veuille utiliser ces factures pour déterminer quels produits le client est le plus susceptible d'acheter dans un mois. Bien que cela soit vraisemblablement lié, nous n'avons manifestement pas toutes les informations nécessaires pour ce faire. Cependant, si nous disposons de l'historique d'achat d'un grand nombre d'individus, il devient possible d'utiliser un algorithme de machine learning pour qu'il en tire un modèle prédictif nous permettant d'apporter une réponse à notre question [62].

1.1.1 Pourquoi utiliser le machine learning ?

Le machine learning peut servir à résoudre des problèmes

- que l'on ne sait pas résoudre (comme dans l'exemple de la prédiction d'achats ci-dessus) ;
- que l'on sait résoudre, mais dont on ne sait formaliser en termes algorithmiques comment nous les résolvons (c'est le cas par exemple de la reconnaissance d'images ou de la compréhension du langage naturel) ;
- que l'on sait résoudre, mais avec des procédures beaucoup trop gourmandes en ressources informatiques (c'est le cas par exemple de la prédiction d'interactions entre molécules de grande taille, pour les quelles les simulations sont très lourdes).

Le machine learning est donc utilisé quand les données sont abondantes (relativement), mais les connaissances peu accessibles ou peu développées. Ainsi, le machine learning peut aussi aider les humains à apprendre : les modèles créés par des algorithmes d'apprentissage peuvent révéler l'importance relative de certaines informations ou la façon dont elles interagissent entre elles pour résoudre un problème particulier [19]. Dans l'exemple de la prédiction d'achats, comprendre le modèle peut nous permettre d'analyser quelles caractéristiques des achats passés permettent de prédire ceux à venir. Cet aspect du machine learning est très utilisé dans la recherche scientifique :quels gènes sont impliqués dans le développement d'un certain type de tumeur, et comment ? Quelles régions d'une image cérébrale permettent de prédire un comportement ? Quelles caractéristiques d'une molécule en font un bon médicament pour une indication particulière ? Quels aspects d'une image de télescope permettent d'y identifier un objet astronomique particulier ?

1.1.2 les contenus du machine learning

Le machine learning repose sur deux piliers fondamentaux :

- d'une part, les données, qui sont les exemples à partir duquel l'algorithme va apprendre ;
- d'autre part, l'algorithme d'apprentissage,qui est la procédure que l'on fait tourner sur ces données pour produire un modèle. On appelle entraînement le fait de faire tourner un algorithme d'apprentissage sur un jeu de données. Ces deux piliers sont

aussi importants l'un que l'autre. D'une part, aucun algorithme d'apprentissage ne pourra créer un bon modèle à partir de données qui ne sont pas pertinentes

- c'est le concept garbage in, garbage out qui stipule qu'un algorithme d'apprentissage auquel on fournit des données de mauvaise qualité ne pourra rien en faire d'autre que des prédictions de mauvaise qualité. D'autre part, un modèle appris avec un algorithme inadapté sur des données pertinentes ne pourra pas être de bonne qualité. Cet ouvrage est consacré au deuxième de ces piliers

- les algorithmes d'apprentissage. Néanmoins, il ne faut pas négliger qu'une part importante du travail de machine learner ou de data scientist est un travail d'ingénierie consistant à préparer les données afin d'éliminer les données aberrantes,gérer les données manquantes, choisir une représentation pertinente, etc.

Un algorithme d'apprentissage permet donc de modéliser un phénomène à partir d'exemples. Nous considérons ici qu'il faut pour ce faire définir et optimiser un objectif. Il peut par exemple s'agir de minimiser le nombre d'erreurs faites par le modèle sur les exemples d'apprentissage. Cet mémoire présente en effet les algorithmes les plus classiques et les plus populaires sous cette forme [21].

♣Exemple

Voici quelques exemples de reformulation de problèmes de machine learning sous la forme d'un problème d'optimisation. La suite de cet ouvrage devrait vous éclairer sur la formalisation mathématique de ces problèmes, formulés ici très librement.

- un vendeur en ligne peut chercher à modéliser des types représentatifs de clientèle, à partir des transactions passées, en maximisant la proximité entre clients et clientes affectés à un même type ;
- une compagnie automobile peut chercher à modéliser la trajectoire d'un véhicule dans son environnement, à partir d'enregistrements vidéo de voitures, en minimisant le nombre d'accidents ;
- des chercheurs en génétique peuvent vouloir modéliser l'impact d'une mutation sur une maladie, à partir de données patient, en maximisant la cohérence de leur modèle avec les connaissances de l'état de l'art ;

- une banque peut vouloir modéliser les comportements à risque, à partir de son historique,en maximisant le taux de détection de non solvabilité.

Ainsi, le machine learning repose d'une part sur les mathématiques, et en particulier les statistiques, pour ce qui est de la construction de modèles et de leur inférence à partir de données,et d'autre part sur l'informatique, pour ce qui est de la représentation des données et de l'implémentation efficace d'algorithmes d'optimisation. De plus en plus,les quantités de données disponibles imposent de faire appel à des architectures de calcul et de base de données distribuées. C'est un point important mais que nous n'abordons pas dans cet mémoire.

1.1.3 Et l'intelligence artificielle, dans tout ça ?

Le machine learning peut être vu comme une branche de l'intelligence artificielle. En effet, un système incapable d'apprendre peut difficilement être considéré comme intelligent. La capacité à apprendre et à tirer parti de ses expériences est en effet essentielle à un système conçu pour s'adapter à un environnement changeant. L'intelligence artificielle, définie comme l'ensemble des techniques mises en oeuvre afin de construire des machines capables de faire preuve d'un comportement que l'on peut qualifier d'intelligent, fait aussi appel aux sciences cognitives, à la neurobiologie, à la logique, à l'électronique, à l'ingénierie et bien plus encore. Probablement parce que le terme « intelligence artificielle » stimule plus l'imagination, il est cependant de plus en plus souvent employé en lieu et place de celui d'apprentissage automatique.

1.2 Méthode par d'apprentissage

pour qu'une machine arive à apprendre, on doit lui fournir cette capacité qui représente un ensemble de méthode d'apprentissage inspirées de la façon dont nous, les êtres humains, apprenons à faire des choses. Le but de l'apprentissage est d'induire une fonction qui prédise les réponses associées à de nouvelles observation en commettant une erreur de prédiction

la plus faible possible [16]. Parmit ces méthodes, on note : apprentissage superrvisé, non supervisé et par renforcement.

1.2.1 Apprentissage supervisé

Dans l'apprentissage supervisé, l'agent observe quelques couples types entrée-sortie et apprend une fonction de l'entrée vers la sortie [16]. C'est-à-dire, notre échantillon des données est sous forme de couple (X, Y) ou X c'est l'ensemble des aspects qui représente les catégories de la cible Y.

Définition 1.2.1. (**Apprentissage supervisé**)

On appelle apprentissage supervisé la branche du machine learning qui s'intéresse aux problèmes pouvant être formalisés de la façon suivante : étant données n observations $\{\vec{X}^i\}_{1,\dots,n}$ décrites dans un espace $\mathcal{X}$, et leurs étiquettes $\{\vec{Y}^i\}_{1,\dots,n}$ décrites dans un espace $\mathcal{Y}$, on suppose que les étiquettes peuvent être obtenues à partir des observations grâce à une fonction $\phi : \mathcal{X} \to \mathcal{Y}$ fixe et inconnue : $Y^i = \phi(\vec{x}^i) + \epsilon_i$,où ϵ_i est un bruit aléatoire. Il s'agit alors d'utiliser les données pour déterminer une fonction f : $\mathcal{X} \to \mathcal{Y}$ telle que, pour tout couple $(\vec{x}, \phi(\vec{x})) \in \mathcal{X} \times \mathcal{Y}, f(\vec{x}) \approx \phi(\vec{x})$.

L'espace sur lequel sont définies les données est le plus souvent $\mathcal{X} = \mathbb{R}^p$. Nous verrons cependant aussi comment traiter d'autres types de représentations, comme des variables binaires, discrètes, catégoriques, voire des chaînes de caractères ou des graphes [60].

1.2.1.1 Classification binaire

Dans le cas où les étiquettes sont binaires, elles indiquent l'appartenance à une classe. On parle alors de classification binaire.

Définition 1.2.2. (**Classification binaire**)

Un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est binaire, autrement dit $\mathcal{Y} = \{0, 1\}$ est appelé un problème de classification binaire.

♣**Exemple**

Voici quelques exemples de problèmes de classification binaire :

- Identifier si un email est un spam ou non ;
- Identifier si un tableau a été peint par Picasso ou non ;
- Identifier si une image contient ou non une girafe ;
- Identifier si une molécule peut ou non traiter la dépression ;
- Identifier si une transaction financière est frauduleuse ou non.

1.2.1.2 Classification multi-classe

Dans le cas où les étiquettes sont discrètes, et correspondent donc à plusieurs(strictement supérieur à deux) classes, on parle de classification multi-classe.

Définition 1.2.3. (**Classification multi-classe**)

Un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est discret et fini, autrement dit $\mathcal{Y} = \{1, 2, ..., C\}$ est appelé un problème de classification multi-classe.C est le nombre de classes [62].

♣**Exemple**

Voici quelques exemples de problèmes de classification multi-classe :

- Identifier en quelle langue un texte est écrit ;
- Identifier lequel des 10 chiffres arabes est un chiffre manuscrit
- Identifier l'expression d'un visage parmi une liste prédéfinie de possibilités (colère, tristesse, joie,etc.) ;
- Identifier à quelle espèce appartient une plante ;
- Identifier les objets présents sur une photographie.

1.2.1.3 Régression

Dans le cas où les étiquettes sont à valeurs réelles, on parle de régression.

Définition 1.2.4. (**Régression**)

Un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est $\mathcal{Y} = \mathbb{R}$ est appelé un problème de régression [62].

♣Exemple

Voici quelques exemples de problèmes de régression :

- Prédire le nombre de clics sur un lien ;
- Prédire le nombre d'utilisateurs et utilisatrices d'un service en ligne à un moment donné ;
- Prédire le prix d'une action en bourse ;
- Prédire l'affinité de liaison entre deux molécules ;
- Prédire le rendement d'un plant de maïs.

1.2.1.4 Régression structurée

Dans le cas où l'espace des étiquettes est un espace structuré plus complexe que ceux évoqués précédemment, on parle de régression structurée -en anglais, structured regression, ou structured output prediction. Il peut par exemple s'agir de prédire des vecteurs, des images, des graphes, ou des séquences. La régression structurée permet de formaliser de nombreux problèmes, comme ceux de la traduction automatique ou de la reconnaissance vocale (text-to-speech et speech-to-text, par exemple). Ce cas dépasse cependant le cadre du présent ouvrage, et nous nous concentrerons sur les problèmes de classification binaire et multi-classe, ainsi que de régression classique [62].

1.2.2 Apprentissage non-supervisé

On dit que l'apprentissage est non supervisé lorsqu'on ne connait pas les valeurs en sortie et que l'algorithme doit travailler sur l'ensemble des aspects X où il doit reconnaitre les structures communes entre ces derniers pour prédire la cible Y. Dans l'apprentissage non supervisé, l'agent apprend des structures dans les données d'entrée, même s'il ne dispose par de feedback explicite sur ses actions [58]. On peut ainsi regrouper des données dans

des Chusters (c'est le Clustering), détecter des anomalies, ou encore déduire la dimension de données très riches en compilant les dimensions ensembles [59].

Définition 1.2.5. (**Apprentissage non supervisé**)

On appelle apprentissage non supervisé la branche du machine learning qui s'intéresse aux problèmes pouvant être formalisés de la façon suivante : étant données n observations $\{\vec{X}^i\}_{1,\dots,n}$ décrites dans un espace $\mathcal{X}$,il s'agit d'apprendre une fonction sur $\mathcal{X}$ qui vérifie certaines propriétés [63].

1.2.2.1 Clustering

Tout d'abord, le clustering, ou partitionnement, consiste à identifier des groupes dans les données. Cela permet de comprendre leurs caractéristiques générales, et éventuellement d'inférer les propriétés d'une observation en fonction du groupe auquel elle appartient [63].

1.2.3 Apprentissage par renforcement

L'apprentissage par renforcement, c'est apprendre à agir par essais et erreur. Dansce paradigme, un agent peut percevoire son état et effectuer des actions. Après chaque action, une récompense numérique est donnée. Le but de l'agent est de maximiser la récompense totale qu'il reçoit au cours du temps [34].

1.3 Processus de Machine Learning

La figure suivante permet une explication des différentes phases du processus de Machine Learning :

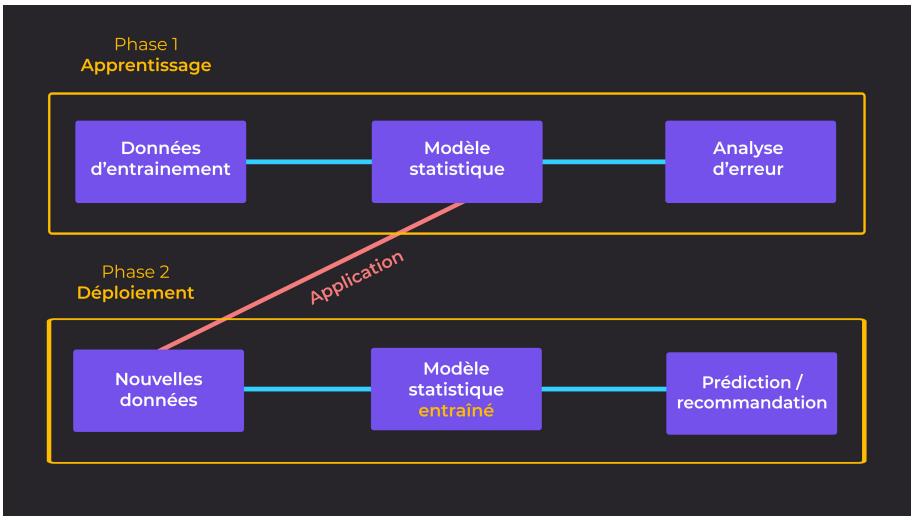


FIGURE 1.1 – -Le processus de Machine Learning [1]

Afin de mieux expliquer le processus de Machine Learning, nous commençons par la définition de quelques concepts de base :

1. **"Dataset"**

   C'est un ensemble de données qu'on fournit à une machine sous forme d'un couple d'exemples (X, Y) dans l'apprentissage supervisé où X représente les questions et Y les réponses au problème que la machione doit résoudre. Dans l'apprentissage non supervisé, le dataset contient que des question X. On peut pas démarrer un projet sans avoir de datasets.

2. **Modèle et ses paramétres**

   C'est une fonction mathmatique qu'on développe à partir du dataset fournit et qui peut être linéaire ou bien non-lineaire. Les coefficients de cette fonction sont les paramètres du modèle et ils sontles prédicteurs X et l'annotation Y que l'on veut généraliser.

3. **Les hyper-paramètres**

   Les hyper-paramètres sont les valeurs de réglage du modèle : le nombre d'itération, les valeures de seed (valeure aléatoire initieles), la solution initiale et les autres paramètres spécifiques des différents modèle testés [18].

4. **Fonction Coût**

   La phase de validation établit la performation du modèle en termes de taux de faux positifs(les fausses alertes)et de faux négatifs (les ratés) que l'on doit réduire simultanément [18]. Ceci se fait par une fonction coût qui représente un ensemble d'erreurs qu'un modèle nous retourne par apport à notre dataset. La validité du modèle dépend de la foction coût à partir de laquelle la machine distingue les notre modèle qui minimisent cette dernière.

5. **La généralisation**

   Consiste à intégrer le modèle dans la processus de big data, et dépense l'horizon méthodoligique concernant l'industrialisation des processus de plus en plus souvent assurée par une distribution du calcul [18]. la processus de Machine Learning ce résume comme suit : Premièremet, on obtient l'échantillon d'entrainement qui représent les prédicateurs X, ce sont les données d'entrainement, et l'annotation Y que l'on veut généraliser (ensemble de dataset). Puis on associe les hyper paramètres à notre modèle. Un entrainement du modèle sera effectué et passera par la phase de validation où l'efficacité du modèle s'étudieraa et un calcule de nombre d'erreurs se réalisera par la fonction coût pour valider le modèle. Sachant qu'un algorithme doit être choisit, une fonction issue d'un ensemble de fonction défini. Au préalable, réalise l'erreurs moyenne la plus faible sur les exemples de la base d'entrainement. Enfin, on passe à la phase de généralisation oùun test du modèle se déroulera sur les données globales.

1.4 Algorithme d'apprentissage

En Machine Learning, le traitement d'un algorithme avec des données nous permet d'avoir un modèle. On peut disting quatre type de modèle selon leur côté supervisé, non supervisé et par renforcement, où chaque type est basé sur un ensemble d'algorithmes.

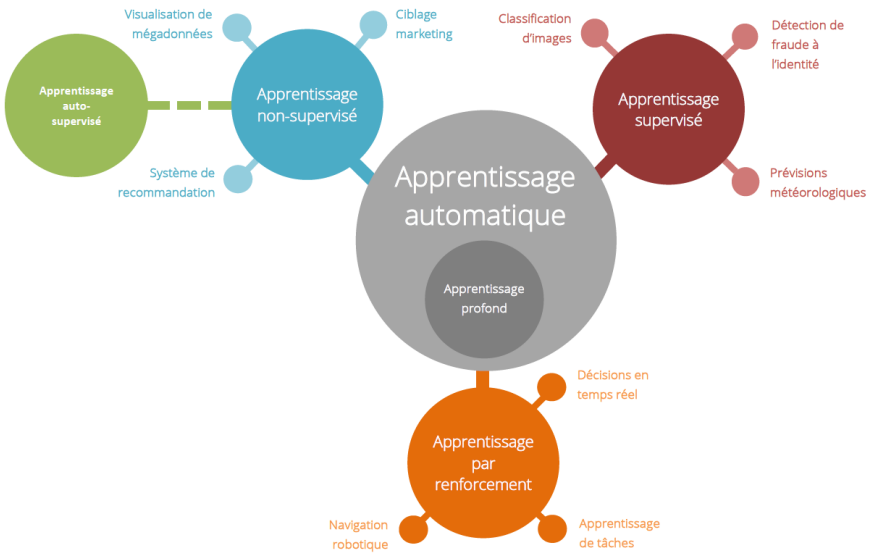


FIGURE 1.2 – Quelques algorithmes des 3 types d'apperntissage du Machine Learning :supervisé ou non supervisé et renforcement [2]

1.5 Régression logistique

La régression logistique est une approche statistique qui peut être employée pour évaluer et caractériser les relation entre une variable réponse de type binaire et une, ou plusieurs, varaibles explicatives. Dans la régression loqistique, ce n'est pas la réponse binaire (malade/pas malade) qui est directement modélisée, mais la probabilité de réalisation d'une des deux modalités (être malade par exemple) [36].

Supposons maintenant que nous voulions résoudre un problème de classification binaire de manière linéaire, c'est-à-dire modéliser $y \in \{0, 1\}$à l'aide d'une combinaison linéaire de variables. Il ne semble pas raisonnable de modéliser directement y comme une combinaison linéaire de plusieurs variables réelles : une telle combinaison est susceptible de prendre non pas deux mais une infinité de valeurs. Nous pourrions alors envisager un modèle probabiliste, dans lequel $\mathbb{P}(Y = y|X = \vec{x})$ soit modélisé par une combinaison linéaire des variables de $\vec{x}$. Cependant, $\mathbb{P}(Y = y|X = \vec{x})$ doit être comprise entre 0 et 1, et intuitivement, cette fonction n'est pas linéaire : si $\mathbb{P}(Y = 0|X = \vec{x})$ est très proche de 1, autrement dit qu'il

est très probable que $\vec{x}$ est négative, une petite perturbation de $\vec{x}$ ne doit pas beaucoup affecter cette probabilité ; mais à l'inverse, si $\mathbb{P}(Y = 0|X = \vec{x})$ est très proche de $0,5$, autrement dit que l'on est très peu certain de l'étiquette de $\vec{x}$, rien ne s'oppose à ce qu'une petite perturbation de $\vec{x}$ n'affecte cette probabilité. C'est pour cela qu'il est classique de modéliser une transformation logit de $\mathbb{P}(Y = y|X = \vec{x})$ comme une combinaison linéaire des variables [43].

1.6 Naïve Bayes

La bayes naif est un classifieur probabiliste se basant sur le théorème de bayes. Il permet de classer des éléments d'aprés leurs caractéristiques, on passant par une phase d'apprentissage. Les modèle sont naif car ils assument que tous les attributs décrivant un élément à chaque à classer sont conditionnellement indépendants.A est conditionnellement de indépendant de B si $P(A|B, C) = P(A|C)$ [55].

Première formule de bayes : A et B, deux événements [55].

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Deuxième formule de bayes : A un événement et B un système complet d'événements[55] :

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{P(A|B_i)P(B_1)+...+P(A|B_n)P(B_n)}$$

Conclusion

Dans ce chapitre, nous avons présenté des généralités sur machine learning, nous avons présenté les différents types d'apprentissage supervisé, non supervisé et par renforcement avec des exemples explicatifs. Dans le chapitre suivant, nous aborderons les méthodes du machine learning avec que nous illustrerons par des exemples d'applications.

CHAPITRE 2

LES MÉTHODES DE MACHINE LEARNING

Introduction

Le Machine Learning est massivement utilisé en Data Science et l'analyse de données (sciences des données). Il permet de développer, de tester et d'appliquer des algorithmes d'analyse prédictive sur différents types de données afin de prédire le futur.

2.1 Méthode des plus proches voisins

La méthode des plus proches voisins a le défaut d'être très sensible au bruit : si une observation est mal étiquetée, ou (ce qui est assez probable en raison d'une part du bruit de mesure et d'autre pert de la nature vraisemblablement incomplète de notre représentation) mal positionnée, tous les points dans sa cellule de Voronoi seront mal étiquetés. Pour rendre cette méthode plus robuste, on se propose de combinerles « opinions »de plusieurs voisins de l'observation que l'on cherche àétiqueter [14].

2.1.1 Méthode des k plus proches voisins

Définition 2.1.1. (**Algorithme des k plus proches voisins**)

Étant donné un jeu $\mathcal{D} = \{(\vec{\mathsf{x}}^i, \mathsf{y}^i)_{i,\dots,n}\}$de n observations étiquetées, une distance d sur$\mathcal{X}$,et un hyperparamètre $k \in \mathbb{N}^*$, on applelle algorithme des k plus proches voisins, ou KNN pour k nearest neighbors, l'algorithme consistant à étiqueter une nouvelle observation $\vec{\mathsf{x}}$ en fonction des étiquettes des k points du jeu d'entraiinement dont elle est la plus proche. En notant $\mathcal{N}_k(\vec{\mathsf{x}})$ l'ensemble des k plus proches voisins de $\vec{\mathsf{x}}$ dans $\mathcal{D}$'[15] :

- pour un problème de classification, on applique le vote dela majorité, et $\vec{\mathsf{x}}$ prend l'étiquette majoritaire parmi celles de ces k plus proches voisins :
$$f(\vec{\mathsf{x}}) = \arg\max_c \sum_{i:\vec{\mathsf{x}}^i \in \mathcal{N}_k(\vec{\mathsf{x}})} \delta(Y^i, c)$$
- pour un problème de régression,$\vec{\mathsf{x}}$ prend comme étiquette la moyenne des étiquettes de ses k plus proches voisins :
$$f(\vec{\mathsf{x}}) = \frac{1}{k} \sum_{i:\vec{\mathsf{x}^i} \in \mathcal{N}_k(\vec{\mathsf{x}})} Y^i.$$

Dans le cas où $k = 1$, on retrouve l'algorithme du plus proche voisin. L'algorithme des k plus proches voisins est un exemple d'apprentissage non paramétrique : la fonction de décision s'exprime en fonction des données observées et non pas comme une formule analytique fonction des variables. On peut rapprocher son fonctionnement de celui d'un raisonnement par cas, qui consiste à agir en se remémorant les choix déja effectuées dans des situation semblables précédemment recontrées, qui se retrouve par exemple lorsqu'un médecin traite un patient en se remémorant comment d'autres patients avec des symptômes similaires ont guéri. La frontiére de décision de l'algorithme des k plus proches voisins est linéaire par morceaux : quand k augmente, elle devient plus « simple », car le vote de la majorité permet de lisser les aspérités crées par les exemples individuels [20].

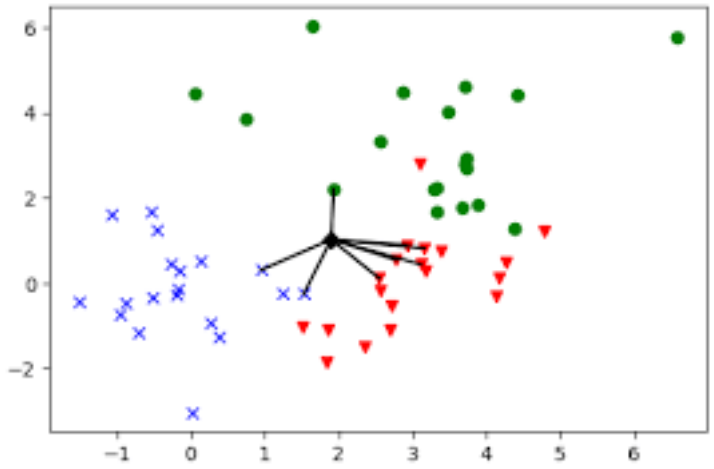


FIGURE 2.1 – Classification avec l'algorithme des k plus proches voisins [68]

2.1.2 Apprentissage paresseux

On parle parfois d'apprentissage paresseux, ou lazy learning en englais, pour qualifier l'algorithme des k plus proches voisins. En effet, la procédure d'apprentissage consiste uniquement à stocker les données du jeu d'entrainement et ne comporte aucun calcul. Attention, cela peut être un facteur limitant, au niveau de la mémoire, si le jeu d'entrainement est très grand.

Á l'inverse, la procédure de prédiction requiert de calculer la distance de l'observation à étiqueter à chaque observation du jeu d'entrainement, ce qui peut être intensif en temps de calcul si ce jeu d'entrainement est très grand. Une prédictionrequiert en effet de calculer n distances, une opération d'une complexité de l'ordre de $\mathcal{O}(np)$ en p dimensions, puis de trouver les k plus petites de ces distance, une opération d'une complexité en $\mathcal{O}(n \log k)$[44].

2.1.3 Nombre de plus proches voisins

Comme nous l'avons décrit ci-dessus, nous proposons d'utiliser $k > 1$ pour rendre l'algorithme des k plus proches voisins plus robuste au bruit. Cependant, à l'inverse, si $k = n$, l'algorithme prédira, dans le cas d'un problème de classification, la classe majoritaire

dans $\mathcal{D}$, et dans le cas d'un problème de régression, la moyenne des étiquettes de $\mathcal{D}$, ce qui parait tout aussi peu satisfaisant.

Il faudra donc choisir une valeur de k intermédiair, ce que l'on fera généralement en utilisant une validation croisée. La valeur $k \approx \sqrt{n}$ est aussi parfois utilisée [56].

2.1.4 Variantes

Définition 2.1.2. $(\varepsilon - Voisins)$

Plutôt que de considérer un nombre fixe de voisins les plus proches, on peut préférer considérer touts les ecemples d'apprentissage suffisamment proches de l'observation à étiqueter : cela permet de mieux utiliser le jeu d'entrainement dans les zones où il est dense. De plus, les prédictions faites en se basant sur des exemples proches (non pas relativement, mais dans l'absolu) sont intuitivement plus fiables que celles faites en se basant sur des exemples d'observations éloignées [69].

2.2 Algorithme des k plus proches voisins

On peut schématiser le fonctionnement de K-NN en l'écrivant en pseudo-code suivant[6] :

Début Algorithme

Données en entrée :

- un ensemble de données D .
- une fonction de définition distance d.
- Un nombre entier K.

Pour une nouvelle observation X dont on veut prédire sa variable de sortie y Faire :

1. Calculer toutes les distances de cette observation X avec les autres observations du jeu de données D.
2. Retenir les K observations du jeu de données D les proches de X en utilisation le fonction de calcul de distance d.
3. Prendre les valeurs de y des K observations retenues :

(a) Si on effectue une régression, calculer la moyenne (ou la médiane) de y retenues.

(b) Si on effectue une classification , calculer le mode de y retenues.

4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation X.

Fin Algorithme

exemple

Reprenons notre exemple de reconnaissance de la langue dans un texte.

Texte	U	H	Langue
1	5,61	0,5	F
2	2,99	5,09	A
3	5,28	0,52	F
4	2,57	5,87	A
5	6,91	0,59	F
6	2,68	5,56	A
7	5,06	0,63	F
8	2,91	5,28	A
9	3,09	4,9	A
10	5,75	0,6	F

Pour chaque donnée du tableau on dispose d'un couple de prédicteurs (fréquence "U",fréquence "H") auquel on peut associer un point dans un repère.

On va donc pouvoir positionner les différentes données dans cet espace (repère dans ce cas - 2 prédicteurs).

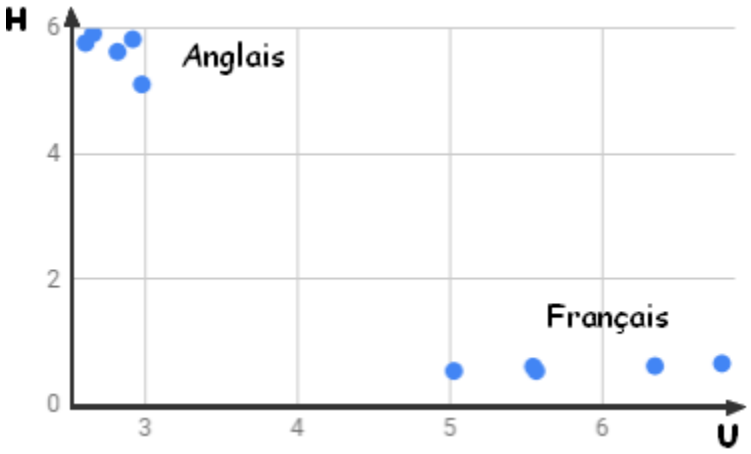


FIGURE 2.2 – Exemple de reconnaissance de la langue dans un texte.

L'algorithme KNN suppose que des objets similaires existent à proximité dans cet espace (plus proches voisins). En d'autres termes, des choses similaires sont proches les unes des autres. Cette notion de proximité peut-être formaliser par un calcul de distance entre des points du graphique. La distance la plus courament utilisée est la distance Euclidienne (il en existe d'autres : distance Manhattan, distance Hamming, ...).

Etant donné un nouveau texte dont on souhaite deviner la langue en fonction de la fréquence d'apparition de la lettre U et de la lettre H, appelons $I = (x_I, y_I)$ le point du repère associé à ce texte inconnu !

Etant donné un texte connu de notre base d'apprentissage, appelons $C = (x_C, y_C)$ le point du repère associé à ce texte.

Nous pouvons calculer la distance entre notre texte inconnu et notre texte connu grâce à la formule :

$$distance(I, C) = \sqrt{(x_C - x_I)^2 + (y_C - y_I)^2}$$

Il faut calculer cette distance entre le point I et tous les points C_j de notre base d'appentissage. Puis sélectionner les k plus proches voisins de I. La valeur de k étant à définir (généralement entre 3 et 5).

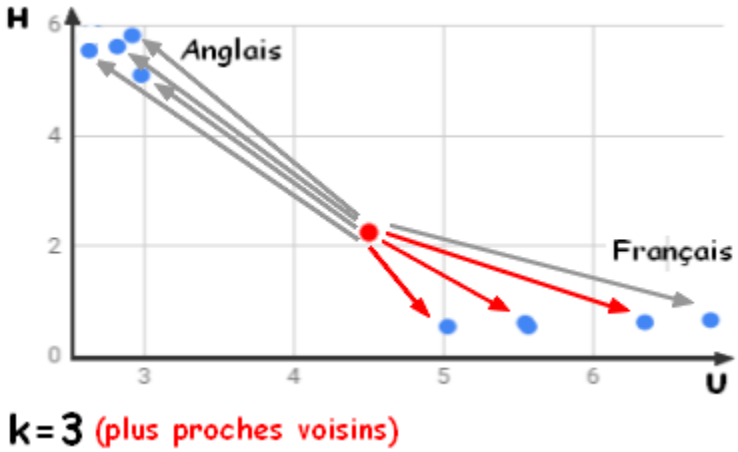


FIGURE 2.3 – Exemple sur la distance entre notre texte inconnu et notre texte connu

Reste à définir l'étiquette de notre inconnu.

- Dans le cas d'une classification on choisira l'étiquette majoritaire parmi les k voisins.
- Dans le cas d'une régression on pourra calculer la moyenne des étiquettes des k voisins[7].

2.3 Réseaux de neurones artificiels

Définition 2.3.1. Les réseaux de neurones artificiels sont des réseaux fortement connectés de processus élémentaires forctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérachique de réseaux est évidemment un réseau [66].

- Propriété des réseaux de neurones.

D'une manière générale un réseaux de neurnoes possède les propriétés suivantes :

1-Parallélisme

Cette notions se situe à la base de l'architecture des réseaux de neurones considérés comme ensembles d'entités élémentaires qui travillent simultanément.

2-Capacité d'adaptation

Celle-ci se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage qui permet au réseaux de tenir compte des nouvelles contraintes ou de nouvelles données du monde extérieur. De plus elle se caractérise dans certains réseaux par leur capacité d'auto-organisation qui assure leur stabilité en tant que systèmes dynamiques.

2.3.1 Modéle de réseaux de neurones artificiels

Le perceptron (**figure 2.2**) est formé d'un couche d'entrée de p neuroes, ou unités, correspondant chacune à une variable d'entrée. Ces neuroes transmettent la valeur de leur entrée à la couche suivante. À ces p neuroes on rajoute généralement une unité de biais, qui transmet toujours la valeur 1. Cette unité correspond à la colonne de 1 que nous avons ajoutée aux données dans les modèles linéaires [35]. On remplacera dans ce qui suit tout vecteur $\vec{x} = (x_1, x_2, ..., x_p)$ par sa version augmentée d'un 1 : $\vec{x} = (1, x_1, x_2, ..., x_p)$. La première et unique couche du perceptron (après la couche d'entrée) contient un seul neurone, auquel sont connectées toutes les unités de la couche d'entrée. Ce neurone calcule une combinaison linéaire $\mathsf{o}(\vec{x}) = w_0 + \sum_{j=1}^{P} w_j x_j$ des signaux $x_1, x_2, ..., x_p$ qu'il reçoit en entrée, auquel il applique une fonction d'activation a, dont il transmet en sortie le résultat. Cette sortie met en œuvre la fonction de décision du perceptron. Ainsi, si l'on appelle w_j le poids de connexion entre l'unité d'entrée j et le neurone de sortie, ce neurone calcule.

$$f(\vec{x}) = a(\mathsf{o}(\vec{x})) = a(w_0 + \sum_{j=1}^{P} w_j x_j) = a((\vec{w}, \vec{x}))$$

Il s'agit donc bien d'un modèle paramétrique.

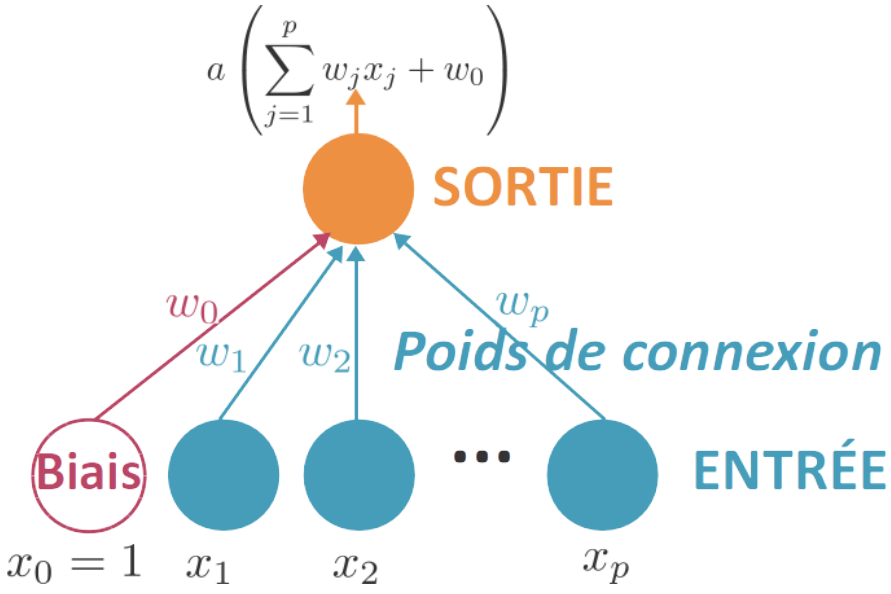


FIGURE 2.4 – Architecture d'un perceptron [3]

2.3.2 Fonctions d'activation

Dans le cas d'un problème de régression, on utilisera tout simplement l'identité comme fonction d'activation. Dans le cas d'un problème de classification binaire, on pourra utiliser[41] :

- Pour prédire directement une étiquette binaire, une fonction de seuil :

$$f : \vec{x} \mapsto \begin{cases} 1 & \text{si} \quad \mathsf{o}(\vec{x}) \leq 0 \\ 0 & \text{sinon} \end{cases}$$

- Pour prédire la probabilité d'appartenir à la classe positive, comme dans le cas de la régression logistique, une fonction logistique :

$$f : \vec{x} \mapsto \frac{1}{e^{\mathsf{o}(\vec{x})}} = \frac{1}{1+exp(\vec{w},\vec{x})}$$

2.3.3 Classification multi-classe

Dans le cas d'un problème de classification multi-classe, on modifiera l'architecture du perceptron de sorte à n'avoir non plus 1 mais C neurones dans la couche de sortie, où C est le nombre de classes. Les $P + 1$ neurones de la couche d'entrée seront ensuite tous

connectés à chacun de ces neurones de sortie (on aura donc $(p+1)C$ poids de connexion, notés w_j^c)[45].

2.3.4 Vitesse d'apprentissage

Cet algorithme a un (hyper)paramètre, $\eta > 0$, qui est le pas de l'algorithme du gradient et que l'on appelle la vitesse d'apprentissage (ou learning rate) dans le contexte des réseaux de neurones artificiels. Cet hyperparamètre joue un rôle important : s'il est trop grand, l'algorithme risque d'osciller autour de la solution optimale,voire de diverger. À l'inverse, s'il est trop faible, l'algorithme va converger très lentement. Il est donc essentiel de bien choisir sa vitesse d'apprentissage. En pratique, on utilise souvent une vitesse d'apprentissage adaptative : relativement grande au début, puis de plus en plus faible au fur et à mesure que l'on se rapproche de la solution. Cette approche est à rapprocher d'algorithmes similaires développés dans le cas général de l'algorithme du gradient (comme par exemple la recherche linéaire par rebroussement) [48].

2.3.5 Classification binaire

Le cas de la classification binaire, utilisant un seuil comme fonction d'activation, est historiquement le premier à a voir été traité. La fonction de coût utilisée est connue sous le nom de critère du perceptron [11] :

$$L(f(\vec{x}^i, y^i)) = \max(0, -y^i o(\vec{x}^i)) = \max(0, -y^i(\vec{w}, \vec{x}))$$

2.3.6 Régression

Dans le cas de la régression, on utilise pour le coût empirique la fonction de coût quadratique [57] :

$$L(f(\vec{x}^i, y^i)) = \tfrac{1}{2}(y^i - f(\vec{x}^i))^2 = \tfrac{1}{2}(y^i - (\vec{w}, \vec{X}))^2$$

2.4 Arbre des décision

Un arbre de décision est une structure arborescente qui représente différents choix possibles et un résultat pour chaque cheminement.

2.4.1 Type Arbre des décision

Il existe deux principaux types d'arbre de décision en fouille de données :

- Les arbres de classification (**Classification Tree**) permettent de prédire à quelle classe la variable-cible appartient, dans ce cas la prédiction est une étiquette de classe.
- Les arbres de régression (**Regression Tree**) permettent de prédire une quantité réelle (par exemple, le prix d'une maison ou la durée de séjour d'un patient dans un hôpital), dans ce cas la prédiction est une valeur numérique.

  Le terme d'analyse d'arbre de classification et de régression (**CART**, d'après l'acronyme anglais) est un terme générique se référant aux procédures décrites précédemment et introduites par Breiman et al. Les arbres utilisés dans le cas de la régression et dans le cas de la classification présentent des similarités mais aussi des différences, en particulier en ce qui concerne la procédure utilisée pour déterminer les séparations des branches [24].

2.4.2 Construction d'un arbre de décision

L'apprentissage par arbre de décision consiste à construire un arbre depuis un ensemble d'apprentissage constitué de n-uplets étiquetés. Un arbre de décision peut être décrit comme un diagramme de flux de données (ou flowchart) où chaque nœud interne décrit un test sur une variable d'apprentissage, chaque branche représente un résultat du test, et chaque feuille contient la valeur de la variable cible (une étiquette de classe pour les arbres de classification, une valeur numérique pour les arbres de régression) [54].

2.4.3 Cas des arbres de classification

Dans le cas des arbres de classification, il s'agit d'un problème de classification automatique. Le critère d'évaluation des partitions caractérise l'homogénéité (ou le gain en homogénéité) des sous-ensembles obtenus par division de l'ensemble. Ces métriques sont appliquées à chaque sous-ensemble candidat et les résultats sont combinés (par exemple, moyennés) pour produire une mesure de la qualité de la séparation [70].

Il existe un grand nombre de critères de ce type, es plus utilisés sont l'entropie de Shannon, l'indice de diversité de Gini et leurs variantes.

- Indice de diversité de Gini : utilisé par l'algorithme **CART**, il mesure avec quelle fréquence un élément aléatoire de l'ensemble serait mal classé si son étiquette était choisie aléatoirement selon la distribution des étiquettes dans le sous-ensemble. L'indice de diversité de Gini peut être calculé en sommant la probabilité pour chaque élément d'être choisi, multipliée par la probabilité qu'il soit mal classé. Il atteint sa valeur minimum (zéro) lorsque tous les éléments de l'ensemble sont dans une même classe de la variable-cible. Pratiquement, si l'on suppose que la classe prend une valeur dans l'ensemble $\{1, 2, ..., m\}$, et si f_i désigne la fraction des éléments de l'ensemble avec l'étiquette i dans l'ensemble, on aura :
$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - f_i^2) = 1 - \sum_{i=1}^{m} f_i^2$$
- Gain d'information : utilisé par les algorithmes **ID3** et **C4.5**, le gain d'information est basé sur le concept d'entropie de Shannon en théorie de l'information [51]. L'entropie permet de mesurer le désordre dans un ensemble de données et est utilisée pour choisir la valeur permettant de maximiser le gain d'information. En utilisant les mêmes notations que pour l'indice de diversité de Gini, on obtient la formule suivante :
$$I_E(f) = \sum_{i=1}^{m} f_i log_2 f_i$$

2.4.4 Cas des arbres de régression

Dans le cas des arbres de régression, le même schéma de séparation peut être appliqué, mais au lieu de minimiser le taux d'erreur de classification, on cherche à maximiser la variance inter-classes (avoir des sous-ensembles dont les valeurs de la variable-cible soient les plus dispersées possibles). En général, le critère utilise le test du chi carré [13].

Remarque

Certains critères permettent de prendre en compte le fait que la variable-cible prend des valeurs ordonnées, en utilisant des mesures ou des heuristiques appropriées [1].

Chaque ensemble de valeurs de la variable de segmentation permet de produire un nœud-fils. Les algorithmes d'apprentissage peuvent différer sur le nombre de nœud-fils produits : certains (tels que CART) produisent systématiquement des arbres binaires, et cherchent donc la partition binaire qui optimise le critère de segmentation. D'autres (comme CHAID) cherchent à effectuer les regroupements les plus pertinents en s'appuyant sur des critères statistiques. Selon la technique, nous obtiendrons des arbres plus ou moins larges. Pour que la méthode soit efficace, il faut éviter de fractionner exagérément les données afin de ne pas produire des groupes d'effectifs trop faibles, ne correspondant à aucune réalité statistique.

2.4.5 Définir la taille de l'arbre

Il n'est pas toujours souhaitable en pratique de construire un arbre dont les feuilles correspondent à des sous-ensembles parfaitement homogènes du point de vue de la variable-cible. En effet, l'apprentissage est réalisé sur un échantillon que l'on espère représentatif d'une population. L'enjeu de toute technique d'apprentissage est d'arriver à saisir l'information utile sur la structure statistique de la population, en excluant les caractéristiques

1. Des heuristiques sont notamment utilisées lorsque l'on cherche à réduire la complexité de l'arbre en agrégeant les modalités des variables utilisées comme prédicteurs de la cible. Par exemple, pour le cas des modalités d'une variable de classes d'âge, on ne va autoriser que des regroupements de classes d'âge contiguës.

spécifiques au jeu de données étudié.Plus le modèle est complexe (plus l'arbre est grand, plus il a de branches, plus il a de feuilles), plus l'on court le risque de voir ce modèle incapable d'être extrapolé à de nouvelles données, c'est-à-dire de rendre compte de la réalité que l'on cherche à appréhender.

En particulier, dans le cas extrême où l'arbre a autant de feuilles qu'il y a d'individus dans la population (d'enregistrements dans le jeu de données), l'arbre ne commet alors aucune erreur sur cet échantillon puisqu'il en épouse toutes les caractéristiques, mais il n'est pas généralisable à un autre échantillon. Ce problème, nommé surapprentissage ou surajustement (overfitting), est un sujet classique de l'apprentissage automatique et de la fouille de données.

On cherche donc à construire un arbre qui soit le plus petit possible en assurant la meilleure performance possible. Plus un arbre sera petit, plus il sera stable dans ses prévisions futures. Il faut réaliser un arbitrage entre performance et complexité dans les modèles utilisés. À performance comparable, on préférera toujours le modèle le plus simple, si l'on souhaite pouvoir utiliser ce modèle sur de nouveaux échantillons [13].

2.5 Machines à vecteurs de support et méthodes à noyaux

Les machines à vecteurs de support (aussi appelées machines à vecteurs supports), ou SVM de l'anglais support vector machines, sont de puissants algorithmes d'apprentissage automatique. Elles se basent sur un algorithme linéaire proposé par Vladimir Vapnik et Aleksandr Lerner en 1963 (Vapnik et Lerner,1963), mais permettent d'apprendre bien plus que des modèles linéaires. En effet, au début des années 1990, Vladimir Vapnik, Bernhard Boser, Isabelle Guyon et Corinna Cortes (Boser et al., 1992 ; Cortes et Vapnik, 1995) ont trouvé comment les étendre efficacement à l'apprentissage de modèles non linéaires grâce à l'astuce du noyau. Ce chapitre présente cette approche dans ses différentes versions pour un problème de classification, et introduit ainsi la famille des méthodes à noyaux [17].

2.5.1 Le cas linéairement séparable : SVM à marge rigide

Dans cette section, nous allons supposer qu'il est possible de trouver un modèle linéaire qui ne fasse pas d'erreur sur nos données : c'est ce qu'on appelle un scénario linéairement séparable.

Définition 2.5.1. (**Séparabilité linéaire**)

Soit $\mathcal{D} = \{(\vec{x}^i, y^i)_{i,\dots,n}\}$ un jeu de données de n observations. Nous supposons que $\vec{x}^i \in \mathbb{R}^p et y^i \in \{0, 1\}$. On dit que $\mathcal{D}$ est linéairement séparable s'il existe au moins un hyperplan dans $\mathbb{R}^p$ tel que tous les points positifs (étiquetés $+1$) soient d'un côté de cet hyperplan et tous les points négatifs (étiquetés -1) de l'autre. Dans ce cas, il existe en fait une infinité d'hyperplans séparateurs qui ne font aucune erreur de classification (voir figure 2.3). Ces hyperplans sont des modèles équivalents du point de vue de la minimisation du risque empirique [22].

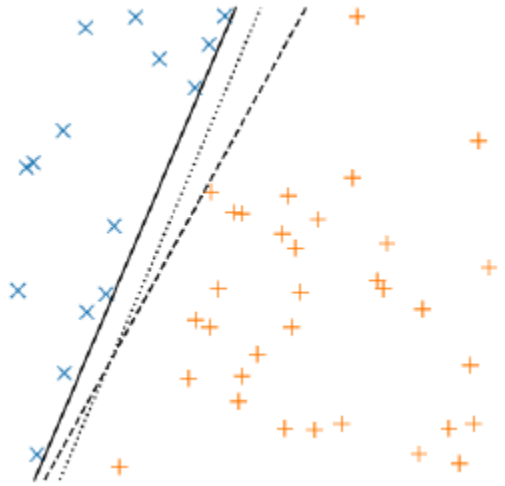


FIGURE 2.5 – Une infinité d'hyperplans (en deux dimensions, des droites) séparent les points négatifs(x) des points positifs($+$) [67].

2.5.2 Marge d'un hyperplan séparateur

En l'absence d'information supplémentaire, l'hyperplan séparateur en pointillés sur la figure 2.3 semble préférable. En effet, celui-ci, étant équidistant de l'observation positive la plus proche et de l'observation négative la plus proche, coupe en quelque sorte la poire en deux pour la région située « entre » les points positifs et les points négatifs. La marge d'un hyperplan séparateur permet de formaliser cette intuition [25].

Définition 2.5.2. (**Marge**)

La marge γ d'un hyperplan séparateur est la distance de cet hyperplan à l'observation du jeu d'entraînement la plus proche.

L'hyperplan séparateur que nous cherchons est donc celui qui maximise la marge. Il y a alors au moins une observation négative et une observation positive qui sont à une distance γ de l'hyperplan séparateur :

dans le cas contraire, si par exemple toutes les observations négatives étaient à une distance supérieure à γ de l'hyperplan séparateur, on pourrait rapprocher cet hyperplan des observations négatives et augmenter la marge.

Nous pouvons alors définir, en plus de l'hyperplan séparateur H, les hyperplans H_+ et H_- qui lui sont parallèles et situés à une distance γ de part et d'autre. H_+ contient au moins une observation positive, tandis que H_- contient au moins une observation négative[27].

Définition 2.5.3. (**Vecteurs de support**)

On appelle vecteurs de support les observations du jeu d'entraînement situés à une distance de l'hyperplan séparateur. Elles « soutiennent » les hyperplans H_+ et H_-.

C'est de là que vient le nom de la méthode, appelée Support Vector Machine ou SVM en anglais, et machine à vecteurs de support en français. On rencontre aussi parfois le nom de « séparatrice à vaste marge », qui respecte les initiales SVM.

Si l'on venait à déplacer légèrement une observation qui est vecteur de support, cela déplacerait la zone d'indécision et l'hyperplan séparateur changerait. À l'inverse, si l'on

déplace légèrement une observation qui n'est pas vecteur de support, H n'est pas affecté : les vecteurs de support sont les observations qui soutiennent la solution.

Toutes les observations positives sont situées à l'extérieur de H_+, tandis que toutes les observations négatives sont situées à l'extérieur de H_- [28].

Définition 2.5.4. (**Zone d'indécision**)

On appelle zone d'indécision la zone située entre H_- et H_+. Cette zone ne contient aucune observation [31].

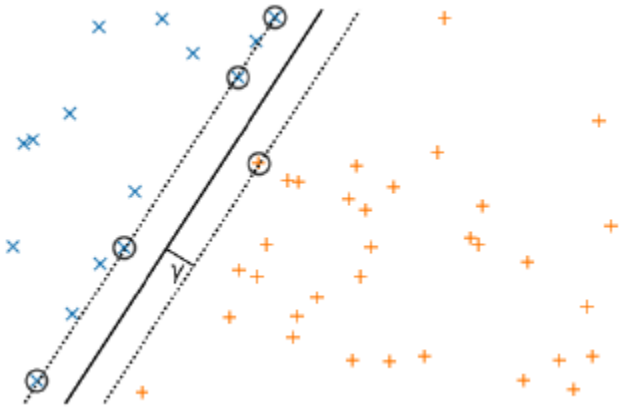


FIGURE 2.6 – illustre ces concepts [65].

Figure 2.4– La marge γ d'un hyperplan séparateur (ici en trait plein) est sa distance à l'observation la plus proche. Quand cette marge est maximale, au moins une observation négative et une observation positive sont à une distance γ de l'hyperplan séparateur. Les hyperplans (ici en pointillés) parallèles à l'hyperplan séparateur et passant par ces observations définissent la zone d'indécision. Les observations situées sur ces hyperplans (cerclées) sont les vecteurs de support.

2.5.3 Le cas linéairement non séparable : SVM à marge souple

Nos données ne sont généralement pas linéairement séparables. Dans ce cas, quel que soit l'hyperplan séparateur que l'on choisisse, certains des points seront mal classifiés ;

d'autres seront correctement classifiés, mais à l'intérieur de la zone d'indécision [40]. Ces concepts sont illustrés sur la figure 2.5.

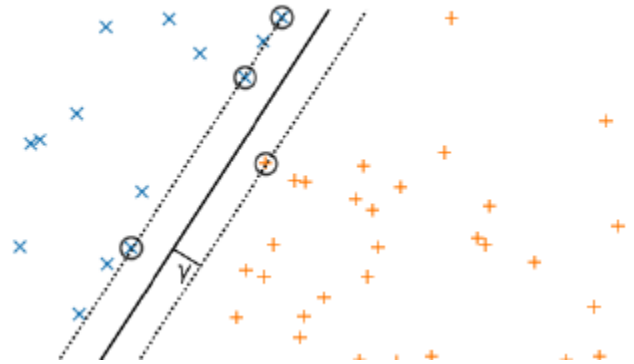


FIGURE 2.7 – Aucun classifieur linéaire ne peut séparer parfaitement ces données. Les observations marquées d'un carré sont des erreurs de classification. L'observation marquée d'un triangle est correctement classifiée mais est située à l'intérieur de la zone d'indécision. Si elle était à sa frontière, autrement dit, si elle était vecteur de support, la marge serait beaucoup plus étroite [65].

2.5.4 Le cas non linéaire : SVM à noyau

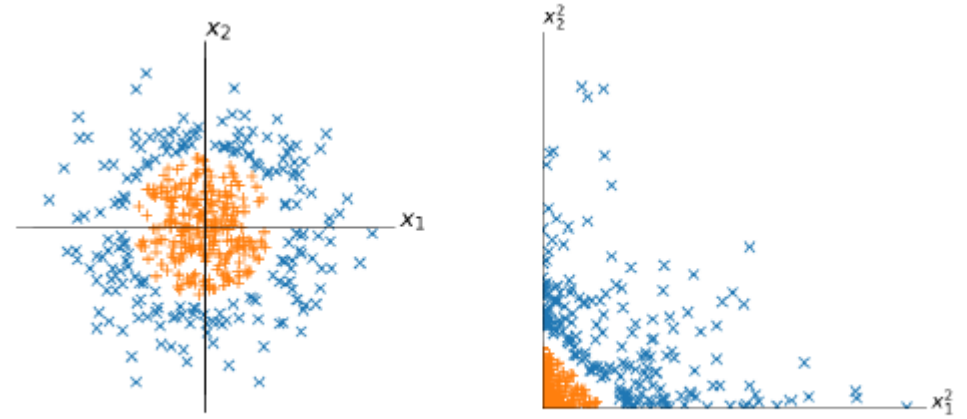
Il est fréquent qu'une fonction linéaire ne soit pas appropriée pour séparer nos données (voir par exemple la figure 2.6). Que faire dans ce cas ?

2.5.5 Espace de redescription

Dans le cas des données représentées sur la figure 2.6, un cercle d'équation $x_{21}+x_{22} = R^2$ semble bien mieux indiqué qu'une droite pour séparer les deux classes. Or si la fonction $f : \mathbb{R}^2 \to R, \vec{x} = x_1^2 + x_2^2 - R^2$ n'est pas linéaire en $\vec{x} = (x_1, x_2)$, elle est linéaire en (x_1^2, x_2^2) : Définissons donc l'application $\phi : \mathbb{R}^2 \to \mathbb{R}^2, (x_1, x_2) \mapsto (x_1^2, x_2^2)$:La fonction de décision f est linéaire en $\phi(\vec{x})$: $f(\vec{x}) = \phi(\vec{x})_1 + \phi(\vec{x})_2 - R^2$. Nous pouvons donc l'apprendre en utilisant une SVM sur les images des données par l'application ϕ.

Plus généralement, nous allons maintenant supposer que les observations sont définies

sur un espace quelconque X, qui peut être $\mathbb{R}^p$ mais aussi par exemple l'ensemble des chaînes de caractères sur un alphabet donné, l'espace de tous les graphes, ou un espace de fonctions[42].



(A) Un cercle semble bien mieux indiqué qu'une droite pour séparer ces données.

(B) Après transformation par l'application ϕ : $(x_1, x_2) \mapsto (x_1^2, x_2^2)$, les données sont linéairement séparables dans l'espace de redescription.

FIGURE 2.8 – Transformer les données permet de les séparer linéairement dans un espace de redescription [67].

2.5.6 Noyau

Dans l'espace de redescription, les images des observations dans $\mathcal{H}$ apparaissent uniquement dans des produits scalaires sur $\mathcal{H}$. Nous pouvons donc, plutôt que la fonction $\phi : \mathcal{X} \to \mathcal{H}$, utiliser la fonction suivante, appelée noyau[52] :

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$\vec{x}, \vec{x}^i \mapsto (\phi(\vec{x}), \phi(\vec{x}^i))_{\mathcal{H}}$$

2.5.7 Noyaux

Caractérisation mathématique

Définition 2.5.5. (**Noyau**)

Nous appelons noyau toute fonction k de deux variables s'écrivant sous la forme d'un produit scalaire des images dans un espace de Hilbert de ses variables [61]. Ainsi, un noyau est une fonction continue, symétrique, et semi-définie positive :

$$\forall N \in \mathbb{N}, \forall (\vec{x}^1, \vec{x}^2, ..., \vec{x}^N) \in \mathcal{X}^N \text{ et } (a_1, a_2, ..., a_N) \in \mathbb{R}^N, \sum_{i=1}^{N} \sum_{l=1}^{N} a_i a_l k(\vec{x}^i, \vec{x}^l) \geqslant 0.$$

2.6 Algorithme de support vector machines

- Ce sont des algorithmes d'apprentissage initialement construits pour la classification binaire.
- L'idée est de rechercher une règle de décision basée sur une séparation par hyperplan de marge optimale.
- Méthode relativement récente qui découle de premiers travaux théoriques de Vapnik et Chervonenkis en 1995, d émocratisés à partir de 2000.
- Le principe de l'algorithme est d'intégrer lors de la phasé d'apprentissage une estimation de sa complexité pour limiter le phénomène d'over-fitting.
- Méthode qui ne substitue pas au problème dèjà compliqué de la classification un problème encore plus complexe comme l'estimation d'une densité de probabilités (par exemple).

L'algorithme se base principalement sur 3 astuces pour obtenir de tres bonnes performances tant en qualitè de prédiction qu'en complexite de calcul.

- On cherche l'hyperplan comme solution d'un problème d'optimisation sous-contraintes. La fonction à optimiser intègre un terme de qualité de prédiction et un terme de complexité du modèle.
- Le passage à la recherche de surfaces séparatrices non linéaires est introduit en utilisant un noyau kernel qui code une transformation non linéaire des données.
- Numeriquement, toutes les équations s'obtiennent en fonction de certains produits scalaires utilisant le noyau et certains points de la base de données (ce sont les

Support Vectors) [8].

Description de l'algorithme SVM

Soit X_t un exemple d'entrainement, $X = \{X^t, Y^t\}$ où $Y^t = \begin{cases} +1 & \text{si} \quad x_t \in c_1 \\ -1 & \text{si} \quad x_t \in c_2 \end{cases}$

la distance entre cet exemple et le plan de paramètres w et $w_0$est :

$\textbf{dist}(x_t; w, w_0) = \frac{|w_0 + w^t x^t|}{\|w\|}$.

La marge correspond donc à :

$\textbf{marge}(w, w_0) = \min_t dist(x_t; w, w_0)$.

On trouve L'hyperplan séparateur optimal en cherchant w, w_0 tel que [12] :

$$Y^t(w_0 + w^t x^t) \geq +1.$$

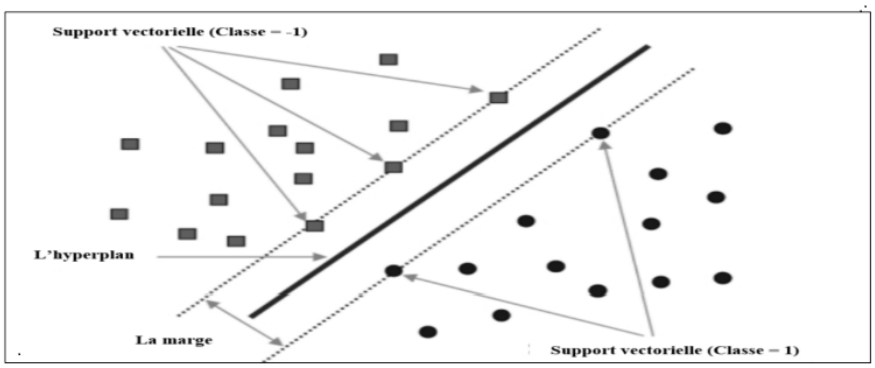


FIGURE 2.9 – Exemple d'un SVM avec un noyau linéaire [12]

2.7 Outils informatiques pour le Machine Learning

Il n'y a jamais de langage meilleur que les autres pour le machine learning, tout dépend de l'utilisation que l'on en fait. Les experts de machine learning travaillant sur l'analyse des sentiments privilégieront Python et *R*. En revanche, Java sera plus utilisé par ceux qui travaillent sur la sécurité réseau/cyber-attaques et la détection des fraudes, les deux domaines où Python sera moins prioritaire [9].

En dehors de l'analyse de sentiment, R est également une bonne solution pour les domaines biologiques/recherche – en bio-ingénierie et bio-informatique.

Faciles à utiliser, les nombreux algorithmes de machine learning restent difficiles à implémenter. Ils sont donc aujourd'hui packagés dans des bibliothèques logicielles fournies par de grandes communautés de programmeurs :

- NumPy pour le traitement de tableaux multidimensionnels,
- SciPy pour l'algèbre linéaire et l'informatique scientifique,
- Matplotlib pour la visualisation,
- Pandas pour les données chronologiques (sachant que la plupart des données en finance proviennent de séries chronologiques),
- Keras pour les réseaux neuronaux, etc.

L'acquisition de données, l'entraînement de modèles, de déploiement de réseaux de neurones requièrent à l'origine d'importantes compétences techniques.

Parmi ces frameworks, on compte notamment PyTorch, CNTK, MXNet, et Google Tensorflow, l'un des outils les plus utilisés en IA pour développer et exécuter des applications de machine learning et de deep learning.

Conclusion

Le machine learning, ou apprentissage statistique, est un domaine de la modélisation statistique et de l'intelligence artificielle. L'objectif du machine learning est de reconnaître parmi des données des structures souvent trop difficiles à détecter ou à mesurer manuellement.

CHAPITRE 3

QUELQUES APPLICATIONS DE MACHINE LEARNING EN FINANCES

Introduction

Les avancées technologiques de ces 20 dernières années, telles que la blockchain ou le Big Data, sont à l'origine de nombreux bouleversements humains mais aussi sociétaux. L'entrée dans un monde de plus en plus numérique a décuplé la création de données et l'exploitation de celles-ci à grande échelle. Cette exploitation s'appuie sur des théories mathématiques et statistiques dont les possibilités de mise en oeuvre s'accélèrent avec les progrès en matière de calcul. Parmi les disciplines rattachées à la science de la donnée, le Machine Learning connaît une expansion rapide. Ce focus se destine à un public non averti désireux d'en savoir plus sur les usages actuels du Machine Learning au sein des services financiers.

3.1 la gestion d'actifs

3.1.1 Application du Machine Learning à l'allocation d'actifs

La difficulté pour le client va être de choisir un type de fonds pour investir et les héberger dans son produit d'épargne. Les combinaisons sont multiples et le client a besoin généralement de l'expertise de son conseiller pour l'accompagner dans ses choix. Car, bien en amont des rendements espérés par un épargnant, il existe un véritable travail d'analyse réalisé par les sociétés de gestion dotées de nombreuses compétences : les gérants de portefeuilles, les analystes quantitatifs ou les risk managers, dont les expertises servent in fine la performance servie aux clients en contrepartie de frais de gestion (en moyenne $3,47\%$ en 2018). Les gérants de portefeuille, aidés par les analystes quantitatifs, doivent déterminer la meilleure allocation possible pour les fonds sous gestion. L'allocation d'actifs consiste en la définition d'une stratégie d'investissement en fournissant un portefeuille modèle (soit la définition d'une stratégie de répartition entre plusieurs catégories d'actifs). Le recours au Machine Learning prend ici tout son sens avec une suggestion personnalisée en fonction de l'aversion au risque du client, totalement automatisée et venant remplacer ou seconder le conseiller en patrimoine. Ce type de gestion de fonds pilotée permet de ne plus faire face aux aléas humains et de recourir à des algorithmes présentant une couverture large et à même de répondre aux attendus des investisseurs. Ils ont par ailleurs l'avantage d'être moins coûteux qu'une gestion humaine. La répartition des fonds ainsi que leur évolution dans le temps seront observées et travaillées par la machine qui pourra avoir, dans certains cas, un rôle totalement autonome, ou semi-autonome (de l'ordre de la suggestion et non de l'action). L'exposition du portefeuille est calculée en fonction de la classe d'actifs, du pays ou encore par secteur d'activité. Un portefeuille modèle pourrait ainsi avoir cette apparence :

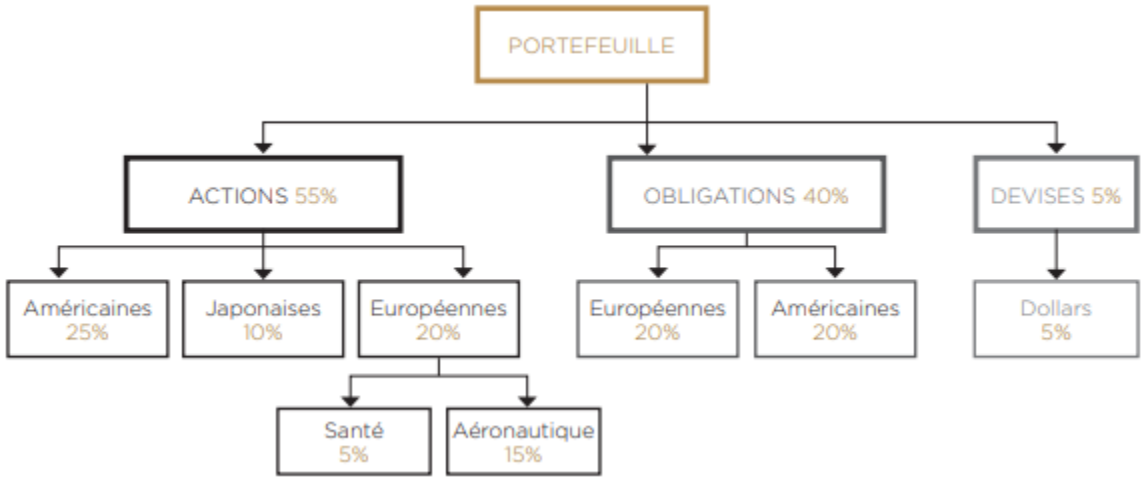


FIGURE 3.1 – Exemple de portefeuille de gestion [26]

Une stratégie d'allocation va être généralement décomposée en plusieurs phases :

- Déterminer les classes d'actifs (exemple : 55% d'actions, 40% d'obligations et 5% de devises étrangères).
- Identifier les localisations (il est ici préférable d'avoir des actions américaines plutôt que des actions japonaises ou européennes).
- Choisir la répartition sectorielle au sein des classes d'actifs, des instruments financiers et des émetteurs susceptibles de générer de la performance (exemple : santé, aéronautique). Évidemment tout l'enjeu de la gestion de portefeuille ou du trading est de déterminer à l'avance, la classe d'actifs ou le secteur qui va le plus performer dans les jours, les mois ou les années à venir. Le Machine Learning peut ainsi devenir un véritable a tout pour essayer de prévoir l'évolution des marchés financiers [26].

3.1.2 Application du Machine Learning à la gestion de portefeuille

En gestion d'actifs, le Machine Learning s'avère d'une grande utilité pour prédire le cours des actifs. Il faut ainsi voir les marchés financiers comme une immense toile d'araignée. Tous les actifs sont liés entre eux et le moindre mouvement d'un actif a des répercussions sur un certain nombre d'autres actifs. C'est ce qu'on appelle la corrélation. Les actifs peuvent

ainsi être corrélés positivement ou négativement. Les actifs d'un même secteur et/ou d'une même région sont souvent corrélés positivement [26].

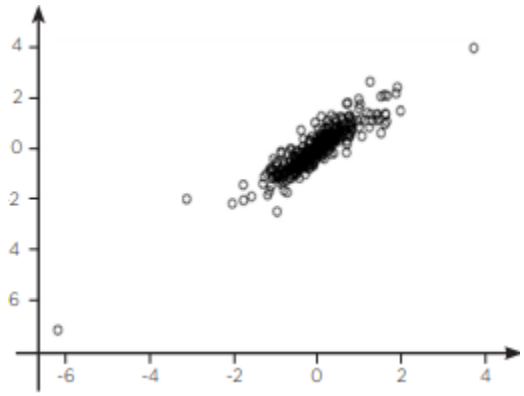


FIGURE 3.2 – Corrélation linéaire entre les actions Société Générale et BNP Paribas [26]

Prenons l'exemple de deux banques françaises, la Société Générale et BNP Paribas. Les données utilisées pour observer la corrélation sont le prix des actions du 01/01/2016 au 01/06/2017. Les points ont tendance à former une ligne, permettant donc de déduire que ces deux actifs sont corrélés positivement. Cela se confirme, par ailleurs, par un coefficient de corrélation égal à $0,911$.

De nombreuses données possibles

Il existe de nombreuses données possibles pour essayer de déterminer le cours futur d'un actif. Ces données peuvent être numériques, comme des séries de prix d'actions, de taux et d'autres indicateurs macroéconomiques. Elles peuvent également être issues de sources plus diverses : articles de presse, états financiers, publications sur les réseaux sociaux et annonces officielles grâce à l'analyse de texte, mais aussi, de manière plus large, au travers des résultats de navigation sur internet ou d'images. En effet les prix des actions sont, à titre d'exemple, très sensibles à l'image de l'entreprise d'où l'importance des réseaux sociaux.

Utilisation d'un arbre de décisions

L'arbre de décisions est l'une des méthodes les plus utilisées pour la prévision de rendement d'actifs financiers. Ci-dessous l'exemple d'Airbus. La question est de savoir s'il est

pertinent d'investir dans des actions Airbus, compte tenu des informations suivantes :

- Des indices boursiers : le CAC40, le Dow Jones et l'Euro Stoxx 50 ;
- Des compagnies aéronautiques : Boeing, Bombardier ;
- Une compagnie aérienne : Air France. Le jeu de données sera constitué de données journalières, comprises entre le 01/12/2003 et le 31/05/2019. Ces données sont séparées volontairement en deux parties, l'une pour déterminer le modèle (par exemple 90% des données) et l'autre pour l'évaluer (les 10% restant). En utilisant l'algorithme C5.0 sur R^5, nous obtenons l'arbre de décisions suivant :

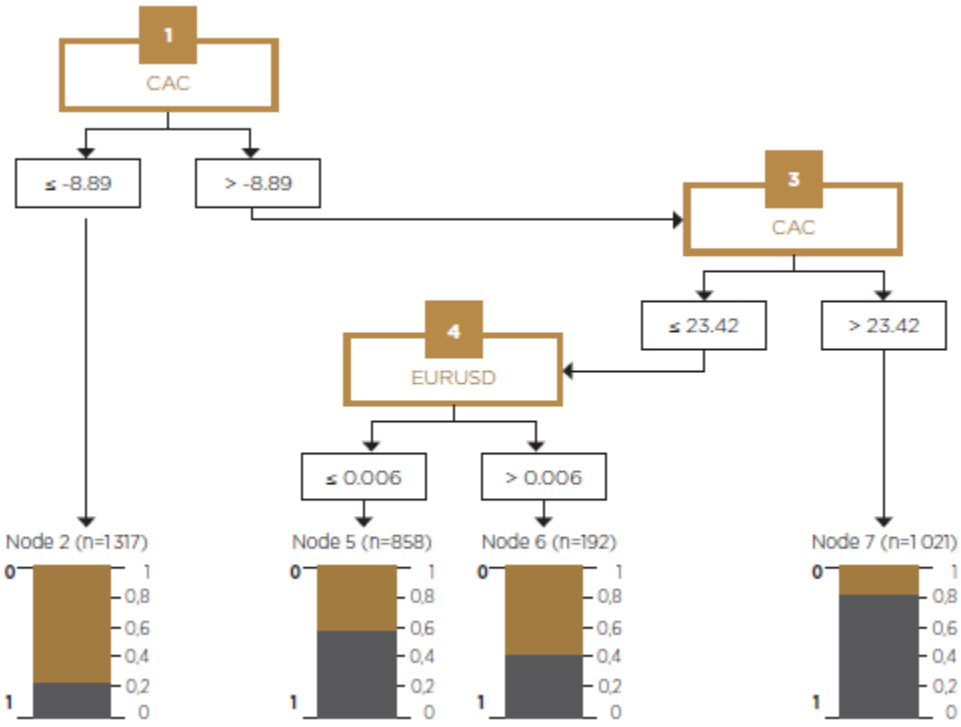


FIGURE 3.3 – Arbre de décisions – Rendement Airbus [26]

Le chiffre « 1 » correspond à un rendement positif et donc à un achat tandis que le « 0 » a un rendement négatif. Le modèle se base uniquement sur les rendements du CAC40 et du taux de change EUR/USD pour prédire le rendement de l'action d'Airbus. Ainsi si le rendement du CAC40 est inférieur à -8,89, le rendement de l'action d'Airbus a de fortes chances d'être négatif. Au contraire s'il est supérieur à 23,42, l'action d'Airbus aura sans

doute un rendement positif. Entre les deux, il faut utiliser le rendement du taux de change EUR/USD pour prédire celui de l'action d'Airbus. En utilisant les données de test, il est alors possible de générer le tableau de validation croisée suivant :

	Prédiction		
Réalité	0 (rendement négatif)	1 (rendement positif)	Total en ligne
0 (rendement négatif)	131 (soit 34,7% de l'échantillon)	35 (soit 9,3% de l'échantillon)	166
1 (rendement positif)	72 (soit 19,1% de l'échantillon)	139 (soit 36,9% de l'échantillon)	211
Total en colonne	203	174	377

FIGURE 3.4 – Tableau de validation croisée – Rendement Airbus [26]

Le modèle est assez fiable lorsqu'il prédit que le rendement de l'action Airbus sera positif, évitant ainsi tout investissement à perte. En revanche, il existe un risque de passer à côté de bonnes opportunités : en effet dans 19% des cas, le modèle prédit une chute du cours de l'action, alors que finalement celui-ci monte. Il est évidemment nécessaire d'être en possession de beaucoup plus de données, que ce soit en nombre de facteurs explicatifs ou de volume d'observations, pour avoir un modèle performant. Utiliser les arbres de décisions devient une véritable aide pour les gérants d'actifs. Tout en gardant leurs convictions personnelles, ils peuvent savoir ce qui est statistiquement le plus probable. Cependant, les rendements étant corrélés aux risques, il s'agit alors d'être capable de les mesurer pour mieux s'en protéger.

3.2 le risque de crédit : Notation et Scoring

La constitution d'un portefeuille de marché est un exercice complexe. Le recours au Machine Learning représente un atout d'envergure afin de pouvoir identifier au mieux le risque de crédit associé. Si l'utilité de ce procédé a été démontrée ci-avant, il s'avère que les éléments analysés reposent sur des informations fournies telles que les notations de titres corrélées à des situations de marché. Cependant, si les notations fournies par l'ensemble des grandes agences sont utilisées, il est également intéressant d'observer comment ces

dernières sont déterminées, et surtout, comment elles vont être impactées par le Machine Learning.

3.2.1 La dette souveraine

Les dettes souveraines, matérialisées par des obligations, représentent une catégorie de produits financiers prisés des investisseurs et autres gérants de fonds. Entre 2014 et 2019, l'encours obligataire de la dette mondiale a été multiplié par $2,5$ pour atteindre les 250000 milliards de dollars. La notation des obligations par les principales agences de notations (telles que Moody's, Fitch et $S\&P$) devient donc primordiale pour lesinvestisseurs en quête d'informations qualitatives. Le processus de notation est cependant une analyse longue et complexe. Le recours à l'intelligence artificielle et aux algorithmes de Machine Learning vient ici simplifier mais également affiner le travail des analystes (sans pour autant le remplacer). Ainsi, afin de prédire les notations des dettes souveraines (exemple : AA pour la dette française), plusieurs critères vont être observés pour décrire et analyser la capacité des emprunteurs souverains à tenir leurs engagements [26].

Voici à titre d'exemple quelques indicateursanalysés :

- Indicateur 1 : Dette externe / Exports
- Indicateur 2 : Balance externe
- Indicateur 3 : Taux d'inflation
- Indicateur 4 : Equilibre fiscal La liste globale des indicateurs observés va constituer le jeu de données à analyser. L'avantage de ces derniers est qu'ils sont reconnus comme fiables et reposent généralement sur des mesures économiques solides. Dans le cadre des notations souveraines, le modèle aujourd'hui privilégié est le modèle ACRP (Automated Credit Rating Prediction), lui-même basé sur le modèle du réseau de neurones[1] artificiels. Concrètement, il va analyser les différentes relations entre les variables explicatives (les indicateurs mentionnés ci-avant) et la variable à expliquer

1. Unité effectuant un traitement sur les données qu'il reçoit. Charge aux neurones d'envoyer ou non le résultat de son traitement aux neurones suivants.

(la note de la dette) pour : es indicateurs mentionnés ci-avant) et la variable à expliquer (la note de la dette) pour : es indicateurs mentionnés ci-avant) et la variable à expliquer (la note de la dette) pour :

- Classifier (soit analyser un grand nombre d'individus, ici nos dettes souveraines, que l'on cherche à répartir en catégories, nos notations) ;
- Effectuer une régression (soit une modélisation établissant une ou des estimations futures à partir de données issues du passé). Au travers du modèle généré, il est alors possible de déterminer une note qui sera alors attribuée à une dette souveraine.

  Une étude coréalisée par le Luxembourg Institute of Science and Technology et l'Université de Saarland (Allemagne) a ainsi pu fournir les pourcentages de précision dudit modèle pour ces deux formes :

Critères de performance	Réseaux neuronaux (ACRP) Classification	Réseaux neuronaux (ACRP) Régression
Notes souveraines correctement classifiées (en %)	40,4	34,6
Notes souveraines correctement classifiées (en %) avec un degré d'écart toléré	63,6	68,9
Notes souveraines correctement classifiées (en %) avec deux degrés d'écart tolérés	80,4	87,3
Notes souveraines correctement classifiées (en %) avec trois degrés d'écart tolérés	87,6	96,7

FIGURE 3.5 – Comparaison de deux formes du modèle ACRP [26]

La qualité des résultats est variable. En effet, le modèle, tant en classification qu'en régression, ne donnera la note précise que dans $40,4\%$ et $34,6\%$ des cas, ce qui est bon mais pas suffisant. Cependant, dès lors que l'on accorde une certaine marge d'erreur avec un ou plusieurs degrés tolérés dans la notation (un degré étant l'écart séparant une notation AAA d'une notation $AA+$), les résultats s'améliorent sensiblement (près de 9 cas sur 10 bien classés à 2 échelons près).

Le modèle de réseaux neuronaux se révèle d'une très grande efficience avec une note proche de la note réelle, en faisant un excellent indicateur pour les analystes qui viendront, dans certains cas, affiner le résultat. Cette forte efficacité fait du modèle de réseaux neu-

ronaux un modèle de référence pour cette typologie d'émetteurs de dette, cependant cela s'applique-t-il à d'autres typologies ?

3.2.2 La dette corporate

L'analyse d'une dette d'Etat diffère totalement de celle d'une dette corporate. En effet, si les deux se basent sur des ratios, ces derniers sont bien différents, de même que les indicateurs externes pouvant présenter un impact sur leurs résultats. Afin de pouvoir donner une indication fiable et indépendante, les entreprises du monde entier vont solliciter et payer des agences pour obtenir une notation de leur capacité à honorer le règlement de leurs dettes. Précieuse pour les investisseurs, cette note est générée suite à une longue et très complète analyse de ladite entreprise. Afin d'être en mesure de proposer leurs coûteux services (de l'ordre du demi-million d'euros pour noter une entreprise) au plus grand nombre et pour obtenir des informations complémentaires au travail des analystes, ces agences vont avoir également recours aux modèles de Machine Learning (tout comme pour les dettes souveraines). Combinés aux informations historiques des agences, de nombreux ratios vont servir d'éléments de référence. En voici quelques exemples :

- Dettes de long-terme/capital total investi ;
- Ratio de dettes ;
- Résultat opérationnel/capitaux reçus ;
- Marge brute. Là aussi, le modèle ACRP, basé sur le modèle du réseau de neurones artificiels est utilisé et privilégié par les agences de notation. Cela s'explique en particulier par sa redoutable efficacité. Afin d'illustrer le propos, quatre jeux d'entreprises pour lesquelles nous détenons les informations financières suffisantes ont été utilisés :
- USA-A : comprenant 265 entreprises américaines et uniquement 5 ratios analysés ;
- USA-B : comprenant 265 entreprises américaines et uniquement 16 ratios analysés ;
- TAIWAN-A : comprenant 75 entreprises taiwanaises et uniquement 5 ratios analysés ;

- TAIWAN-B : comprenant 75 entreprises taiwanaises et uniquement 16 ratios analysés. Après application du modèle ACRP, les résultats suivants sont obtenus :

Jeu de données	Réseaux neuronaux (ACRP) Classification
USA-A	78,87%
USA-B	80,00%
TAIWAN-A	79,73%
TAIWAN-B	77,03%

FIGURE 3.6 – Résultats du modèle ACRP [26]

La classification des entreprises est convaincante, néanmoins il est surprenant de voir une précision plus importante pour TAIWAN-A que pour TAIWAN-B alors que ce dernier dispose d'un plus grand nombre de données à observer. Plusieurs hypothèses peuvent expliquer cela :

- Les dépendances entre les variables ne sont pas prises en compte explicitement par le réseau et la corrélation entre variables n'est plus un indicateur satisfaisant de leur dépendance ;
- Les nouvelles données d'entrées n'ont pas véritablement de lien avec la cible, pouvant détériorer la performance de l'algorithme. De même, de nouvelles valeurs d'entrées qui diffèrent de façon significative de celles qui ont été utilisées pour l'apprentissage du réseau peuvent dégrader les résultats générés (il s'agit du phénomène d'extrapolation) ;
- Un terme a été ajouté à la fonction d'erreur de l'algorithme pour pénaliser des valeurs de poids jugées trop importantes. Une modération des poids mal calibrée peut nuire à la performance du réseau en encourageant le sous-ajustement, ceci détériorant la précision des résultats attendus ;
- Encore une fois, la notation fournie ne sera pas suffisante, cependant, elle sera un indicateur très fort pour l'analyste qui n'aura que peu d'impact dans la décision finale (dans plus de 9 cas sur 10, la note est bonne ou affiche un seul degré d'écart avec la note définitive) ; Encore une fois, la notation fournie ne sera pas suffisante,

cependant, elle sera un indicateur très fort pour l'analyste qui n'aura que peu d'impact dans la décision finale (dans plus de 9 cas sur 10, la note est bonne ou affiche un seul degré d'écart avec la note définitive) ;

- Si, dans le cas des dettes corporate, le modèle est plus pertinent, cela tient notamment du fait que la notation d'une dette souveraine prend en compte de nombreux critères difficilement quantifiables tels que la politique intérieure, extérieure, les conflits commerciaux ou autres guerres.Si, dans le cas des dettes corporate, le modèle est plus pertinent, cela tient notamment du fait que la notation d'une dette souveraine prend en compte de nombreux critères difficilement quantifiables tels que la politique intérieure, extérieure, les conflits commerciaux ou autres guerres.

3.3 Le risque de marché

Le risque de marché représente un enjeu majeur du monde financier. Nous le caractérisons souvent par le risque de pertes résultant de l'évolution des prix du marché et des valeurs des actifs observés. Ce risque est particulièrement suivi d'un point de vue règlementaire. Il s'agit de l'un des volets mis en avant par le comité Bâlois (parmi d'autres) ainsi que l'un des principaux motifs de l'existence d'EMIR - European Market Infrastructure Regulation (se concentrant sur les produits dérivés). En effet, les produits de marché représentent des montant colossaux et peuvent être à l'origine d'impacts considérables pour les gestionnaires d'actifs (les encours gérés pesaient 4000 milliards d'euros en France en 2018). L'objectif de la Direction des Risques d'un établissement financier sera alors de réduire autant que possible le risque de perte en capital sur les investissements effectués, tant pour compte propre que dans le cadre d'opérations visant à alimenter SICAV et autres OPCVM qui seront ensuite vendus à la clientèle. Focalisons-nous ici sur le risque action[49]

3.3.1 Diminution du risque avec le Machine Learning

Comme expliqué ci-avant, les algorithmes de Machine Learning ont la particularité d'être friands de grandes quantités de données, qui plus est de données de qualité. Effectivement, les jeux de données issus des marchés présentent l'avantage d'être facilement accessibles tout en étant massifs et de bonne qualité car produits par des processus industrialisés peu sensibles aux erreurs de manipulation ou aux saisies. Les algorithmes seront alors en mesure de procéder à l'étude approfondie de ces ensembles afin de distinguer les titres entre eux, créer des sous-ensembles et les classifier. Certains algorithmes, dits algorithmes de prédiction tenteront même de prédire de nouvelles données. En ressortiront les titres présentant les risques les plus importants (variation de la volatilité, à court terme comme à long terme) qui pourront ainsi être écartés des placements réalisés ou même de ceux prévus (en fonction des politiques d'investissement).

Il s'agit d'un véritable outil venant compléter les analyses réalisées par les Risk Managers qui pourront les intégrer dans leurs estimations et alors accéder à des informations jusqu'alors inaccessibles. Car là où l'humain se perd face à une quantité massive de données, l'algorithme s'épanouit, apprend et s'améliore. Il est ainsi de raison de s'interroger quant à l'efficience des modèles existants. Sont-ils tous pertinents face à cette typologie d'utilisation ? Lequel présente les meilleurs résultats ? A ce titre, prenons 3 modèles populaires :

- Le Naïve Bayes,
- Les arbres de décisions,
- Les réseaux neuronaux.

Ces trois modèles affichent des caractéristiques spécifiques dans leur fonctionnement, maiségalement dans leur simplicité d'utilisation, et donc dans leurs résultats. Pour illustrer cette comparaison, nous nous reposerons sur les résultats issus d'une étude réalisée par la Southern Illinois University présentant des éléments chiffrés venant illustrer nos propos.

Conclusion

Historiquement, le recours aux mathématiques et aux statistiques constitue un acquis de longue date au sein de l'industrie bancaire. En cela le Machine Learning constitue logiquement, au sein de la banque, une étape de plus dans le recours à la data science. Un usage d'autant plus d'actualité qu'il profite d'une forte incitation des pouvoirs publics pour accélérer la digitalisation de l'économie. En l'état de son déploiement opérationnel, il apparaît quelque peu prématuré de qualifier le Machine Learning de révolution pour le monde de la finance. En réalité, il constitue une des facettes de la transformation numérique des services financiers, qui a vu la donnée devenir l'un de ses principaux leviers. Les banques lèvent progressivement, les unes après les autres, tous les freins qui limitaient un recours élargi au Machine Learning. Ces freins touchaient à la donnée, devenue désormais très variée et disponible en grande quantité avec une qualité croissante (les autorités y veillent). Ils portaient également sur les puissances de calcul, qui bénéficient des derniers progrès technologiques en la matière. Ils concernaient enfin, l'outillage du data scientist, qui n'a jamais été aussi pléthorique, avec de nombreuses solutions logicielles,

CHAPITRE 4

LE MACHINE LEARNING APPLIQUÉ À LA DÉTECTION DES FRAUDES EN FINANCE

Introduction

Les fraudes bancaires ont aujourd'hui un impact financier important et nécessitent d'être détectées au plus vite. Ces difficultés s'accompagnent de complexités additionnelles dues au caractère temporel et très déséquilibré des données ainsi qu'à l'évolution constante de patterns ou motifs de fraudes. De plus, dans ce domaine particulier non seulement les décisions doivent pouvoir être expliquées, mais le modèle derrière la prise de décision doit être compréhensible. Les modèles d'apprentissage dits "black boxes" même expliqués à posteriori, ne peuvent donc être envisagés. C'est pourquoi, l'utilisation d'un langage symbolique, via un apprentissage supervisé de règles de décision est privilégiée.

if oldbalanceorigin - amount - newbalanceorigin = 0
and newbalanceorigin = 0
then fraud = true.

L'objectif de ce chapitre est d'induire un ensemble de règles métier dites business rules, à partir de données labellisées (exemple partiel de données inspiré de Synthetic Financial

Datasets For Fraud Detection [10] (SFD) ci-dessous) et de modèles de machine learning. Ceci permettra d'avancer dans la détection et prédiction de fraudes dans le contexte challenging de la finance et du monde bancaire où une complète transparence est nécessaire.

amount	**type**	**old balance origin**	**new balance origin**	**fraud(label)**
5324	payment	5324	0	true
695	Cashout	6870	6175	false

TABLE 4.1 – Synthetic Financial Datasets For Fraud Detection (SFD)[10]

4.1 Détection de fraudes bancaires

La détection de fraudes est le problème d'apprentissage qui vise à classifier automatiquement des événements comme frauduleux ou honnêtes. Dans le cadre d'un travail de recherche, la limitation principale réside dans le fait qu'il n'ya pas de données réelles disponibles publiquement pour étudier la détection de fraudes [47]. Dans le cadre d'une application réelle, ce sont les contraintes opérationnelles qui prennent le dessus. Comme relevé par [47], dans ce cas, le choix de la technique d'apprentissage dépend plus de contraintes techniques issues d'obligations opérationnelles et computationelles que de contraintes techniques liées aux données.

4.2 Induction de règles

Comme expliqué par J. Fürnkranz dans [39], il existe deux familles principales de méthodes pour induire des ensembles de règles : l'extraction de règles à partir d'arbre de décision (CART [23], C4.5 [53]) et le recouvrement séquentiel, c'est-à-dire l'apprentissage de règles à partir de données (CN2 [29], RIPPER [30]). Des extensions et de nouvelles approches basées sur ces modèles ont été proposés depuis. Nous pouvons mentionner FURIA, extension de RIPPER, qui construit des conditions floues ("fuzzy") apportant de la flexibilité aux règles, au détriment d'une augmentation du temps d'apprentissage [46]. Sood

et al. présente en 2020, BiLevCSS, une approche basée sur deux phases d'apprentissage rassemblant règles d'association et RIPPER qui sur- passe en terme de justesse (accuracy) les modèles existants[64]. Les algorithmes sur lesquels nous nous concentrons sont CN2 et RIPPER. CN2 [29]. Cet algorithme a pour idée générale de créer une nouvelle règle tant que les données d'entraînement ne sont pas toutes couvertes par les règles trouvées. RIPPER [30]. Cet algorithme dont la particularité est de proposer une étape de post-processing des règles, est toujours aujourd'hui l'état de l'art pour l'apprentissage de règles génériques. Il présente certaines limites en ce qui concerne la complexité du langage des règles générées. Plus le nombre de conditions possibles est élevé, plus l'apprentissage sera long. La génération de conditions comportant des agrégats ou des structures plus complexes est donc compliquée dans un temps raisonnable.

4.3 Expériences et résultats

Expériences et résultats Framework. Pour mener à bien ce projet, un environnement de travail spécifique basé sur des conteneurs est mis en place. Il comprend des modules distincts permettant :

— le tracking des expériences (MLFlow)

— l'apprentissage de modèles (Orange [37], scikit-learn [50], module personnalisés etc),

— l'utilisation d'un langage de règle commun (module R2L implémenté pour le projet)

— une utilisation interactive (Jupyter Notebook)

Données. Avant toutes expérimentations sur des données réelles, un dataset open-source et fictif de données bancaires intitulé : Synthetic Financial Datasets For Fraud Détection (SFD) [10] est choisi. Ce dataset comprend 6 millions d'observations dont 8213 transactions frauduleuses ($\approx 1\%$). Découpage training/testing choisi : 10/90%. Pour valider nos résultats sur des datasets de références, 9 datasets complémentaires disponibles dans la base de données UCI [38] sont sélectionnés (iris, adult, wine, heart disease cleveland (Robert Detrano, M.D., Ph.D.), breast cancer wisconsin, car evaluation, abalone, forest fires

[33], wine quality (red & white) [32]). Seuls les datasets adaptés pour la classification sont considérés dans la suite de ce document. Découpage training/testing choisi : 67/33%. Les premières expériences consistent à observer quelles sont les performances des algorithmes de l'état de l'art sur ces datasets.

Pre-processing. Des premiers tests sont réalisés sans et avec une étape de pré-processing manuel des données pour SFD. Celle ci se révèle être indispensable, un module automatique de feature generation et selection est créé avec un accent particulier sur la conservation de la compréhensibilité des règles. Ce module est basé sur une analyse dimensionnelle des données et sur l'information mutuelle apportée avec la classe objectif.

Premiers résultats. De manière générale RIPPER dépasse CN2, comme le montrent les métriques présentées dans la Table ci-dessous. De plus, le nombre de règles générées est comparable pour la plupart des datasets, excepté pour adult où CN2 a généré 916 règles contre 35 pour RIPPER. Cependant une analyse plus poussée sur la qualité de l'ensemble des règles générées ainsi que sur la complexité des règles est nécessaire. Moyennant un coût computationnel, la qualité des résultats pourraient être améliorée en modifiant le protocole expérimental avec de la validation croisée. Par ailleurs, travailler à partir d'un dataset synthétique généré avec des règles connues (avec et sans ajout d'erreurs) permettrait d'évaluer plus précisément la qualité de l'apprentissage.

dataset	model	acc	*bal_acc*	f1	precision	recall
raw SFD	cn2	0.999	0.798	0.691	0.822	0.596
raw SFD	ripper	1	0.873	0.839	0.956	0.748
processed SFD	cn2	1	0.994	0.993	0.997	0.988
processed SFD	ripper	1	0.998	0.996	0.997	0.996
car evalution	cn2	0.816	0.52	0.828	0.884	0.816
car evalution	ripper	0.837	0.699	0.855	0.874	0.837
breast cancer wisconsion	cn2	0.65	0.65	0.788	1	0.65
breast cancer wisconsion	ripper	0.965	0.959	0.965	0.965	0.965
heart disease cleveland	cn2	0.515	0.318	0.525	0.567	0.515
heart disease cleveland	ripper	0.556	0.478	0.574	0.619	0.556
wine	cn2	0.889	0.914	0.889	0.902	0.889
wine	ripper	0.933	0.929	0.933	0.934	0.933
adult	cn2	0.782	0.747	0.839	0.934	0.782
adult	ripper	0.843	0.815	0.857	0.884	0.843
iris	cn2	O.98	0.979	0.98	0.981	0.98
iris	ripper	0.98	0.979	0.98	0.981	0.98

TABLE 4.2 – Synthetic Financial Datasets For Fraud Détection (SFD)[33]

Conclusion

Dans le domaine bancaire, la détection de fraudes doit être explicable et le modèle de décision compréhensible. Cela justifie uan prentissage direct de règles. CN2 et RIPPER sont comparés sur différents datasets.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons présenté, dans un premier temps, les éléments du machine learning, les contenus de machine learning où nous avons expliqué les méthodes d'apprentissages Machine Learning.

Nous avons par la suite décrit les méthodes de machine learning avec les algorithmes adequats, décrit quelques applications de Machine Learning en finances (la gestion d'actifs, la gestion de portefeuille).

Nous avons à la fin présenté une exemple d'application du machine Learning appliqué à la détection des fraudes en finance avec apprentissage de règles.

Le but de ce mémoire est de présenter des bases solides sur les concepts et les algorithmes de ce domaine en plein essor. Il aidera à identifier les problèmes qui peuvent être résolus par une approche machine learning, à les formaliser, à identifier les algorithmes les mieux adaptés à chaque cas, à les mettre en oeuvre, et enfin à savoir évaluer les résultats obtenus, découvrir comment le machine learning ou apprentissage automatique est utilisé dans différents secteurs notamment la finance, la bourse, ...

On peut envisager certaines perspectives pour continuer la thématique abordée ; qui peuvent s'inscrire dans le cas d'études d'exemples propres à l'Algérie (des entreprises algériennes, avec des données réelles).

BIBLIOGRAPHIE

[1] http://ml_openclassrons.com/.

[2] https://www.coe.int/fr/web/artificial-intelligence/glossary.

[3] https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-non-/lineaires/4730716-entrainez-un-reseau-de-neurones-simple.

[4] https://fr.wikipedia.org/w/index.php?title=Histoire_de_l%/27intelligence_artificielle&oldid=132868931.

[5] https://en.wikipedia.org/w/index.php?title=De_Arte_Combinatoria&oldid=782514244.

[6] https://mrmint.fr/introduction-k-nearest-neighbors.

[7] file:///C:/Users/pc/Downloads/KNN.html.

[8] https://www.math.univ-toulouse.fr/~gadat/Ens/M2SID/12-m2-SVM.

[9] https://www.esilv.fr/quels-outils-informatiques-pour-le-machine-learning/#:~:text=Parmi%/20/0frameworks%/2C%/20on%/20compte,learning%/20et%/20de%/20deep%/20learning.

[10] Synthetic financial datasets for fraud detection. Library Catalog : www.kaggle.com, 2016.

[11] A. B. A. B. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622. Polytechnic Institute of Brooklyn, 1962.

[12] J. S. Abarbanell and B. J. Bushee. Fundamental analysis, future earnings, and stock prices. *Accounting Research*, 35(1) :1–24, 1997.

[13] D. T. Larose (adaptation française T. Vallaud). *Des données à la connaissance : Une introduction au data-mining (1 Cédérom), Vuibert*. 2005.

[14] C. C. Aggarwal. Recommender systems. 2016.

[15] D. W. Aha. Special issue on lazy learning.artificial intelligence review. 11((1–5)) :7–423.

[16] M. R. Amini. *Apprentissage machine de la théorie à la pratique*. Eyrolles, 2015.

[17] Aronszajn. Transactions of the american mathematical society. *Theory of reproducing kernels*, 68(3) :337–404, 1950.

[18] S. Balech and C. Benavent. Les techniques du NLP pour la recherche en sciences de gestion. Technical report, CRIISEA - Centre de Recherche sur les Institutions, l'Industrie et les Systèmes Économiques d'Amiens, 2019.

[19] R. S. Barto and A. G. Sutton. *Reinforcement Learning : An introduction. MIT Press, Cambridge, MA*. 1998.

[20] J. L. Bentley. Multidimensional binary search trees used for associative searching. communications of the acm. 18 :509–517, 1957.

[21] F. Benureau. *Self-Exploration of Sensorimotor Spaces in Robots.* Thèse de doctorat, Université de Bordeaux, 2015.

[22] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. in proceedings of the fifth annual workshop on computational learning theory. *Pittsburgh, Pennsylvania, United States. ACM*, page 144–152, 992.

[23] L. Breiman. *Classification and Regression Trees*. Taylor and Francis, 1984.

[24] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *CART : Classification and Regression Trees, Wadsworth International*. 1984.

[25] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2 :121–167, 1998.

[26] A. Caiazzo, C. N. Champassak, G. Floch, and H. Salem. *Le machine Learning En Finance*. Square, JUILLET 2020.

[27] C.-C. Chang and C.-J. Lin. Libsvm : A library for support vector machines. 2008.

[28] V. Cherkassky and F. Mulier. Learning from data : Concepts, theory, and methods. wiley, new york. 1998.

[29] P. Clark and T. Niblett. *The CN2 Induction Algorithm*, volume 3. 1989.

[30] W. W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, page pages 115–123. Morgan Kaufmann, 1995.

[31] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20 :273–297, 1995.

[32] P. Cortez, A. Cerdeira, F. Almeida, T. MATOS, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47 :547–553, 2009.

[33] P. Cortez and A. J. R. Morais. A data mining approach to predict forest fires using meteorological data. In *New trends in artificial intelligence : proceedings of the 13th Portuguese Conference on Artificial Intelligence*, page 12, 2007.

[34] M. R. Coulom. *Apprentissage par renforcement utilisant des réseaux de neurones, avec des application au contôle moteur*. PhD thesis, l'institut national polytechnique de GRENOBLE, 2002.

[35] G. Cybenko. Approximation by superpositions of a sigmoidal function. mathe- matics of control, signals and systems. 2(4) :303–314, 1989.

[36] C. Della-Vedova. Introduction à la régression logistique. https://delladata.fr/regression-logistique/, 2020.

[37] J. Demšar. *Orange : Data mining toolbox in python*, volume 14. 2011.

[38] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[39] J. Fürnkranz. *Foundations of Rule Learning*. Springer Science and Business Media, 2012.

[40] B. Gaüzère. *Application des méthodes à noyaux sur graphes pour la prédiction des propriétés des molécules.thèse de doctorat*. PhD thesis, Université de Caen, 2013.

[41] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[42] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, page 857–871, 1971.

[43] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction, Second Edition*. Springer-Verlag, New York, 2nd edition., 2009.

[44] K. Hechenbichler and K. Schliep. Weighted k-nearest-neighbor techniques and ordinal classification. SFB 386. Discussion paper 399, 2006.

[45] K. Hornik. Approximation capabilities of multilayer feeedforward networks.neural netwarks. 4(2) :251–257, 1991.

[46] J. Hühn. *FURIA : an algorithm for unordered fuzzy rule induction*.

[47] C. Phua l. *A Comprehensive Survey of Data Mining-based Fraud Detection Research*, volume 28(3). May 2012, arXiv : 1009.6119.

[48] K. Minsky and S. Papert. perceptrons : an introduction to computational geometry. MIT press, cambridge, MA. 1972.

[49] J. P. Mueller and L. Massaron. *Machine Learning For Dummies*. Wiley, 2016.

[50] F. Pedregosa. *Scikit-learn : Machine learning in Python*, volume 12. 2011.

[51] S. Madeh Piryonesi and Tamer E. El-Diraby. « data analytics in asset management : Cost-Effective prediction of the pavement condition index ». *Infrastructure Systems*, vol.26 :p.04019036, mars 2020.

[52] J. C. Platt. Sequential minimal optimization : a fast algorithm for training support vector machines. technical report msr-tr-98-14, microsoft research. 1998.

[53] J. R. Quinlan. C4.5 : Programs for machine learning. elsevier. June 1993.

[54] R. J. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[55] Jean-Marc Quéré. *WinDev 9 : Implémentation de méthodes décisionnelles*. 2005.

[56] F. Ricci, L. Rokach, B. shapira, and B. Kantor. *Recommender systems handbook*. Springer, 2 edition, 2016.

[57] F. Rosenblatt. The perceptron-a perceiving and recognizing automaton. technical report. *Cornell Aeronautical Laboratory*, pages 65–460–1, 1957.

[58] S. Russell and P. Norvig. *Apprentissage machine de la théorie à la pratique*. Eyrolles, 2010.

[59] G. Saint-Cirgue. Apprendre le machine learning en une semaine. 2019.

[60] A. L. Samuel. Some studies in machine learning using the game of chechers. *IBM-journal of research and development*, 44(1.2) :206–226, 1959.

[61] C. Saunders and A. Demco. Kernels for strings and graphs. in perspectives of neural-symbolic integration, studies in computational intelligence. *Springer, Berlin, Heidelberg*, page 7–22, 2007.

[62] D. W. Scott. Multivariate density estimation. *Wiley(New York)*, 1992.

[63] D. W. Scott. *Multivariate density estimation*. Wiley, New York, 1992.

[64] N. Sood. Bi-level associative classifier using automatic learning on rulesin. database and expert systems applications. *Springer International Publishing*, page pages 201–216, 2020.

[65] D.M. Tax and R.P. Duin. *Support vector data description. Machine learning*, volume 54(1). 2004.

[66] C. Touze. Les réseaux de neurones artificiels, introduction au connexionnisme, 1992.

[67] V. Vapnik and A. Lerner. *Pattern recognition using generalized portrait method. Automation and Remote Control*, volume 24. 1963.

[68] J. A. Vayssade. Classification par k plus proches voisins. http://sleek-think.ovh/index.php/10-cours/l3-maths-pour-l-info/27-classification-par-k-plus-proches-voisins.

[69] A. R. Webb. *Statistical Pattern Recognition*. Wiley, 1999.

[70] D. Zighed and R. Rakotomalala. *Graphes d'Induction-Apprentissage et Data Mining*. 2000.

Résumé

Dans ce travail, nous avons abordé le machine learning et son application dans le domaine de la finance. Plus précisément, nous avons présenté les méthodes utilisées dans l'apprentissage, à savoir la méthode des plus proches voisins, méthode de réseaux de neurones artificiels, arbres de décision et méthode de Machines à vecteurs de support (SVM) pour développer afin de tester et d'appliquer des algorithmes d'analyse prédictive sur différents types de données afin de prédire le futur. Par la suite, nous avons présenté quelques applications de Machine Learning en finances comme la gestion d'actifs, la gestion de portefeuille et le risque de marché pour optimiser leurs coûts, d'améliorer l'expérience de leurs clients et de développer leurs services. Nous avous à la fin présenté une exemple d'application du machine Learning appliqué à la détection des fraudes en finance avec apprentissage de règles.

Mots-clés : Machine Learning, k-plus proches voisins, Réseaux de neurones, Régresssion, SVM, detection de fraudes.

Abstract

In this work, we have discussed machine learning and its application in the field of finance. Specifically, we presented the methods used in machine learning, namely Nearest Neighbors method, Artificial Neural Networks method, Decision Trees and Support Vector Machines (SVM) method to develop in order to test and apply predictive analysis algorithms on different types of data in order to predict the future. Subsequently, we presented some Machine Learning applications in finance such as asset management, portfolio management and market risk to optimize their costs, improve the experience of their customers and develop their services. At the end, we presented an example of the application of machine learning applied to fraud detection in finance with rule learning.

Key-words : Machine Learning, k-nearest neighbours, neural networks, regression, SVM, fraud detection.