République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département Informatique

Mémoire en vue de l'obtention du diplôme de Master

en Génie des Systèmes Informatiques

Thème

Apprentissage profond pour la santé intelligente : Application à la santé cardiovasculaire

---

Membres du jury :

- AID Aicha (Président)
- ALIOUAT Wahiba (Examinateur)
- HAMID Rabah (Examinateur)
- CHOUIREF Zahira (Encadreur)

Réalisé par :

- MANAA Abderzak
- KESSOURI Mohamed

2021/2022

Remerciements

Nous remercions, en premier lieu, Allah, le tout puissant, de nous avoir permis et accorder la volonté, la patience et le courage pour réaliser ce travail. Nos parents respectifs, qui sont pour nous une source de vie car sans leurs sacrifices et leur précieux conseils ne nous pourrons pas arriver jusqu'au bout. Que Dieu les garde afin que leur regard puisse suivre nos destinées.

Nous présentons notre immense gratitude et nos remerciements, les plus sincères, à notre encadreur Dr. CHOUIREF Zahira, enseignante à la faculté des sciences et des sciences appliquées de Bouira, pour son encadrement, sa disponibilité, ses orientations pertinentes et avisées, sa patience, et surtout ses qualités humaines ont constitué un apport considérable, sans lequel, ce travail n'aurait pas vu le jour.

Nous tenons également à remercier et exprimer notre profond respect aux membres de jury d'avoir accepté d'évaluer ce travail. Leurs pertinentes remarques et suggestions permettrons surement d'enrichir et d'améliorer notre travail.

Nous tenons à exprimer nos sincères remerciements à tous les enseignants qui nous ont enseigné et qui par leurs compétences et sérieux nous ont permis de poursuivre nos études.

Merci à tous...

MANAA Abderzak

KESSOURI Mohamed

Résumé

Les maladies cardiovasculaires regroupent un certain nombre de troubles affectant le coeur et les vaisseaux sanguins, elles sont considérées comme l'une des principales causes de décès au niveau mondial ces dernières années. Les patients atteint de maladie cardiaque ne se sentent malades qu'au tout dernier stade de la maladie et les dommages deviennent irrémédiables et la plupart des patients cardiaques meurent avant de recevoir un traitement.

Les techniques de machine Learning (ML) ont une supériorité écrasante pour résoudre le problème des soins aux patients cardiaques car elles peuvent prédire la maladie à un stade précoce en se basant sur les signaux physiologiques recueillis par les objets connectés afin d'éviter des complications antérieures comme l'infection de l'artère coronaire et la diminution de la fonction des vaisseaux sanguins.

Le principal objectif de ce projet est de développer une approche basée sur l'apprentissage profond pour aider à diagnostiquer et prédire les maladies cardiovasculaires à partir d'un ensemble de données relatives à des patients réels.

L'approche proposée permet de prédire le risque de maladie, car elle peut nettoyer les données, les normaliser et extraire des caractéristiques pertinentes à partir de données structurées, et représente ces caractéristiques extraites efficacement avec un faible poids dimensionnel et spécifique à l'aide de la sélection de caractéristiques pour produire des résultats optimaux avec deux méthodes de validation différentes afin d'améliorer les performances et la complexité pratique du modèle proposé et ainsi réduire l'erreur de prédiction des maladies cardiovasculaires. Le modèle prédictif proposé est testé sur une application réalisée pour les besoins de ce test.

Les résultats montrent que les performances du modèle prédictif proposé (Dense-DNN) sont meilleures que les performances des modèles ML (la SVM, la régression logistique, la forêt aléatoire, l'arbre de décision, Bayes naïf gaussien, le k-plus proche voisin et le XGBoost) avec un accuracy de 91,7% et 95% avec la méthode de validation « split-validation » en utilisant respectivement l'ensemble de données traitées et l'ensemble de données traitées et réduites et un accuracy de 84,8% et 85,4% avec la méthode de validation « cross-validation » en utilisant respectivement l'ensemble de données traitées et l'ensemble de données traitées et réduites.

Mots-clés : Techniques de Deep Learning, Filtrage des données, Sélection de caractéristiques, Algorithme génétique, Algorithmes de classification, Métriques d'évaluation, Métriques d'erreur, Complexité pratique, Prédiction des maladies cardiovasculaires.

Abstract

Cardiovascular diseases include a number of disorders affecting the heart and blood vessels, they are considered one of the main causes of death worldwide in recent years. Patients with heart disease do not feel sick until the very last stage of the disease and the damage becomes irreparable and most heart patients die before receiving treatment.

Machine Learning (ML) techniques have overwhelming superiority in solving the problem of cardiac patient care as they can predict disease at an early stage based on physiological signals collected by IoT devices to avoid earlier complications such as coronary artery infection and decreased blood vessel function.

The main objective of this project is to develop an approach based on deep learning to help diagnose and predict cardiovascular diseases from a set of data relating to real patients.

The proposed approach helps predict disease risk, as it can clean data, normalize and extract relevant features from structured data, and represents these extracted features efficiently with low dimensional and specific weight using the selection of features to produce optimal results with two different validation methods in order to improve the performance and practical complexity of the proposed model and thus reduce the prediction error of cardiovascular diseases. The proposed predictive model is tested on an application created for the purposes of this test.

The results show that the performances of the proposed predictive model (Dense-DNN) are better than the performances of the ML models (the SVM, the Logistic Regression, the Random Forest, the Decision Tree, the Gaussian Naive Bayes, the K-Nearest Neighbors and the XGBoost) with an accuracy of 91.7% and 95% with the split-validation method using respectively the processed dataset and the processed and reduced dataset and a accuracy of 84.8% and 85.4% with the cross-validation method using the processed dataset and the processed and reduced dataset, respectively.

Keywords : Deep learning techniques, Data filtering, Feature selection, Genetic algorithm, Classification algorithms, Evaluation metrics, Error metrics, Practical complexity, Cardiovascular disease prediction.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

OMS	Organisation Mondiale de la Santé
AVC	Accidents Vasculaires Cérébraux
IoT	Internet of Things (Internet des objets)
DNN	Deep Neural Network (Réseau neuronal profond)
MCV	Maladie CardioVasculaire
ML	Machine Learning (Apprentissage automatique)
KDD	Knowledge Discovery in Databases (Extraction d'information d'une base de données)
UCI	University of California, Irvine (Université de Californie à Irvine)
PSO	Particle Swarm Optimization (Optimisation par essaims particulaires)
ADN	Acide DésoxyriboNucléique
MLP	MultiLayer Perceptron (Perceptron multicouche)
SVM	Support Vector Machine (Machine à vecteurs de support)
LR	Logistic Regression (Régression logistique)
RF	Random Forest (Forêt aléatoire)
DT	Decision Tree (Arbre de décision)
NB	Naïve Bayes (Naïve bayésienne)
KNN	K-Nearest Neighbors (K plus proches voisins)
XGBoost	eXtreme Gradient Boosting
TP	True Positive (Vrai positif)
FN	False Negative (Faux négatif)
FP	False Positive (Faux positif)
TN	True Negative (Vrai négatif)
MAE	Mean Absolute Error (Erreur absolue moyenne)
RMSE	Root Mean Squared Error (Racine de l'erreur moyenne quadratique)

ROC	Receiver Operating Characteristic (Caractéristique de fonctionnement du récepteur)
AUC	Area Under the Curve (Aire sous la courbe)
NN	Neural Network (Réseau de neurones)
CNN	Convolutional Neural Network (Réseau neuronal convolutif)
DL	Deep Learning (Apprentissage profond)
GNB	Gaussian Naïve Bayes (Naïve Bayes gaussien)
BDD	Base De Données
AVG	AVeraGe (Moyenne)
AdaBelief	Adapting stepsizes by the Belief in observed gradients
SGD	Stochastic Gradient Descent (Descente de gradient stochastique)
Adam, ADAM	ADAptive Moment estimation (Estimation adaptative du moment)
EWA, EWMA	Exponentially Weighted Moving Average (Moyenne mobile pondérée exponentiellement)
ReLU	Rectified Linear Unit (Unité linéaire rectifiée)
RAM	Random Access Memory (Mémoire à accès aléatoire)

Introduction générale

Selon l'OMS (chiffres 2015) [10], on estime à 17,7 millions le nombre de décès imputables aux maladies cardio-vasculaires, soit 31% de la mortalité mondiale totale. Parmi ces décès, on estime que 7,4 millions sont dus à une cardiopathie coronarienne et 6,7 millions à un AVC. Plus des trois quarts des décès liés aux maladies cardiovasculaires interviennent dans des pays à revenu faible ou intermédiaire. Sur les 17 millions de décès survenant avant l'âge de 70 ans et liés à des maladies non transmissibles, 82% se produisent dans des pays à revenu faible ou intermédiaire et 37% sont imputables aux maladies cardiovasculaires. Ils ne peuvent souvent pas bénéficier des programmes intégrés de soins de santé primaires pour la détection précoce et le traitement des personnes à risque par rapport aux habitants des pays à revenu élevé.

Avec la migration de la population rurale vers les villes urbaines, l'utilisation optimisée de l'infrastructure existante a ouvert la voie à la conception de diverses applications dans les villes intelligentes. Une de ces applications est la santé intelligente, qui est essentielle pour la survie de l'humanité.

Le principe de base de la santé intelligente est l'utilisation d'appareils intelligents, de capteurs et L'IoT (Internet of Things) pour fournir des services de santé améliorés à faible coût aux résidents, facilement, à n'importe quel endroit et à n'importe quel moment. Puisque le ratio médecins-patients dans les pays en développement est très faible, les soins de santé intelligente peuvent jouer un rôle vital pour le suivi des patients à distance grâce à des capteurs corporels, qui peuvent transmettre les données au cloud, auquel les médecins peuvent accéder et fournir des ordonnances. Ces capteurs sont facilement disponibles sur le marché, qui sont implantables ou portables et peuvent être utilisés pour la surveillance de leur santé à distance.

Cependant, les défis majeurs de la santé intelligente restent les mêmes, c'est-à-dire, l'analyse et la gestion des données. Les données dans les services de santé sont recueillies à un rythme rapide qui nécessite un processus spécialisé. Ainsi, une technique de réduction des données doit être intégrée dans les système existant qui ne prend en compte que les données importantes.

Cette question a suscité beaucoup d'intérêt de la part des chercheurs ; l'une des tâches critiques dans ce domaine est de prédire la maladie présente dans le corps humain. Même les médecins ne sont pas efficaces pour prédire la maladie [11]. Cependant, ils ont besoin d'un système de soutien pour prédire la maladie. Certains algorithmes sont pris en charge mais il faut améliorer les performances du système au-delà du système existant. Par conséquent, pour aider les médecins, il y a un énorme potentiel de recherche dans la prédiction des maladies cardiovasculaires chez les humains.

Les algorithmes d'apprentissage automatique sont le type de système de soutien proposé dans ce travail avec la technique d'apprentissage profond pour aider le personnel soignant au diagnostique et à la prédiction des maladies cardiovasculaires. L'un des algorithmes d'apprentissage profond, à savoir le réseau profond (DNN), diagnostique mieux les maladies que les méthodes existantes. Ce modèle basé sur DNN traite un grand volume de données.

L'approche proposée permet de prédire le risque de maladie, car elle peut extraire des caractéristiques pertinentes à partir de données structurées, et représente ces caractéristiques extraites efficacement avec un faible poids dimensionnel et spécifique à l'aide de la sélection de caractéristiques pour produire des résultats optimaux avec deux méthodes de validation différentes afin d'améliorer les performances et la complexité pratique du modèle proposé et ainsi réduire l'erreur de prédiction des maladies cardiovasculaires. Le modèle prédictif proposé est testé sur une application réalisée pour les besoins de ce test.

Le reste du manuscrit est structuré comme suit : Le premier chapitre donne des généralités sur les maladies cardiovasculaires. Le deuxième chapitre décrit les travaux de la littérature existante. Le troisième chapitre donne un aperçu sur la méthodologie proposée. Le quatrième chapitre décrit, évalue et discute des résultats expérimentaux et teste le modèle prédictif proposé sur une application de bureau réalisée pour les besoins de ce test.

Chapitre 1

Généralités sur les maladies cardiovasculaires

1.1 Introduction

Les Maladies CardioVasculaires (MCV) sont responsables de la majorité des décès dans le monde. Leur incidence augmente dans tous les pays, bien que leur prise en charge s'améliore constamment.

Les habitants les plus pauvres des pays à revenu faible ou intermédiaire sont les plus touchés. Il est amplement démontré que les maladies cardiovasculaires et d'autres maladies non transmissibles contribuent à la pauvreté des ménages du fait des dépenses de santé catastrophiques et du niveau élevé des paiements directs auxquels ceux-ci doivent faire face.

Au niveau macroéconomique, les maladies cardiovasculaires prélèvent un lourd tribut sur les économies des pays à revenu faible ou intermédiaire.

Les personnes souffrant de maladies cardiovasculaires ou exposées à un risque élevé de maladies cardiovasculaires (du fait de la présence d'un ou plusieurs facteurs de risque comme l'hypertension, le diabète, l'hyperlipidémie ou une maladie déjà installée) nécessitent une détection précoce et une prise en charge comprenant soutien psychologique et médicaments, selon les besoins [10].

1.2 Structure du coeur

Le coeur est un organe essentiellement musculaire tapissé en dedans par l'endocarde qui se continue par l'endothélium vasculaire. Il est recouvert à sa surface par le péricarde viscéral ou épicarde. Cette masse musculaire ou myocarde est constituée de fibres auriculaires et ventriculaires qui s'insèrent sur une solide charpente de tissu fibreux.

Le squelette fibreux du coeur est formé par les quatre anneaux fibreux valvulaires : l'atrioventriculaire gauche situé à gauche et légèrement en arrière de l'atrioventriculaire droit, piriforme ; l'aortique en avant des deux orifices atrioventriculaires et le pulmonaire en avant et à gauche de l'orifice aortique ; ces deux derniers anneaux sont festonnés. Ces quatre anneaux sont réunis par trois amas fibreux, plus épais, ou trigones.

Le trigone antérieur droit : il est situé au-dessous de la commissure entre la valvule semi-lunaire droite et la valvule semi-lunaire postérieure ; il se poursuit en bas par le septum membranacé interventriculaire. Sa face externe répond à l'oreillette droite de laquelle il peut être séparé. Plus bas et plus en avant, sur sa face externe, s'insère la cuspide septale de la valve atrioventriculaire droite. Ainsi, les deux anneaux atrioventriculaires et l'anneau aortique sont-ils unis entre eux par un bloc fibreux dense.

Le trigone antérieur gauche : il est situé au dessous de la commissure entre les valvules semilunaires coronaires droite et gauche ; il est peu étendu et se poursuit par le segment le plus antérieur du septum interventriculaire ; c'est au dessous de lui, dans le muscle, qu'est effectuée la myotomie de Bigelow dans le traitement de la cardiomyopathie obstructive.

À partir des trigones gauche et droit, des prolongements de tissu fibroélastique encerclent les anneaux atrioventriculaires gauche et droit et s'étendent en avant en formant une couronne festonnée à trois pointes sur laquelle se fixent la racine et les valvules aortiques ; une formation identique à droite de l'aorte constitue la racine et les valvules pulmonaires.

Sur cette charpente fibreuse s'insèrent les fibres myocardiques : en avant les fibres ventriculaires, en arrière les fibres atriales. Parmi ces fibres, certaines sont spécialisées et constituent le système de commande ou système cardionecteur [1].

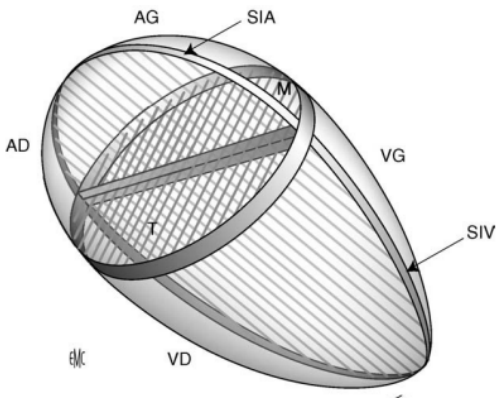


FIGURE 1.1 – Structure du coeur [1]

AD, AG : Atrium Droit, Atrium Gauche

VD, VG : Ventricule Droit, Ventricule Gauche

SIA, SIV : Septum InterAtrial, Septum InterVentriculaire

M, T : valve Mitrale, valve Tricuspide

1.3 Morphologie du coeur

Le coeur est classiquement décrit comme ayant une forme de pyramide triangulaire chez le cadavre et d'oeuf chez le sujet vivant. Il présente un grand axe presque horizontal dirigé en avant, à gauche et un peu en bas. Son axe peut varier avec la morphologie du thorax : il se verticalise lorsque le thorax est étroit ou au contraire s'horizontalise lorsque le thorax est large. Ainsi dans la description modale, l'apex du coeur est en avant et à gauche et sa base regarde en arrière et à droite. Les deux tiers du coeur sont situés à gauche de la ligne médiane.

Le coeur est composé de quatre cavités associées deux à deux permettant ainsi de distinguer un « coeur droit » et un « coeur gauche », qui normalement ne communiquent pas entre eux. En rapport avec leur rôle physiologique, le coeur droit possède une structure adaptée au régime veineux à basse pression, alors que le coeur gauche présente une structure adaptée au régime artériel à haute pression.

À la surface du coeur, les limites des oreillettes et des ventricules sont marquées par des sillons, d'une part les sillons interatriaux et interventriculaires qui passent par le grand axe de la pyramide et d'autre part les sillons auriculoventriculaires qui sont perpendiculaires au grand axe du coeur. Les troncs principaux des artères coronaires et de leurs principales collatérales cheminent dans ces sillons. C'est à ce niveau qu'elles sont abordées lors de la réalisation des pontages coronaires. Ces sillons sont comblés par de la graisse qui déborde sur les parois des cavités, donnant au coeur un aspect plus ou moins graisseux entremêlé de zone de myocarde rougeâtre. Au fond des sillons cheminent les paquets vasculonerveux. Les oreillettes ne sont jamais recouvertes de graisse et ont une couleur allant du mauve au rouge.

On décrit au coeur trois faces (antérodroite, inférieure et latérale gauche), un sommet et une base. Chacune des faces est divisée par le sillon atrioventriculaire en un segment antérieur ou ventriculaire et un segment postérieur ou atrial [1].

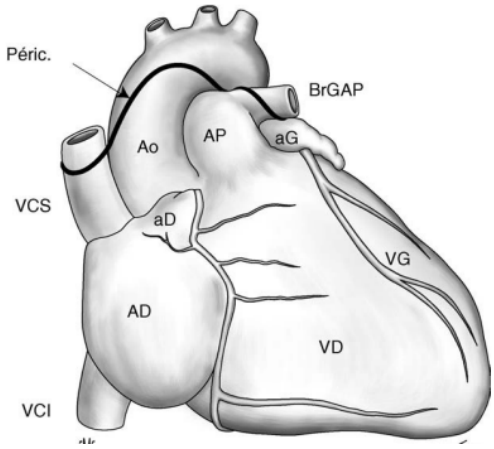


FIGURE 1.2 – Morphologie du coeur [1]

Ao, AP : Aorte ascendante, Artère Pulmonaire

BrGAP : Branche Gauche Artère Pulmonaire

VCS, VCI : Veine Cave Supérieure, Veine Cave Inférieure

AD, aD, aG : Atrium Droit, auricule Droit, auricule Gauche

Péric. : ligne de réflexion du Péricarde

VD, VG : Ventricule Droit, Ventricule Gauche

1.4 Fonctionnement du coeur

Le « service rendu » par le cœur à l'ensemble des organes et tissus est un débit sanguin. Chaque jour, le coeur propulse environ 8000 litres de sang, apportant l'oxygène et les nutriments, et éliminant les déchets du métabolisme. Ce débit doit être fourni sous une certaine pression, permettant le réglage de la distribution sanguine dans chaque organe en fonction de ses besoins propres sans compromettre l'équilibre général.

Quatre valves cardiaques, situées entre les oreillettes et les ventricules d'une part, et à la sortie des ventricules d'autre part, empêchent, lorsqu'elles sont fermées, le reflux du sang dans le mauvais sens. La fermeture des valves produit le son familier du battement du coeur [2].

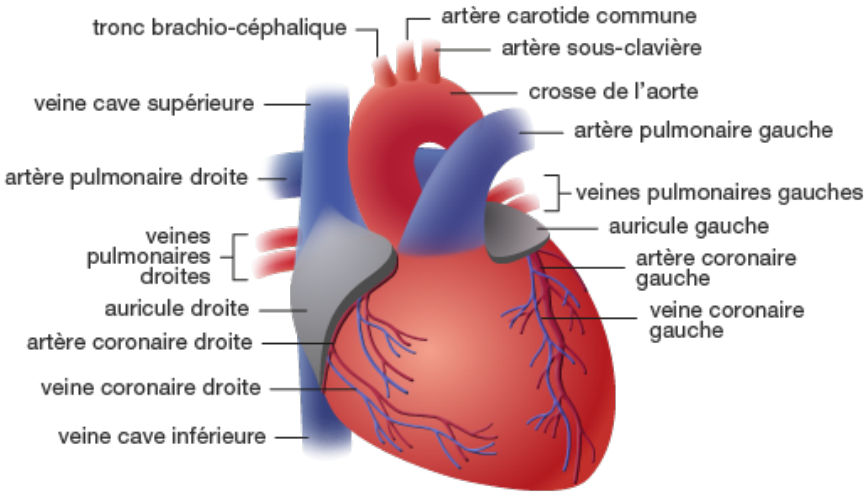


FIGURE 1.3 – Vue générale du coeur [2]

1.4.1 Circulation du sang

On peut résumer la circulation sanguine comme suit [2] :

1. Le sang désoxygéné arrivant de toutes les parties du corps,
2. pénètre dans l'oreillette droite,
3. qui se contracte et éjecte le sang dans le ventricule droit.
4. La valve située entre ces deux compartiments se ferme. Le ventricule droit se contracte et propulse le sang dans le tronc pulmonaire.
5. La valve située à la base du tronc pulmonaire se ferme. Le sang est envoyé vers les poumons où il s'enrichit en oxygène.
6. Le sang oxygéné arrivant des poumons
7. est recueilli par l'oreillette gauche
8. qui se contracte et expulse le sang dans le ventricule gauche.
9. La valve située entre ces deux compartiments se ferme. Le ventricule gauche se contracte et propulse le sang dans l'aorte.
10. La valve située au départ de l'aorte se ferme. Le sang est distribué dans tout l'organisme.

Les artères coronaires, alimentées par l'aorte, assurent l'approvisionnement en sang oxygéné du cœur lui-même.

Les contractions cardiaques se déroulent en parallèle dans les parties droite et gauche du coeur :

1. contraction des oreillettes droite et gauche
2. contraction des ventricules droite et gauche

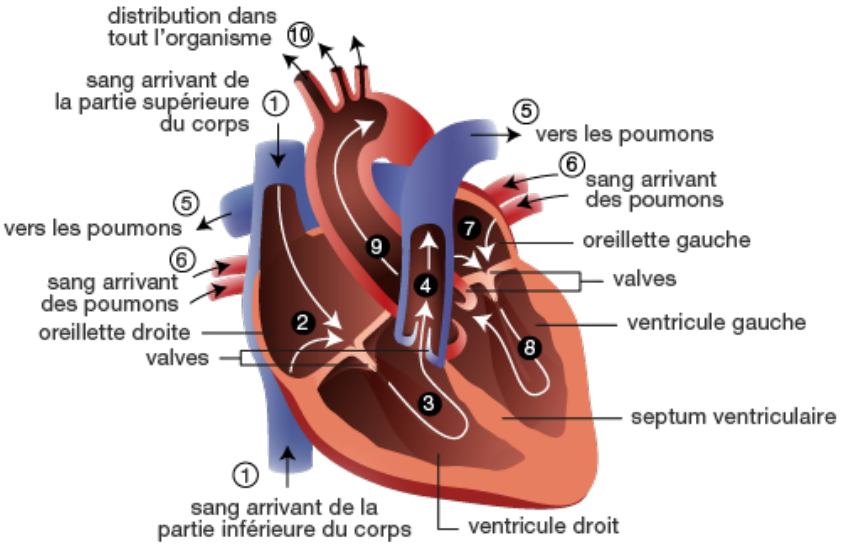


FIGURE 1.4 – Détails du coeur [2]

1.5 Les maladies cardiovasculaires

Les maladies cardiovasculaires constituent un ensemble de troubles affectant le cœur et les vaisseaux sanguins, qui comprend :

— les cardiopathies coronariennes (touchant les vaisseaux sanguins qui alimentent le muscle cardiaque)

— les maladies cérébro-vasculaires (touchant les vaisseaux sanguins qui alimentent le cerveau)

— les artériopathies périphériques (touchant les vaisseaux sanguins qui alimentent les bras et les jambes)

— les cardiopathies rhumatismales, affectant le muscle et les valves cardiaques et résultant d'un rhumatisme articulaire aigu, causé par une bactérie streptocoque

— les malformations cardiaques congénitales (malformations de la structure du cœur déjà présentes à la naissance)

— les thromboses veineuses profondes et les embolies pulmonaires (obstruction des veines des jambes par un caillot sanguin, susceptible de se libérer et de migrer vers le coeur ou les poumons).

Les infarctus et les accidents vasculaires cérébraux sont généralement des événements aigus et sont principalement dus au blocage d'une artère empêchant le sang de parvenir au cœur ou au cerveau. Leur cause la plus courante est la constitution d'un dépôt gras sur les parois internes des vaisseaux sanguins alimentant ces organes. Les accidents vasculaires cérébraux peuvent aussi résulter du saignement d'un vaisseau sanguin cérébral ou de caillots.

Les infarctus et les AVC sont généralement dus à la présence de plusieurs facteurs de risque associés comme le tabagisme, une mauvaise alimentation et l'obésité, la sédentarité et l'utilisation nocive de l'alcool, l'hypertension, le diabète et l'hyperlipidémie [10].

1.6 Les facteurs de risque

Les principaux facteurs de risques des cardiopathies et des AVC, sont une mauvaise alimentation, un manque d'activité physique, le tabagisme et l'usage nocif de l'alcool.

Les effets des facteurs de risque comportementaux peuvent se traduire chez les personnes par une hypertension, une hyperglycémie, une hyperlipidémie, le surpoids et l'obésité. Ces « facteurs de risque intermédiaires » peuvent être évalués dans les établissements de soins de santé primaires et ils sont le signe d'un risque accru d'infarctus, d'accident vasculaire cérébral, de défaillance cardiaque et d'autres complications.

On a constaté que cesser de fumer, réduire l'apport en sel dans son alimentation, consommer des fruits et des légumes, pratiquer une activité physique régulière et éviter l'usage nocif de l'alcool permettaient de réduire le risque de maladie cardiovasculaire. En outre, le traitement médicamenteux du diabète, de l'hypertension et de l'hyperlipidémie peut s'avérer nécessaire pour diminuer le risque cardiovasculaire et prévenir les infarctus et les AVC. Les politiques de santé, qui créent des conditions propices pour qu'il soit à la fois abordable et possible de faire les bons choix en matière de santé, sont essentielles pour inciter les populations à adopter un comportement sain et à s'y tenir.

Il existe aussi un certain nombre de déterminants sous-jacents des maladies cardiovasculaires. Ils proviennent des principales évolutions sociales, économiques et culturelles - la mondialisation, l'urbanisation et le vieillissement de la population. D'autres déterminants des maladies cardiovasculaires sont la pauvreté, le stress et les facteurs héréditaires [10].

1.7 La prévention des maladies cardiovasculaires

Les actions de prévention possibles sont de plusieurs types, elles concernent le diagnostic et la mise en œuvre de ces actions tant au niveau individuel que pour l'ensemble de la population

et il est recommandé de les combiner afin de réduire la très forte charge que représentent les maladies cardiovasculaires. Selon les études coût-efficacité réalisées par l'OMS, les mesures les plus efficientes sont celles qui réduisent, au moyen de l'éducation de la population et de mesures vis-à-vis de l'industrie agroalimentaire, la consommation de sel et de graisses dans la population. La prévention médicamenteuse a de son côté fait des progrès considérables, mais son coût est élevé et sa prescription n'est pas toujours réalisée selon les recommandations existantes.

Il est aujourd'hui possible de déterminer le niveau de risque global d'un patient en fonction de ses facteurs cumulés. Il existe des références en matière de prévention, de dépistage des facteurs de risque et de prise en charge des patients. Elles constituent des outils efficaces et validés d'estimation du risque cardiovasculaire, mais ceux-ci résultent d'une approche épidémiologique et leur utilisation par le médecin reste insuffisante.

L'objectif étant de réduire le risque de survenue de maladies cardiovasculaires dans l'ensemble de la population en incitant à une alimentation équilibrée et une activité physique modérée ; les sujets à risque élevé, et seulement ceux-ci, devant faire l'objet d'une prise en charge médicamenteuse correctement prescrite.

Parmi les exemples d'interventions à l'échelle de la population pouvant être appliquées pour réduire l'occurrence des maladies cardiovasculaires figurent notamment :

— des stratégies complètes de lutte antitabac ;

— des politiques de taxation des produits alimentaires riches en graisses, en sucre et en sel ;

— l'aménagement de voies piétonnes et de pistes cyclables pour augmenter l'activité physique de la population ;

— des stratégies tendant à réduire l'usage nocif de l'alcool ;

— la fourniture de repas sains dans les écoles.

Au niveau de la prévention des premiers infarctus et AVC, les interventions individuelles doivent cibler les personnes présentant un risque cardiovasculaire total de moyen à élevé ou les personnes dont un facteur de risque dépasse les seuils recommandés pour le traitement comme le diabète, l'hypertension et l'hypercholestérolémie.

La première intervention (prise en compte intégrée du risque total) a un meilleur rapport coût/efficacité que la deuxième et pourrait réduire de manière substantielle les accidents cardiovasculaires. Cette approche est envisageable pour les soins de santé primaires dans les pays peu nantis, y compris en faisant appel à des professionnels de santé non-médecins.

En ce qui concerne la prévention secondaire des maladies cardiovasculaires pour les personnes souffrant d'une maladie avérée, dont le diabète, il convient d'appliquer le traitement thérapeutique.

En outre, il convient parfois de pratiquer des interventions chirurgicales coûteuses pour traiter les maladies cardiovasculaires.

Des dispositifs médicaux sont nécessaires pour traiter certaines maladies cardiovasculaires, à savoir : stimulateurs cardiaques, valves prothétiques et patches permettant d'obturer les défauts cardiaques [10].

1.8 Conclusion

Dans ce premier chapitre, nous avons d'abord décrit la structure, la morphologie et le fonctionnement du coeur puis nous avons donné un aperçu sur les maladies cardiovasculaires et leurs principaux facteurs de risque et enfin, nous avons rapporté les recommandations de l'OMS pour la prévention des maladies cardiovasculaires.

Dans le second chapitre nous présenterons l'état des connaissances de l'apprentissage automatique et de l'apprentissage profond dans le domaine de la santé en général et de la santé cardiovasculaire en particulier et les différentes approches d'aide au diagnostique préventif.

Chapitre 2

Etat des connaissances

2.1 Introduction

Dans les systèmes de santé, de grandes quantités de données sur les patients et de connaissances médicales sont stockées dans des bases de données et de nouveaux outils et technologies d'analyse et de classification des données sont nécessaires pour exploiter ces informations. Actuellement, les algorithmes Machine Learning (ML) sont utilisés pour l'analyse automatique des données médicales.

Les algorithmes et techniques déployés en ML peuvent être encadré dans un processus plus général connu sous le nom de découverte de connaissances dans des bases de données ou simplement d'exploration de données. Certaines de ces techniques étaient décrit il y a plus de 50 ans, mais ces dernières années, l'intérêt dans et autour d'eux a considérablement augmenté, en partie grâce à des progrès de la programmation algorithmique, augmentation de la capacité de traitement des ordinateurs modernes (unité de processeur graphique pour la vidéo et les graphiques cartes et unité de traitement tensoriel pour l'apprentissage neuronal), et la croissance de la disponibilité des données (Big Data ou Données Massives en anglais).

Au cours des dernières années, ces types d'algorithmes et de techniques ont commencé à être appliqué aux environnements cliniques, y compris dans les domaines de radiologie diagnostique, électrophysiologie cardiaque, diabète, dermatologie et psychiatrie. Compte tenu de leur caractère pratique, leur accessibilité et les résultats impressionnants obtenus jusqu'à présent.

2.2 Apprentissage automatique

C'est un ensemble de techniques donnant la capacité aux machines d'apprendre automatiquement un ensemble de règles à partir de données. Contrairement à la programmation qui

consiste en l'exécution de règles prédéterminées.

2.2.1 Types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage employé. Il existe plusieurs méthodes d'apprentissage automatique :

Apprentissage supervisé

En apprentissage supervisé, l'algorithme est guidé avec des connaissances préalables de ce que devraient être les valeurs de sortie du modèle. Par conséquent, le modèle ajuste ses paramètres de façon à diminuer l'écart entre les résultats obtenus et les résultats attendus. La marge d'erreur se réduit ainsi au fil des entraînements du modèle, afin d'être capable de l'appliquer à de nouveaux cas.

Apprentissage non supervisé

L'apprentissage non supervisé n'utilise pas de données étiquetées. Il est alors impossible à l'algorithme de calculer de façon certaine un score de réussite. Son objectif est donc de déduire les regroupements présents dans nos données. Prenons l'exemple, d'un jeu de données de fleurs, on recherche à les regrouper en classes. Ici, nous ne connaissons pas l'espèce de la plante, mais nous voulons essayer de les regrouper, par exemple, si les formes des fleurs sont similaire alors elles sont en rapport avec une même plante correspondante.

Apprentissage semi-supervisé

C'est une hybridation entre l'apprentissage supervisé et l'apprentissage non supervisé. Il est utilisé lorsque la base d'apprentissage est constituée d'un petit nombre de données étiquetées et d'un grand nombre de données non étiquetées. Il existe différentes méthodes telles que : Generative models, Low-density separation, Graph-based methods, Heuristic approaches [12], etc

Apprentissage par renforcement

L'apprentissage par renforcement est un domaine de l'intelligence artificielle dont le but est d'apprendre à suivre une politique optimale à partir d'un signal de supervision très faible, appelé récompense, qui n'indique que pour certains états ou actions à quel point ils sont bons ou mauvais en leur associant une valeur réelle plus ou moins importante.

2.3 Apprentissage profond

Le deep learning ou apprentissage profond est un sous-ensemble de l'apprentissage machine où les réseaux neuronaux artificiels, des algorithmes inspirés du cerveau humain, apprennent à partir de grandes quantités de données. De la même manière que nous apprenons par expérience, l'algorithme d'apprentissage profond exécuterait une tâche de manière répétitive, en la modifiant chaque fois un peu pour améliorer le résultat.

Nous parlons d'apprentissage profond parce que les réseaux neuronaux ont plusieurs couches (profondes) qui permettent l'apprentissage. Tout problème qui nécessite une "réflexion" pour être résolu est un problème que l'apprentissage profond peut apprendre à résoudre.

La quantité de données que nous générons chaque jour est stupéfiante - actuellement estimée à 2,6 quintillions d'octets et c'est cette ressource qui rend l'apprentissage profond possible. Comme les algorithmes d'apprentissage approfondi nécessitent une tonne de données pour apprendre. En plus de la création de données, les algorithmes d'apprentissage approfondi bénéficient d'une puissance de calcul plus importante aujourd'hui, ainsi que de la prolifération de l'intelligence artificielle en tant que service.

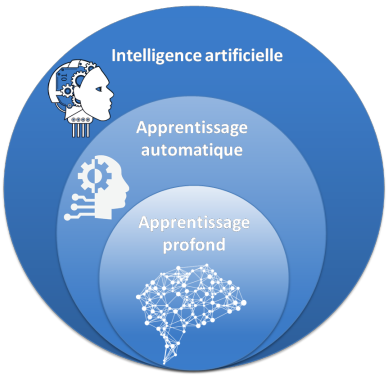


FIGURE 2.1 – Apprentissage profond

2.4 Réseaux de neurones

Avant de décrire les réseaux de neurones artificiels, nous allons très brièvement décrire le neurone biologique.

2.4.1 Neurone biologique

Le neurone est une cellule fondamentale du système nerveux des êtres vivants. Le cerveau humain en contient plusieurs dizaines de milliards. Chaque neurone est composé :

— d'un corps cellulaire contenant le noyau,

— de dendrites, nombreuses et ramifiées, qui conduisent l'influx nerveux de leur périphérie jusqu'au corps cellulaire,

— d'un axone, qui conduit le potentiel d'action émis au niveau du corps cellulaire jusqu'aux dendrites d'autres neurones. L'influx nerveux y est alors transmis par voie chimique au niveau des synapses. Les axones peuvent mesurer plusieurs dizaines de centimètres de longueur et sont entourés de gaines de myéline permettant d'optimiser la transmission de l'influx nerveux.

Chaque neurone reçoit donc en entrée des signaux en provenance d'autres neurones, transmis par les dendrites jusqu'au corps cellulaire où ils s'additionnent. L'importance de chaque signal reçu est modulée à la fois par la longueur de la dendrite lui permettant d'atteindre le corps cellulaire et par l'efficacité de la liaison synaptique entre l'axone présynaptique et la dendrite post-synaptique. S'il dépasse un certain seuil, le signal résultant au niveau du corps cellulaire peut alors donner lieu à un potentiel d'action, c'est-à-dire un pic de potentiel électrique qui se propage à travers l'axone jusqu'aux autres neurones dont les dendrites sont connectées à cet axone : on dit que ce neurone décharge.

En 1949, Donald Hebb postule que les capacités d'apprentissage du cerveau résultent d'une règle très simple, souvent résumée par "des neurones qui déchargent en même temps sont des neurones qui se lient ensemble" [13]. Ainsi, si un neurone A décharge régulièrement juste avant un neurone B, alors un mécanisme biochimique accroît l'efficacité de la cellule A à induire un potentiel d'action dans la cellule B. Ce mécanisme est désigné sous le nom de plasticité synaptique. Dans la règle de Hebb originale, cette précédence temporelle de l'activité de A sur celle de B est essentielle et a été mise en évidence expérimentalement [14]. Elle s'accompagne également de l'effet inverse : si le neurone A décharge régulièrement juste après le neurone B, alors la liaison entre A et B est affaiblie (apprentissage anti-hebbien). Dans les modèles computationnels les plus abstraits, cette précédence temporelle est souvent écartée et la règle se transforme en une association simple entre deux neurones qui sont actifs régulièrement aux mêmes pas de temps de la simulation [15].

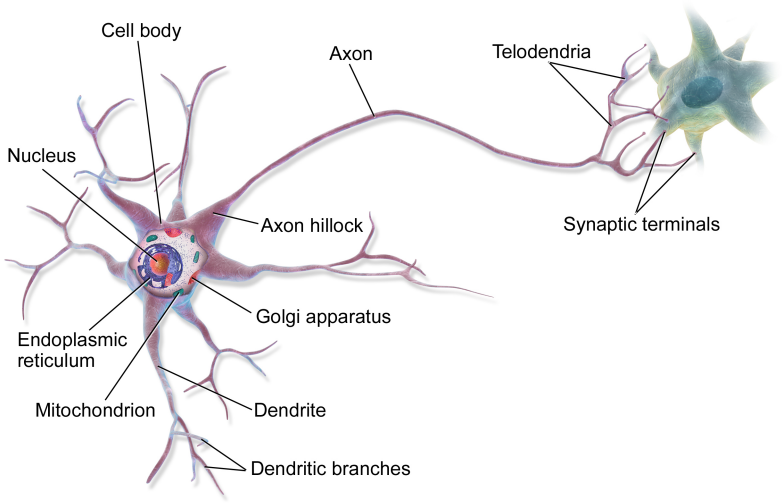


FIGURE 2.2 – Neurone biologique [3]

2.4.2 Neurone formel

En 1943, Warren McCulloch et Walter Pitts proposent un modèle mathématique très simplifié du neurone biologique (McCulloch et Pitts 1943). Il s'agit du premier modèle de neurone formel.

Étant donnés un ensemble d'entrées $x \in \mathbb{R}^n$ et une sortie $y \in \mathbb{R}$, le neurone formel de McCulloch et Pitts associe un poids w_i à chaque entrée x_i et calcule la somme pondérée des entrées par leurs poids respectifs à laquelle s'ajoute un biais b. Le résultat est alors transformé par une fonction d'activation non linéaire σ :

$$y = \sigma\left(\sum_i w_i x_i + b\right)$$

Dans la version originelle, la fonction d'activation est la fonction de Heaviside :

$$H(x) = \begin{cases} 1 & si \quad x \geq 0 \\ 0 & si \quad x < 0 \end{cases}$$

mais d'autres fonctions ont été largement utilisées, comme la fonction sigmoïde

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}$$

Un réseau de neurones est organisé en trois parties :

— La première couche d'un réseau de neurones est celle d'entrée (Input Layer). C'est par cette couche que vont rentrer les données dont vous disposez. Avant de pouvoir "nourrir" le réseau.

— La couche finale, dite de sortie (Output Layer), va vous fournir la classification.

— Et toutes les couches entre la couche d'entrée et de sortie, des couches dites "cachées" (Hidden Layers), sont autant de représentations différentes des données.

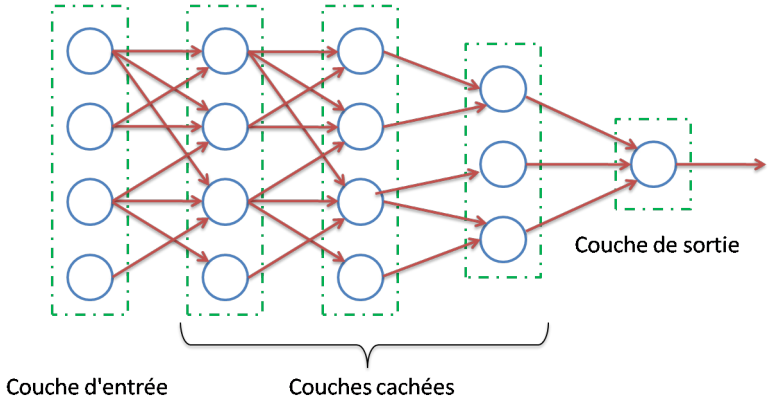


FIGURE 2.3 – Neurone formel

De nombreux modèles de neurones formels ont depuis été développés, du plus simple au plus complexe et du plus abstrait au plus réaliste [15].

2.4.3 Modèles Neuronaux de base

Perceptron

En 1958, Frank Rosenblatt propose de doter les réseaux de neurones d'une règle d'apprentissage supervisé inspirée de l'apprentissage Hebbien, à ceci près que l'activité post-synaptique est remplacée par l'erreur entre l'activité post-synaptique souhaitée y et celle $\hat{y}$ obtenue en sortie du réseau [15] :

$$\Delta w_i \infty (y - \hat{y}) x_i$$

Cette règle permet d'apprendre un classifieur linéaire qui sépare l'espace d'entrée par un hyperplan.

Perceptron multicouches

En empilant plusieurs perceptrons, on obtient un perceptron multicouches. Chaque neurone de chaque couche se comporte toujours comme un classifieur linéaire, mais l'utilisation de couches intermédiaires permet de créer des partitions complexes de l'espace. Ceci permet de projeter les données fournies en entrée dans de nouveaux espaces, dans lesquels une tâche initialement non linéaire peut devenir linéaire.

Cependant, l'utilisation de couches intermédiaires rend impossible d'entraîner ces réseaux en utilisant la règle d'apprentissage du perceptron. C'est pourquoi il a fallu attendre la publication des techniques de rétropropagation du gradient [16–18] et plus particulièrement [19] pour que ces réseaux soient plus largement utilisés. Comme nous le verrons par la suite, ces techniques nécessitent en particulier que les fonctions d'activation utilisées soit dérivables. À partir de ce moment là, la fonction de Heaviside a donc été généralement remplacée par la fonction sigmoïde [15].

2.4.4 Apprentissage des réseaux de neurones

Reproduire les capacités d'apprentissage humaine est, sans doute, l'une des ambitions les plus importantes de la modélisation des réseaux de neurones artificiels. L'apprentissage est alors l'une de leurs propriétés fondamentales. C'est le processus permettant au réseau de se spécialiser sur un problème spécifique à partir de son expérience. Il consiste généralement à modifier les poids synaptiques jusqu'à ce que le réseau puisse effectuer la tâche désirée. Il s'agit de configurer les valeurs des poids synaptiques censés stocker les informations acquises. D'une façon générale, l'apprentissage se traduit par une modification dans la valeur des poids reliant les neurones du réseau. Chaque poids w_{ij} reliant un neurone i à un autre neurone j à l'itération (r) est modifié selon l'équation générale suivante :

$$w_{ij}(r) = w_{ij}(r-1) + \Delta w_{ij}(r-1)$$

Où : $w_{ij}(r)$ et $w_{ij}(r-1)$ sont respectivement les valeurs de ce poids à la r ème et la $(r-1)$ ème itération et $\Delta w_{ij}(r-1)$ est le changement correspondant [20].

2.5 Fouille de données (Data Mining)

L'exploration de données ou fouille de données, a pour objet l'extraction d'un savoir ou d'une connaissance à partir de grandes quantités de données stockées dans des bases ou des entrepôts de données. C'est un outil qui permet de trouver des structures originales et des corrélations informelles entre les données. Le Data Mining permet de mieux comprendre les liens entre des phénomènes en apparence distincts et d'anticiper des tendances encore peu discernables.

2.5.1 Processus de fouille de données

Le processus complet de fouille de données comprend plusieurs étapes [21] :

1. collecte des informations et organisation de ces informations dans une base de données ;

2. nettoyage de la base de données : attributs sans valeur, ayant une valeur invalide (bruit) ; doublons ; normalisation ;
3. sélection des attributs utiles ;
4. extraction d'information d'une base de données (Knowledge Discovery in Databases, ou KDD) ;
5. visualisation des données : histogramme, camembert, arbre, visualisation 3D et plus généralement, exploration interactive de données ;
6. évaluation des résultats de l'extraction de connaissance.

Le processus peut être résumé en trois phases successives : la phase prétraitement des données (étapes : 1, 2 et 3). La phase découverte des phénomènes fréquents par la fouille des données (étape 4). Finalement, la mise en forme et l'évaluation des connaissances extraites (étape 5 et 6).

Prétraitement des données

Cette étape consiste à préparer les données afin d'être exploitées. Elle permet d'améliorer la qualité des données fouillées par les algorithmes d'apprentissage automatiques. Il s'agit du :

Filtrage des données manquantes : Il arrive assez fréquemment que des observations soient incomplètes, c'est à dire les valeurs d'une ou plusieurs variables manquent. Dans les situations de données manquantes, on emploiera plus généralement l'expression incomplétude de données.

Lorsque les observations à données manquantes font partie des observations employées pour la construction d'un modèle, une solution simple est d'ignorer ces observations incomplètes. Bien que fréquemment employée par défaut, cette solution s'avère simpliste car ses conséquences peuvent être, non seulement une diminution des performances du modèle, mais potentiellement un fort biais de modélisation pouvant conduire à un modèle inopérant.

Filtrage des doublons : Il permet l'identification et l'élimination des instances dupliquées dans une base de données.

Normalisation des données : L'ensemble de données sur les maladies cardiaques D^{hd} contient un certain nombre de caractéristiques, et chaque caractéristique comprend des valeurs numériques différentes, ce qui augmente les difficultés lors du processus de calcul. Par conséquent, une technique de normalisation est utilisée pour normaliser l'ensemble des données

D^{hd} dans une plage comprise entre zéro et 1, ainsi que pour diminuer la complexité numérique pendant le processus de calcul de la prédiction des maladies cardiaques.

Plusieurs méthodes peuvent être utilisées pour la normalisation des données. Dans le système proposé, la méthode bien connue de normalisation min-max est utilisée [22, 23]. Cette méthode place une valeur numérique, DV, de l'ensemble de données original D^{hd} en DV_{norm} dans l'intervalle [0, 1] en utilisant la méthode de normalisation min-max. intervalle [0, 1] en utilisant l'équation suivante :

$$DV_{norm} = \frac{D^{hd} - DV_{min}}{DV_{max} - DV_{min}} \times [new\_max - new\_min] + new\_min$$

Ici, DV_{norm}, D^{hd}, DV_{min}, et DV_{max} sont la valeur de données normalisée, la valeur de données d'origine, la valeur de données minimale et la valeur de données maximale, respectivement, dans l'ensemble de données, tandis qu'ils indiquent la plage de la valeur convertie. valeur de données maximale, respectivement, dans l'ensemble de données, tandis que new_max et new_min indiquent la plage de l'ensemble de données converti. l'ensemble de données converti. Nous utilisons $new\_max = 1$ et $new\_min = 0$. Avec cette méthode, toutes les valeurs des caractéristiques sont comprises dans l'intervalle [0, 1].

Sélection d'attributs : La sélection de caractéristique (ou sélection d'attribut ou de variable) est un processus utilisé en apprentissage automatique et en traitement de données. Il consiste, étant donné des données dans un espace de grande dimension, à trouver un sous-ensemble de variables pertinentes.

2.5.2 Les méthodes du processus de fouille de données

Les différentes méthodes du Data Mining se résument, brièvement, comme suit [24] :

Classification : examiner les caractéristiques d'un objet et lui attribuer une classe.

Prédiction : prédire la valeur future d'un attribut en fonction d'autres attributs.

Association : déterminer les attributs qui sont corrélés (associés).

Segmentation : former des groupes homogènes à l'intérieur d'une population.

2.6 Sélection d'attributs

Les données des applications du monde réel peuvent être de haute dimension. Cela est particulièrement vrai pour les applications dans les domaines de la médecine [25].

L'identification de caractéristiques informatives est devenue une étape importante de l'exploration de données, non seulement pour contourner la malédiction de la dimension mais aussi pour réduire la quantité de données à traiter. En général, la sélection de caractéristiques réduit le nombre de caractéristiques tout en conservant des performances d'apprentissage identiques, voire meilleures [26]. Ses avantages ont été démontrés dans diverses applications d'exploration de données et d'apprentissage automatique [27, 28].

Lorsque les caractéristiques redondantes, non pertinentes et bruyantes sont supprimées de l'ensemble de données d'apprentissage, l'efficacité de l'apprentissage s'en trouve améliorée.

2.6.1 Etat de l'art de la sélection d'attributs dans le domaine biomédical

Les bases de données biomédicales sont souvent représentées par un grand nombre de caractéristiques de la maladie et un nombre relativement faible de dossiers des patients. Ces caractéristiques (attributs) ne sont pas toutes pertinentes et peuvent être source de bruit. Plusieurs travaux de recherche ont été réalisés pour remédier à cette malédiction de dimension. Ces travaux peuvent être divisés en plusieurs axes de recherches indépendants :

— **Traitement des données médicales** : Afin de démontrer l'utilité de la sélection des attributs dans le domaine biomédical, plusieurs recherches ont été testées sur les différents ensembles de données médicales disponibles sur le net, comme par exemple le répertoire de « UCI Machine Learning ». Dans [29], les auteurs proposent une technique qui recherche une division stratégique de l'espace des caractéristiques dans le but d'identifier les meilleurs sous-ensembles de caractéristiques pour chaque instance. Cette technique est basée sur l'approche Wrapper, où un algorithme de classification est utilisé comme fonction d'évaluation pour différencier entre plusieurs sous-ensembles d'attributs. Dans [30], des règles d'association et de corrélation des caractéristiques ont été utilisées afin de réduire la dimensionnalité dans le domaine médical. Dans [31] les auteurs se sont basé sur des méthodes d'hybridation des Particules Swarm Optimization (PSO) et la théorie des ensembles.

— **La recherche et analyse de documents médicaux** : Les documents textuels sont généralement représentés comme une matrice attribut-document. Les attributs peuvent être des mots simples à partir du document de texte ou paires plus complexes extraites afin d'enrichir la représentation matricielle [32]. Dans [33] une approche wrapper basée sur la recherche d'information sémantique est proposée. La similarité sémantique consiste à calculer la similarité entre les termes conceptuellement similaires, mais lexicalement dissemblables. D'autres applications dans le domaine biologique ont été revues dans [34].

— **L'imagerie médicale** : La sélection d'attributs dans l'imagerie est aussi très importante [35]. Dans [36] les auteurs ont passé en revue les différentes méthodes qui existent dans la littérature pour la détection du cancer du sein.

— **L'analyse et la prédiction des séquences qui codent les protéines** : Sachant que de nombreuses caractéristiques peuvent être extraites d'une séquence et la plupart des dépendances se produisent entre des positions adjacentes. Pour faire face à la grande quantité d'attributs possibles, et la quantité souvent limitée d'échantillons, les auteurs de [37] ont présenté le modèle interpolé Markov. Ils ont utilisé l'interpolation entre les différents ordres du modèle de Markov en se basant sur des échantillons de petite taille et une méthode de filtrage (chi 2) dans l'objectif de sélectionner les caractéristiques pertinentes. Saeys et al. [38], ont combiné différentes mesures de codage de prédiction potentiel, et ont ensuite utilisé une approche de filtre multivariée pour la couverture de Markov et ainsi ne conserver que les attributs les plus pertinents. Parallèlement de nombreuses méthodes d'analyse de séquence comprennent la reconnaissance de signaux courts, plus ou moins conservées dans la séquence, ce qui représente principalement des sites de liaison pour diverses protéines [39]. Une approche commune pour trouver des motifs réglementaires, est de relier des motifs à des niveaux d'expression des gènes en utilisant une approche de régression. La sélection des fonctionnalités peut ensuite être utilisé pour rechercher les motifs qui maximisent l'ajustement du modèle de régression [40]. Dans [41], les auteurs démontrent les avantages d'employer la sélection d'attributs, en utilisant l'entropie caractéristique de classe comme filtre pour éliminer les caractéristiques non pertinentes.

— **Traitement des données biopuces** : L'avènement des jeux de données de puces à ADN a stimulé une nouvelle ligne de recherche en bioinformatique. Les données de biopuces constituent un grand défi pour les techniques de calcul, en raison de leur grande dimension (jusqu'à plusieurs dizaines de milliers de gènes) et de leurs petites tailles d'échantillon [42]. En outre, des complications expérimentales supplémentaires comme le bruit et la variabilité rendent l'analyse des données de puces à ADN un domaine très passionnant. Afin de faire face aux caractéristiques particulières de ces données, la nécessité évidente de réduction de dimension a fait objet de plusieurs recherches scientifiques [43–46]. En raison de la haute dimensionnalité de la plupart des analyses micropuces, les techniques de sélection rapides et efficaces, telles que les méthodes de filtrage univariées, ont attiré plus d'attention.

2.7 Classification supervisée

La classification supervisée est une technique largement utilisée avec différentes applications dans la vie réelle [47]. Elle permet de générer des règles de classification (modèle) à partir d'un jeu données classées à priori et d'un algorithme d'apprentissage automatique adéquat. Ces règles seront utilisées pour classer les nouvelles instances.

Soit D un ensemble de n exemples et de m attributs ($n \times m$). $Y = \{y_1, y_2, ..., y_p\}$ est l'attribut classe avec p valeurs classes possibles. Chaque instance $x_i \in D$ est caractérisée par m attributs et par sa classe $y_i \in Y$.

L'objectif est, en s'appuyant sur l'ensemble d'exemples étiqueté X et un classificateur Cl, de prédire la classe des nouvelles instances.

$$Cl(x) = y,\ x \textit{ est une nouvelle instance non étiquetée. } y \in Y$$

Une classification est dite binaire, si le nombre de classes $|Y|$ est égale à 2. Le classificateur doit prédire l'une des deux classes pour les nouvelles instances. Une classification est dite multi classes si le nombre de classes $|Y| > 2$.

Pour évaluer un processus Data Mining, comparer les algorithmes d'apprentissage et améliorer les performances sur un jeu de données, ils existent différentes techniques d'évaluation, telles que holdout, k-cross validation et leave-one-out [24].

Holdout : La méthode holdout partitionne le dataset D_n de n instances en deux parties. Une partie D_k pour l'apprentissage et une partie D_t pour le test, généralement $D_k = \frac{2}{3}$ et $D_t = \frac{1}{3}$ [48].

K-cross validation : Appelée estimation rotative [48], le dataset D est divisé aléatoirement en k sous ensembles mutuellement exclusifs $D_1, D_2, ..., D_k D = D_1 \cup D_2 ... \cup D_k$. Le classificateur est testé k fois. A chaque $t \in \{1, 2, ..., k\}$, l'apprentissage se fait sur le dataset $D - D_t$ et testé sur D_t. La performance finale P est la moyenne de toutes les performances P_t. Elle est calculée comme suit :

$$P = \frac{\sum_{t=1}^{k} P_t}{k}$$

Leave-one-out : La méthode leave-one-out est tout simplement la méthode k-cross validation avec $k = n$, n étant le nombre d'instances.

2.8 Algorithmes de classification

Il existe différents algorithmes pour l'extraction de la connaissance selon l'objectif et le type d'apprentissage. Nous allons passer en revue les approches utilisées pour la prédiction des maladies cardiovasculaires [49, 50].

2.8.1 Multi-Layer Perceptron (MLP)

Le perceptron multicouche est un réseau de neurones artificiels à anticipation qui se compose d'un certain nombre de neurones connectés par des poids de liaison. MLP mappe un groupe d'entrées dans un ensemble de sorties souhaitées. La structure de MLP est illustrée à la figure ci-dessous.

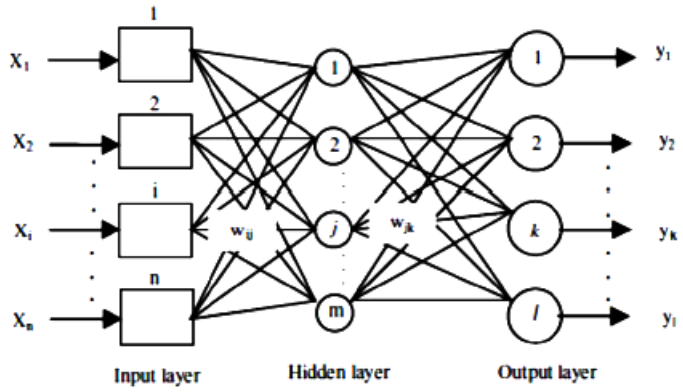


FIGURE 2.4 – Architecture du perceptron multicouche [4]

MLP se compose de trois parties principales d'une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée reçoit les données d'entrée de l'extérieur, puis les transmet à la première couche cachée, qui sera transmise jusqu'à ce qu'elle atteigne enfin la couche de sortie. Ce processus est communément appelé passe avant [51].

L'entrée et la sortie sont accessibles directement, mais pas la couche cachée. Chaque couche est constituée de plusieurs neurones. Les neurones sont connectés entre différentes couches en utilisant le poids et le biais. La sortie du neurone j dans la couche cachée est obtenue à partir de l'équation suivante :

$$H_j = f(\sum_{i=1}^{n} w_{ij} x_i + b_i)$$

où w_{ij} et b_i sont les poids et les biais des neurones de la couche cachée, et f (.) est une fonction d'activation non linéaire (fonction de transfert sigmoïde tangente hyperbolique (tansig)).

La sortie du réseau est illustrée dans l'équation suivante :

$$y = f(\sum_{j=1}^{m} w_{kj} H_i + b_0)$$

où f (.), W_{kj} et b_0 sont la fonction d'activation des neurones de la couche de sortie (fonction de transfert sigmoïde tangente hyperbolique (tansig)), le poids et le biais [52].

2.8.2 Support Vector Machine (SVM)

Le classificateur SVM, développée par Vladimir Vapnik en 1995, est un classificateur puissant, il a fait ses preuves dans plusieurs domaines. Le principe est de projeter les données qui sont non linéairement séparables dans un autre espace de dimension plus élevée où elles peuvent le devenir, en utilisant différents noyaux. Le but du SVM binaire est de trouver un hyperplan optimal qui sépare les deux classes en maximisant la distance. Cette distance est appelée marge. Dans le cas d'une classification binaire, l'hyperplan est une droite. Les points les plus proches, qui seuls sont utilisés pour la détermination de la marge, sont appelés vecteurs de support.

L'hyperplan séparateur est représenté par l'équation :

$$H(x) = w^T x + b$$

w est un vecteur de m dimensions et b est un terme [53]. La fonction de décision, pour un exemple x, peut être exprimée comme suit :

$$\begin{cases} Classe = 1 & Si \quad H(x) > 1 \\ Classe = -1 & Si \quad H(x) < -1 \end{cases}$$

Maximiser la marge revient à maximiser $\frac{2}{\|w\|}$ et qui vaut à minimiser $\|w\|$.

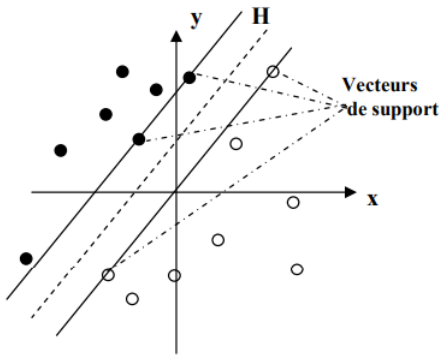


FIGURE 2.5 – SVM classification binaire [5]

SVM réduit le problème multi classe à une composition de plusieurs hyperplans bi-classe

permettant de tracer les frontières de décision entre les différentes classes. Il décompose l'ensemble d'exemples en plusieurs sous-ensembles représentant chacun un problème de classification binaire. A chaque fois un hyperplan de séparation est déterminé par la méthode SVM binaire. On construit lors de la classification une hiérarchie des hyperplans binaires qui est parcourue de la racine jusqu'à une feuille pour décider de la classe d'un nouvel exemple [53].

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de redescription de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien [54].

2.8.3 Logistic Regression (LR)

La régression logistique est un algorithme de classification utilisé pour répartir les observations dans un ensemble discret de classes. Il est classé dans les types de niveau binaire, multi et normal. LR n'indique pas une relation entre attributs non continus, mais permet la prédiction des variables discrètes [55]. Il est très facile à mettre en œuvre et assez efficace pour entraîner le modèle.

La régression logistique s'écrit mathématiquement sous la forme d'une fonction de régression linéaire multiple :

$$Logit(P) = \left(\frac{m(x=1)}{1-(p=1)}\right) = \beta + \beta_1 \times x_1 + \beta_2 \times x_2 - -\beta_i \times x_m \ for\ i = 1...N$$

L'exemple suivant représente une fonction binaire logistique simple :

$$\begin{cases} Si \quad positif \quad alors \quad 1 \\ Si \quad négatif \quad alors \quad 0 \end{cases}$$

$$Hypothesis\ W = AX + B$$

$$H(x) = sig(W)$$

Si W atteint l'infini positif, alors la prédiction est positive, et si W atteint l'infini négatif, alors la prédiction est négative.

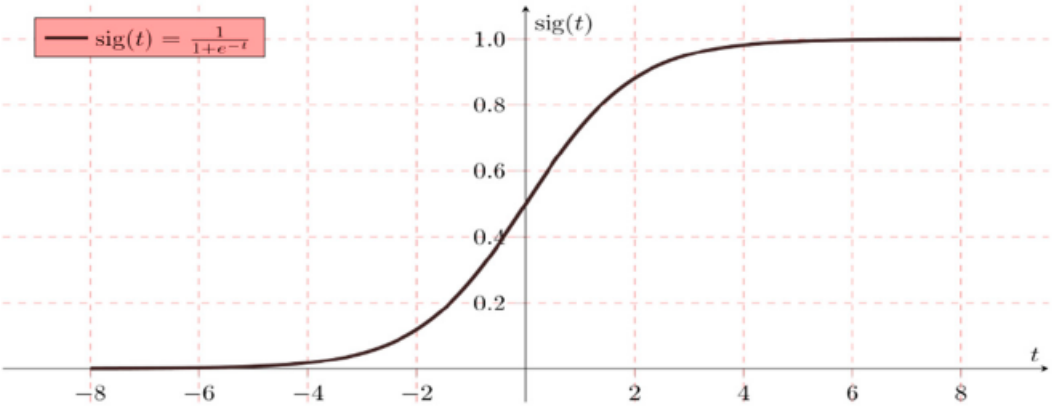


FIGURE 2.6 – Représentation de régression logistique binaire simple (où sig (t) fonction d'activation sigmoïde) [6]

2.8.4 Random Forest (RF)

La forêt aléatoire est une extension de la méthode d'ensachage [56], une méthode d'apprentissage d'ensemble typique. Le principe de la méthode d'ensachage est le suivant : étant donné un ensemble de données contenant m échantillons, un échantillon est sélectionné au hasard et placé dans l'ensemble d'échantillons, puis l'échantillon est remis dans l'ensemble de données initial, de sorte que l'échantillon puisse encore être sélectionné au moment d'échantillonnage suivant. De cette façon, après m opérations d'échantillonnage aléatoire, nous obtenons un ensemble d'échantillons avec m échantillons. Certains échantillons de l'ensemble d'apprentissage initial apparaissent plusieurs fois dans l'ensemble de rééchantillonnage, et d'autres n'apparaissent jamais. T échantillons contenant m échantillons d'apprentissage sont sélectionnés, puis un apprenant de base est formé sur la base de chaque ensemble d'échantillons, puis ces apprenants de base sont combinés. L'ensachage utilise généralement une méthode de vote simple pour les tâches. L'apprenant de base de RF est un arbre de décision, et la sélection d'attributs aléatoires est introduite dans le processus d'apprentissage de l'arbre de décision. La RF est simple, compréhensible, peu coûteuse en calcul et a obtenu des performances puissantes dans de nombreuses tâches du monde réel, connues sous le nom de « méthodes représentant le niveau de la technologie d'apprentissage d'ensemble ». La RF a été appliquée à la sélection de gènes, à la classification par télédétection, à la reconnaissance d'images et à la prédiction de maladies, entre autres, et a obtenu de bons résultats [57–60].

2.8.5 Decision Tree (DT)

Les arbres de décisions sont des techniques très populaires par leur efficacité et leur simplicité dans le domaine de la classification supervisée. Ils fournissent une représentation graphique du modèle facilement interprétable [61]. Le modèle final est constitué d'un noeud racine et des noeuds intermédiaires, des branches et des feuilles. La racine est le point d'entrée à l'arbre. Les feuilles représentent les valeurs classes à prédire. Les branches représentent les résultats de test relatif à chaque noeud. Pour effectuer une classification, l'arbre est parcouru de la racine aux feuilles selon une série de tests à chaque niveau de l'arbre. La théorie de Shannon est à la base de partitionnement de plusieurs arbres de décision. Elle est définie comme suit :

La quantité d'information associé au noeud x est :

$$I(x) = -\sum_{i} P(x_i) \log P(x_i)$$

$$P_i = \frac{n_i}{n_s}$$

n_j représente le nombre d'instances appartenant à la classe j et n_s représente le nombre total d'instance du noeud s.

Le gain d'information est mesuré par la différence entre l'impureté du noeud parent s et la somme des impuretés des p noeuds fils obtenus grâce à un attribut X [62].

$$Gain(s, X) = I(s) - \sum_{i=1}^{p} \frac{n_i}{n} I(s_i)$$

n_j représente le nombre d'instances total du noeud s et n représente le nombre d'instances total du noeud parent.

2.8.6 Naive Bayes (NB)

Naïve Bayes est un classificateur probabiliste simple. Il calcule un ensemble de probabilités en comptant la fréquence et les combinaisons de valeurs dans un jeu de données. L'algorithme utilise le théorème de Bayes et suppose que tous les attributs sont indépendants compte tenu de la valeur de la variable classe. Cette hypothèse d'indépendance conditionnelle est rarement valable dans les applications du monde réel, d'où la caractérisation naïve. Cependant, l'algorithme tend à bien fonctionner et à apprendre rapidement dans divers problèmes de classification supervisée [63].

Compte tenu de la classe y et du vecteur de données $(x_1, x_2, x_3, ..., x_n)$, le théorème de Bayes énonce la relation suivante :

$$P(y|x_1, x_2, x_3, ..., x_n) = \frac{P(y)P(x_1, x_2, x_3, ..., x_n|y)}{P(x_1, x_2, x_3, x_n)}$$

$$P(x_1, x_2, x_3, ..., x_n|y) = \prod_{i=1}^{n} P(x_i|y)$$

Pour tous les x_i, cette relation est simplifiée comme suit :

$$P(y|x_1, x_2, x_3, ..., x_n) = \frac{P(y)\prod_1^n P(x_i|y)}{P(x_1, x_2, x_3, ..., x_n)}$$

Puisque $P(x_1, x_2, x_3, ..., x_n)$ est constant pour l'entrée, nous pouvons utiliser la règle de classification suivante :

$$y = \arg\max(P(y)\prod_{i=1}^{n} P(x_i|y))$$

2.8.7 K-Nearest Neighbors (KNN)

K-Plus Proche Voisin est une méthode anti-paramétrique, qui est utilisée pour la régression et la classification. Il s'agit essentiellement d'une méthode de regroupement, qui considère la distance entre un point et les coordonnées (x, y) et ses voisins. La distance euclidienne est dans l'équation ci-dessous et ses voisins sont déterminés à partir du point et éventuellement situés dans la région la plus proche de ses points voisins [64].

$$distance\ D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Ici, x, y sont deux points dans l'espace Euclidien 'n', xi, yi sont deux vecteurs partant du point initial, 'n' signifie un espace à n dimensions.

2.8.8 eXtreme Gradient Boosting (XGBoost)

XGBoost est un type d'apprentissage automatique supervisé utilisé pour la classification et la modélisation de la régression [65]. XGBoost est un algorithme amélioré basé sur l'implémentation de renforcement de gradient DTs avec plusieurs modifications en termes de régularisation, de fonction de perte et d'échantillonnage de colonnes. Le renforcement de gradient est une technique dans laquelle de nouveaux modèles sont créés et utilisés pour prédire l'erreur ou les résidus, après quoi les scores sont additionnés pour obtenir le résultat final de la prédiction. La méthode de descente de gradient est utilisée pour minimiser le score de perte lorsque de nouveaux modèles sont créés. La fonction objective doit être utilisée pour mesurer la performance du modèle, qui se

compose de deux parties : la perte de formation et la régularisation. Le terme de régularisation pénalise la complexité du modèle et empêche le sur-ajustement. La fonction objectif (fonction de perte et régularisation) peut être présentée comme suit.

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k);$$

où

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

Le terme l est ici la fonction de perte convexe différentiable qui calcule la différence entre la prédiction $\hat{y}_i$ et la cible y_i. Alors que le terme régularisé Ω pénalise la complexité du modèle et le nombre de feuilles de l'arbre sont représentés par T. De plus, chaque f_k correspond à une structure d'arbre indépendante q et au poids de la feuille w. Enfin, le terme γ correspond au seuil et le pré-élagage est effectué lors de l'optimisation pour limiter la croissance de l'arbre et λ est utilisé pour lisser les poids finaux appris afin d'éviter le sur-ajustement [66].

2.9 Techniques d'évaluation

2.9.1 Matrice de confusion

Dans le contexte de la classification supervisée, la matrice de confusion, appelée aussi matrice de contingence, est un outil qui sert à évaluer les performances d'un algorithme de classification. Elle synthétise les informations sur les classes réelles et les classes prédites par le modèle. Les colonnes de la matrice représentent les classes estimées et les lignes représentent les classes réelles des instances testées. Différentes métriques sont calculées à partir de la matrice de confusion [57, 67].

Classes actuelles	**Classes prédites**	
	Positif	**Négatif**
Positif	Vrai Positif (TP)	Faux Négatif (FN)
Négatif	Faux Positif (FP)	Vrai Négatif (TN)

TABLE 2.1 – Matrice de confusion pour une classification supervisée binaire

2.9.2 Métriques d'évaluation

Différentes mesures de performance sont utilisées pour déterminer l'efficacité des modèles prédictifs.

L'accuracy (Acc) représente la capacité de prédiction globale du modèle d'apprentissage en profondeur proposé et des modèles ML. Les vrais positifs (TP) et les vrais négatifs (TN) mesurent la capacité des modèles de classification à prédire l'absence et la présence d'une maladie cardiovasculaire chez le patient. Les faux positifs (FP) et les faux négatifs (FN) identifient le nombre de fausses prédictions générées par les modèles.

La précision (Pre) et le rappel (Rap) mesurent respectivement le succès et la sensibilité du modèle de classification des maladies cardiovasculaires. La fonction de mesure (FM) est utilisée pour déterminer les performances de prédiction [49].

$$Accuracy\ (Acc) : \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$Précision\ (Pre) : \frac{(TP)}{(TP + FP)}$$

$$Rappel\ (Rap) : \frac{(TP)}{(TP + FN)}$$

$$F - mesure\ (FM) : 2 \times \frac{(Précision \times Rappel)}{(Précision + Rappel)}$$

2.9.3 Métriques d'erreur

Les métriques d'erreur sont les mesures les plus utilisées dans la littérature. Elles évaluent la qualité des prédictions générées [68]. Il existe plusieurs types de métriques d'erreur qui consistent à mesurer la « distance » moyenne entre les prévisions et les observations correspondantes. Ainsi, une valeur proche de 0 indique des prédictions parfaites et une valeur avoisinante de 1 désigne de mauvaises prédictions [68–72]. Pour mieux illustrer les mesures qui vont suivre, on suppose que w_i représente la valeur observée et r_i représente la valeur prédite pour un ensemble de B_i prévisions. Ainsi, on peut calculer :

L'erreur absolue moyenne (MAE : Mean Absolute Error) : cette mesure est la métrique d'erreur la plus populaire. Elle évalue la qualité des prédictions fournies. À cet effet, le MAE mesure la déviation absolue moyenne entre une estimation prévue et l'estimation vraie de l'utilisateur [73–78].

$$\left|\bar{E}\right| = \frac{\sum_{b_k \in B_i} |r_i(b_k) - w_i(b_k)|}{|B_i|}$$

La racine de l'erreur quadratique moyenne (RMSE : Root Mean Squared Error) : dans la littérature, le RMSE est largement utilisé, à la place du MSE, pour l'évaluation. Il est utilisé par le fameux concours Netflix Prize [1] pour identifier les meilleurs algorithmes de filtrage [79–83].

$$\left|\bar{E}\right| = \sqrt{\frac{\sum_{b_k \in B_i} (r_i(b_k) - w_i(b_k))^2}{|B_i|}}$$

L'erreur absolue moyenne (MAE) et la racine de l'erreur quadratique moyenne (RMSE) mesurent la variation absolue et la différence entre les valeurs réelles et les valeurs prédites, respectivement.

2.9.4 Courbe ROC et AUC

Les courbes ROC (Receiver Operating Characteristic) résument le compromis entre le taux de vrais positifs et le taux de faux positifs pour le modèle prédictif en utilisant différents seuils de probabilité.

Une aire sous la courbe ROC (ou Area Under the Curve, AUC) de 0.5 (50%) indique que le marqueur est non-informatif. Une augmentation de l'AUC indique une amélioration des capacités discriminatoires, avec un maximum de 1.0 (100%).

2.10 Datasets d'entraînement utilisés pour la prédiction des maladies cardiovasculaires

Les modèles existants ont été testés avec deux ensembles de données sur les maladies cardiovasculaires : Cleveland et hongrois. Ces ensembles de données sont extraits du référentiel d'exploration de données et d'exploration de données en ligne de l'Université de Californie, Irvine (UCI) [84–87]. L'ensemble de données original de Cleveland se compose de 303 cas avec 76 caractéristiques. L'ensemble de données hongrois comprend 294 cas avec 14 caractéristiques. Les deux ensembles de données ont des cas avec des valeurs manquantes.

	Base de données de Cleveland	**Base de données hongroise**
Nombre de cas	303	294
Nombre d'attributs	76	14

TABLE 2.2 – Les bases de données de Cleveland et hongroise

1. Le prix Netflix était un concours ouvert pour le meilleur algorithme de filtrage collaboratif pour prédire les notes des utilisateurs pour les films, sur la base des notes précédentes sans aucune autre information sur les utilisateurs ou les films, c'est-à-dire sans que les utilisateurs soient identifiés sauf par des numéros attribués pour le concours. Le concours était organisé par Netflix (http ://www.netflixprize.com/).

2.11 Travaux connexes

Cette section donne un aperçu des travaux de recherche déjà effectués dans la prédiction des maladies cardiaques à l'aide de techniques d'apprentissage automatique et d'apprentissage profond.

L'étude [88] vise à développer un modèle basé sur ML pour la détection des maladies cardiaques. Les algorithmes utilisés sont : KNN, Random Forest, Decision Tree, Support Vector Machine et Naive Bayes. Le KNN a montré son efficacité dans la détection des maladies cardiaques. Les auteurs ont développé un prototype pour valider les résultats. Le prototype consistait en un ensemble de capteurs permettant de surveiller l'état de santé d'une personne. Le modèle proposé atteint un taux d'accuracy de 88,52%.

Dans [89], les auteurs ont proposé un nouveau système avec une forêt aléatoire hybride basée sur un modèle linéaire. Cette étude applique différentes combinaisons de caractéristiques avec de nombreuses approches de classification. La performance de l'approche proposée est améliorée avec un niveau d'accuracy de 88,7%.

Cet article [90] évalue les performances des approches traditionnelles telles que la régression logistique, K-Nearest Neighbours (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), Neural Networks (NN) et le modèle de prédiction proposé de CNN. L'ensemble de données du référentiel d'apprentissage automatique de l'UCI Cleveland est nettoyé, puis divisé en 80% de formation et 20% de test à des fins de formation et de test. Les auteurs de cet article ont proposé un CNN pour prédire avec précision si un patient avait une maladie cardiovasculaire ou non avec un taux d'accuracy de 94%.

Dans [91] ont proposé un système de prédiction des maladies cardiaques basé sur CNN et KNN. Les résultats montrent que les performances de CNN sont meilleures que les performances de KNN avec un accuracy de 84,5% et que le temps nécessaire à la classification pour CNN est inférieur à celui de KNN.

Les auteurs de [92] ont conçu un système de prédiction des maladies cardiaques basé sur plusieurs techniques d'apprentissage automatique telles que Naïve Bayes, K-Nearest Neighbor, Support Vector Machine, Random Forest et Decision Tree. Les résultats ont montré que Naïve Bayes atteignait l'accuracy le plus élevé par rapport aux autres algorithmes avec un accuracy de 82,17%, 84,28% en utilisant respectivement la technique de validation croisée et la distribution des données par split train-test.

2.11.1 Tableau comparatif

La tableau 2.3 compare les travaux connexes et le travail proposé selon différents critères de comparaison.

Travail	**Algorithmes**	**ML et DL**	**Sélection d'attributs**	**Validation**	**Evaluation**	**Acc**
Aanshi Gupta et al. [88] (2019)	KNN, RF, DT, SVM, NB	ML	Non	Nouveau prototype	Acc, Pré, Rap, FM	88,52%
Senthilkumar Mohan et al. [89] (2019)	HRFLM, DT, LM, SVM, RF	ML	Oui	Split	Acc, Pré, FM, Sen, Spé, MC	88,7%
Tulasi Krishna Sajja et al. [90] (2020)	LR, KNN, NB, SVM, NN, CNN	ML, DL	Non	Split	Acc, ROC	94%
Dhiraj Dahiwade et al. [91] (2019)	KNN, CNN	ML, DL	Non	Split	Acc, Pré, Rap, FM	84,5%
Halima El Hamdaoui et al. [92] (2020)	NB, KNN, SVM, RF, DT	ML	Non	Cross, Split	Acc, Pré, Rap, FM	82,17%, 84,28%
Travail proposé	SVM, LR, RF, DT, GNB, KNN, XGBoost, Dense-DNN	ML, DL	Oui	Cross, Split	Acc, Pré, Rap, FM, MAE, RMSE, MC, ROC, AUC	85,4%, 95%

TABLE 2.3 – Tableau comparatif

HRFLM : Hybride Random Forest and Linear Method

LM : Language Model

Sen : Sensibilité

Spé : Spécificité

MC : Matrice de Confusion

2.12 Conclusion

Dans ce deuxième chapitre, nous avons d'abord décrit les différents types d'apprentissage automatique, l'apprentissage profond, les réseaux de neurones, les modèles neuronaux de base et l'apprentissage des réseaux de neurones puis nous avons donné un aperçu sur le processus de fouille de données et ses différentes méthodes et la sélection d'attributs avec des exemples

dans le domaine biomédical, nous avons ensuite décrit la classification supervisée, différents algorithmes de classification et les différentes techniques d'évaluation et enfin, nous avons présenté les datasets d'entraînement par apprentissage profond utilisés pour la prédiction des maladies cardiovasculaires, cité les travaux connexes et réalisé une comparaison des travaux connexes et du travail proposé selon différents critères.

Dans le troisième chapitre nous développerons une nouvelle approche de classification basée sur l'apprentissage profond, expérimentée et évaluée sur une base de données pour la prédiction des maladies cardiovasculaires.

Chapitre 3

Méthodologie proposée

3.1 Introduction

Les maladies cardiovasculaires sont des maladies qui mettent la vie en danger. Dans le monde, la plupart des gens sont touchés par les maladies cardiovasculaires. Le diagnostic des maladies cardiovasculaires est également l'une des tâches les plus importantes et les plus difficiles pour les praticiens. Ainsi, un modèle de classification efficace est nécessaire pour prédire l'état cardiaque des patients, ces informations aideront les praticiens à prendre des décisions précises et efficaces dès le début de la maladie (stade précoce).

En vue de cela, le prétraitement des données est une étape majeure et essentielle dont l'objectif principal est de préparer les données pour les algorithmes, utiliser un algorithme génétique qui permet d'optimiser la recherche d'un sous-ensemble d'attributs pertinents et les techniques de classification pour améliorer les performances du modèle de prédiction des maladies cardiovasculaires.

3.2 Processus de construction du modèle prédictif

Le processus de construction du modèle prédictif est présenté dans la figure 3.1 et ses différentes étapes se résument comme suit :

A) L'analyse exploratoire des données est réalisée comme suit :

   (a) Les données des patients collectées de la base de données de Cleveland sont visualisées.

B) Le prétraitement des données est réalisé comme suit :

   (b) Les données manquantes et les doublons sont filtrés.

   (c) L'ensemble de données nettoyées est normalisé.

   (d) Un nouvel ensemble de données traitées résulte des étapes précédentes.

(e) Une matrice de corrélation est calculée et utilisée pour évaluer la dépendance entre plusieurs variables en même temps.

(f) La sélection d'attributs à l'aide d'un algorithme génétique pour préserver les attributs les plus performants.

(g) Les attributs sélectionnés sont extraits de l'ensemble de données traitées pour former un nouvel ensemble de données traitées et réduites.

C) La classification et la validation des données est réalisée comme suit :

(h) Sept algorithmes des modèles Machine Learning de classification (SVM, LR, RF, DT, GNB, KNN et XGBoost) sont implémentés avec les deux méthodes de validation (split-validation et cross-validation).

(i) L'algorithme de classification Deep Learning du modèle proposé (Dense-DNN) est implémenté avec les deux méthodes de validation.

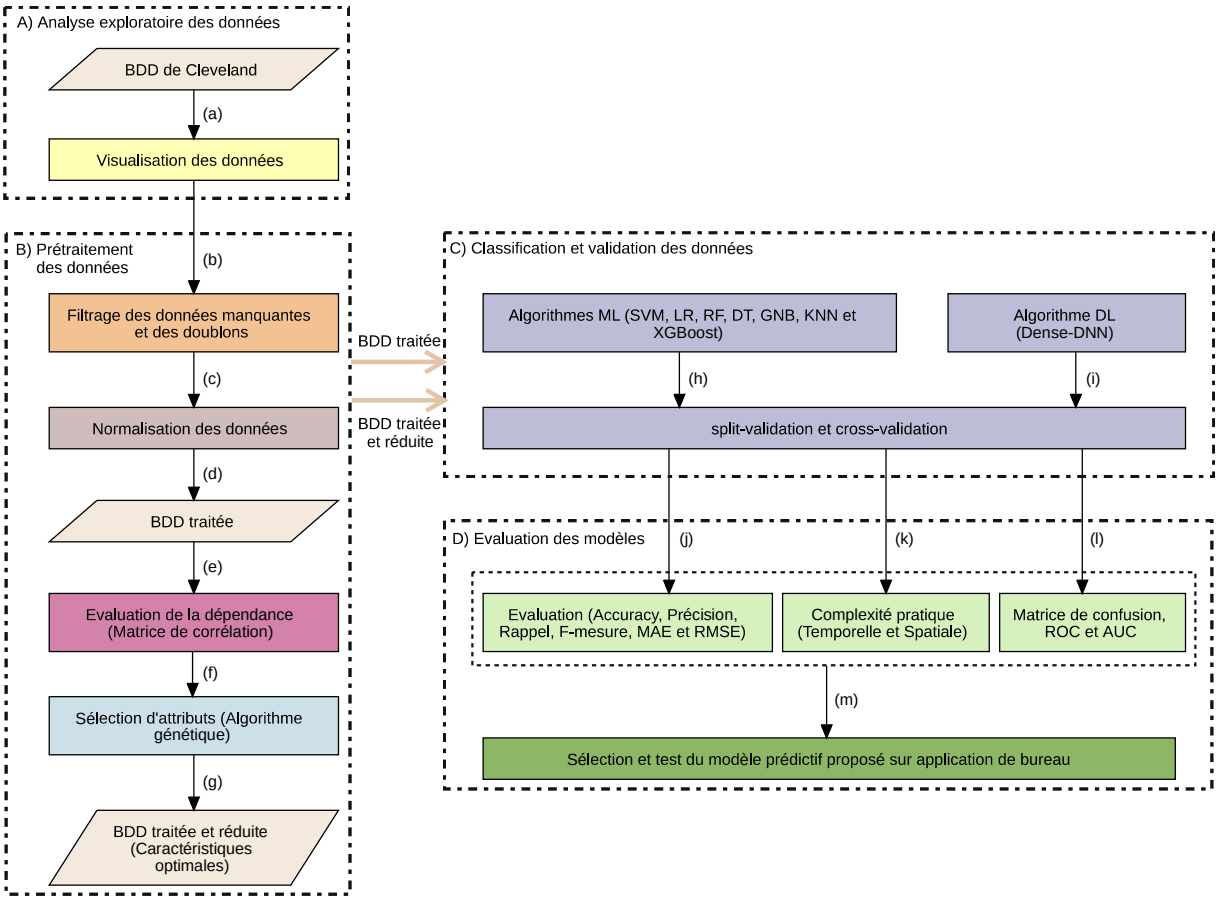


FIGURE 3.1 – Processus de construction du modèle prédictif

D) L'évaluation des modèles est réalisée comme suit :

(j) Les modèles ML et le modèle proposé sont évalués avec les deux méthodes de validation en fonction de l'accuracy, de la précision, du rappel, de la mesure F, de l'erreur absolue moyenne (MAE) et de la racine de l'erreur quadratique moyenne (RMSE) et comparés en utilisant l'ensemble de données traitées et l'ensemble de données traitées et réduites.

(k) Le modèle proposé est évalué avec la complexité pratique (Temporelle et Spatiale) avec les deux méthodes de validation et comparé en utilisant l'ensemble de données traitées et l'ensemble de données traitées et réduites.

(l) Le modèle proposé est évalué à l'aide de la matrice de confusion, de la courbe ROC et de l'aire sous la courbe ROC (AUC) avec la méthode de validation « split-validation » en utilisant l'ensemble de données traitées et réduites.

(m) Le modèle prédictif proposé est finalement sélectionné et testé sur l'application de bureau réalisée pour les besoins de ce test.

3.3 Analyse exploratoire des données

Nous avons utilisé pour ce travaille la base de données de Cleveland [1] qui est la seule qui a été utilisée par les chercheurs en ML à cette date. Cette base de données contient 76 attributs (variables) et 303 instances (patients), mais toutes les expériences publiées se réfèrent à l'utilisation d'un sous-ensemble de 14 attributs d'entre eux (Figure 3.2).

```
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

FIGURE 3.2 – Informations sur la base de données utilisée

1. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation : Robert Detrano, M.D., Ph.D.

3.3.1 Variables (attributs) utilisées

Le tableau 3.1 décrit les 14 attributs utilisés et précise leurs types. Les noms et numéros de sécurité sociale des patients ont été récemment retirés de la base de données, remplacés par des valeurs fictives.

	Variable	**Description**	**Type**
1	age	Âge du patient (en années(29-79))	Numérique
2	sex	Sexe du patient (1 = mâle, 0 = femelle)	Catégorique
3	cp	Type de douleur thoracique (0 : Angine de poitrine ou angor typique (douleur thoracique liée à la diminution de l'apport sanguin au cœur), 1 : Angor atypique (douleur thoracique non liée au cœur), 2 : Douleur non angineuse (spasmes typiquement œsophagiens (non liés au cœur)), 3 : Asymptomatique (douleur thoracique ne montrant aucun signe de maladie))	Catégorique
4	trestbps	Pression artérielle au repos (en mmHg à l'admission à l'hôpital (94-200))	Numérique
5	chol	Cholestérol sérique (en mg/dl (126-564))	Numérique
6	fbs	Glycémie à jeun > 120 mg/dl (1 = vrai, 0 = faux)	Catégorique
7	restecg	Résultats électrocardiographiques au repos (0 : normal, 1 : présentant une anomalie de l'onde ST-T (inversions de l'onde T et/ou élévation ou dépression du segment ST > 0.05 mV), 2 : montrant une hypertrophie ventriculaire gauche probable ou certaine selon les critères d'Estes)	Catégorique
8	thalach	Fréquence cardiaque maximale atteinte (en battements par minute (71-202))	Numérique
9	exang	Angine de poitrine induite par l'exercice (1 = oui, 0 = non)	Catégorique
10	oldpeak	Dépression du segment ST induite par l'exercice par rapport au repos (0-6.2)	Numérique
11	slope	La pente du segment ST d'exercice maximal (0 : ascendante, 1 : plate, 2 : descendante)	Catégorique
12	ca	Nombre de vaisseaux principaux (0-3) colorés par fluoroscopie	Numérique
13	thal	Un trouble sanguin appelé thalassémie (1 : normal, 2 : défaut corrigé, 3 : défaut réversible)	Catégorique
14	target	Diagnostic de maladie cardiaque ou statut de la maladie angiographique (0 : absence de maladie cardiovasculaire ou rétrécissement du diamètre < 45%, 1 : présence de maladie cardiovasculaire ou rétrécissement du diamètre > 45%)	Catégorique

TABLE 3.1 – Variables (attributs) utilisées

Statut de la maladie angiographique : Une angiographie coronarienne (aussi appelée coronarographie) est un test qui consiste à prendre des radiographies des artères coronariennes et des vaisseaux qui alimentent le cœur.

3.3.2 Visualisation des variables

Nous représentons graphiquement les variables par rapport au nombre de patients de la base de données.

Variables catégoriques

target : Diagnostic de maladie cardiaque ou statut de la maladie angiographique (0 : absence de maladie cardiovasculaire ou rétrécissement du diamètre $< 45\%$, 1 : présence de maladie cardiovasculaire ou rétrécissement du diamètre $> 45\%$).

sex : Sexe du patient (0 : femelle, 1 : mâle).

cp : Type de douleur thoracique (0 : Angine de poitrine ou angor typique, 1 : Angor atypique, 2 : Douleur non angineuse, 3 : Asymptomatique).

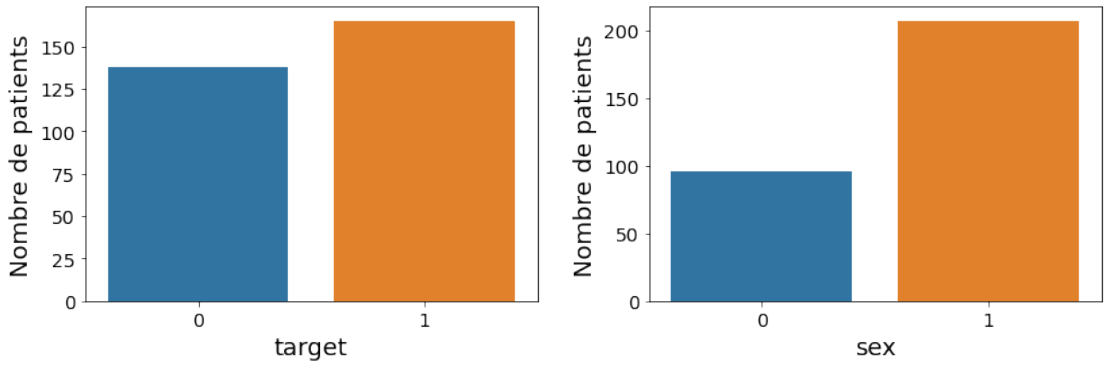
fbs : Glycémie à jeun > 120 mg/dl (0 : faux, 1 : vrai).

restecg : Résultats électrocardiographiques au repos (0 : normal, 1 : présentant une anomalie de l'onde ST-T, 2 : montrant une hypertrophie ventriculaire gauche probable ou certaine).

exang : Angine de poitrine induite par l'exercice (0 : non, 1 : oui).

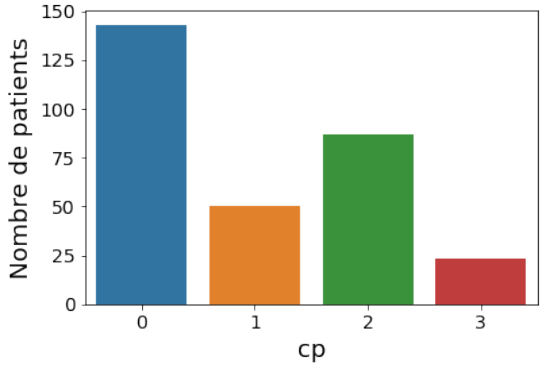
slope : La pente du segment ST d'exercice maximal (0 : ascendante, 1 : plate, 2 : descendante).

thal : Un trouble sanguin appelé thalassémie (1 : normal, 2 : défaut corrigé, 3 : défaut réversible)[0, 1, 2, 3].

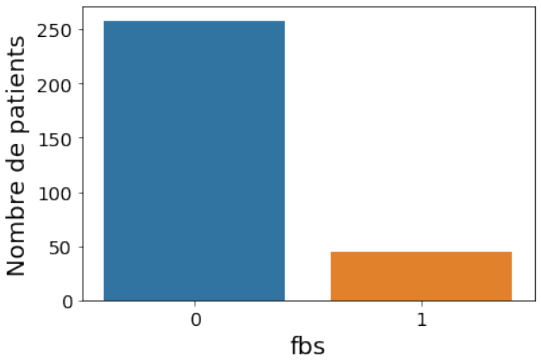


(a) target

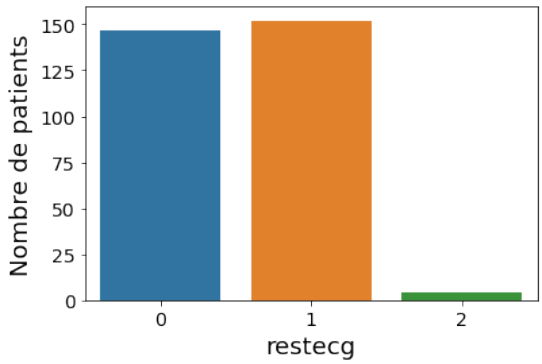
(b) sex



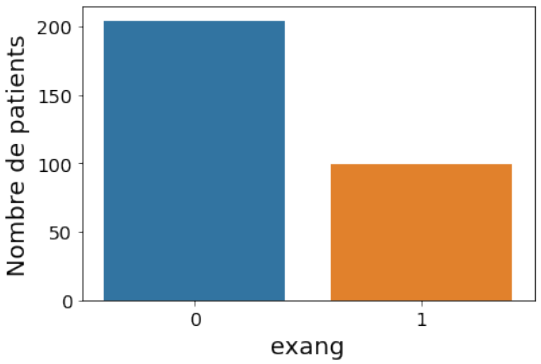
(c) cp



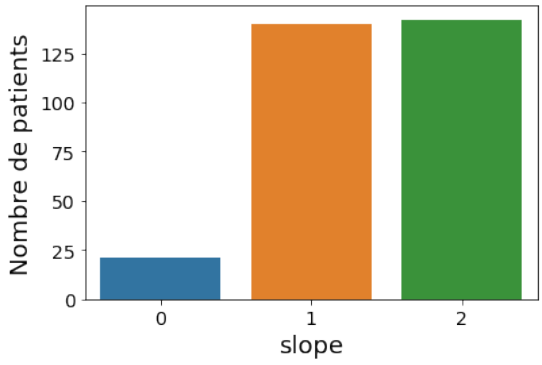
(d) fbs



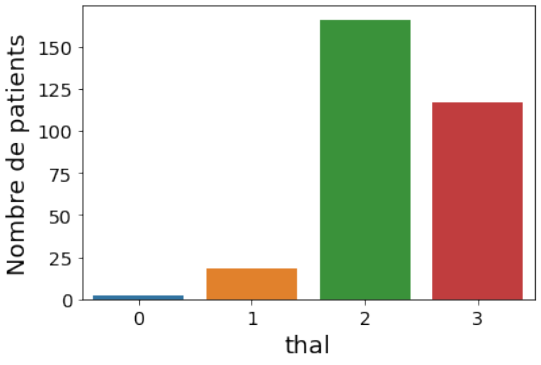
(e) restecg



(f) exang



(g) slope



(h) thal

FIGURE 3.3 – Visualisation des variables catégoriques

Remarques sur les variables catégoriques

Variable	Remarques
target	Le nombre de patients non atteints de maladie cardiovasculaire est 138 avec une proportion de 45,54% et le nombre de patients atteints de maladie cardiovasculaire est 165 avec une proportion de 54,46%. On remarque que les deux classes cibles sont presque égales, les données sont équilibrées.
sex	Le nombre de patients de sexe féminin est 96 avec une proportion de 31,68% et le nombre de patients de sexe masculin est 207 avec une proportion de 68,32%. On remarque un nombre de patients de sexe masculin plus élevé que le nombre de patients de sexe féminin.
cp	Le nombre de patients atteints d'une douleur thoracique de type 0 est 143 avec une proportion de 47,19%, de type 1 est 50 avec une proportion de 16,5%, de type 2 est 87 avec une proportion de 28,71% et de type 3 est 23 avec une proportion de 7,59%. On remarque que presque la moitié des patients sont atteints d'une douleur thoracique de type 0 (Angine de poitrine ou angor typique (douleur thoracique liée à la diminution de l'apport sanguin au coeur)). On remarque que presque la moitié des patients sont atteints d'une douleur thoracique de type 0.
fbs	Le nombre de patients avec un taux de Glycémie à jeun $>$ 120 mg/dl est 45 avec une proportion de 14,85% et le nombre de patients avec un taux de Glycémie à jeun $<$ 120 mg/dl est 258 avec une proportion de 85,15%. On remarque que la majorité des patients (85%) ont un taux de Glycémie à jeun $<$ 120 mg/dl.
restecg	Le nombre de patients avec des résultats électrocardiographiques au repos de type 0 est 147 avec une proportion de 48,51%, de type 1 est 152 avec une proportion de 50,17% et de type 2 est 4 avec une proportion de 1,32%. On remarque que presque la moitié des patients ont des résultats normaux et l'autre moitié présentant une anomalie de l'onde ST-T (inversions de l'onde T et/ou élévation ou dépression du segment ST $>$ 0,05 mV).
exang	Le nombre de patients avec une angine de poitrine induite par l'exercice est 99 ce qui représente presque un tier des patients(32,67%).
slope	La pente du segment ST d'exercice maximal est ascendante pour 21 patients avec une proportion de 6,93%, plate pour 140 patients avec une proportion de 46,2 et descendante pour 142 patients avec une proportion de 46,86%. On remarque que pour la majorité des patients, la pente est soit plate soit descendante à probabilité presque égale.
thal	Le nombre de patients représentant un état normal par rapport au trouble sanguin appelé thalassémie est 18 avec une proportion de 5,94%, le nombre de patients représentant un défaut corrigé de thalassémie est 166 avec une proportion de 54,79% et le nombre de patients représentant un défaut réversible de thalassémie est 117 avec une proportion de 38,61%. On remarque que la majorité des patients représentent un défaut de thalassémie, plus que la moitié représentent un défaut corrigé et plus qu'un tier représentent un défaut réversible.

TABLE 3.2 – Remarques sur les variables catégoriques

Variables numériques

age : Âge du patient(en années)[29-79].

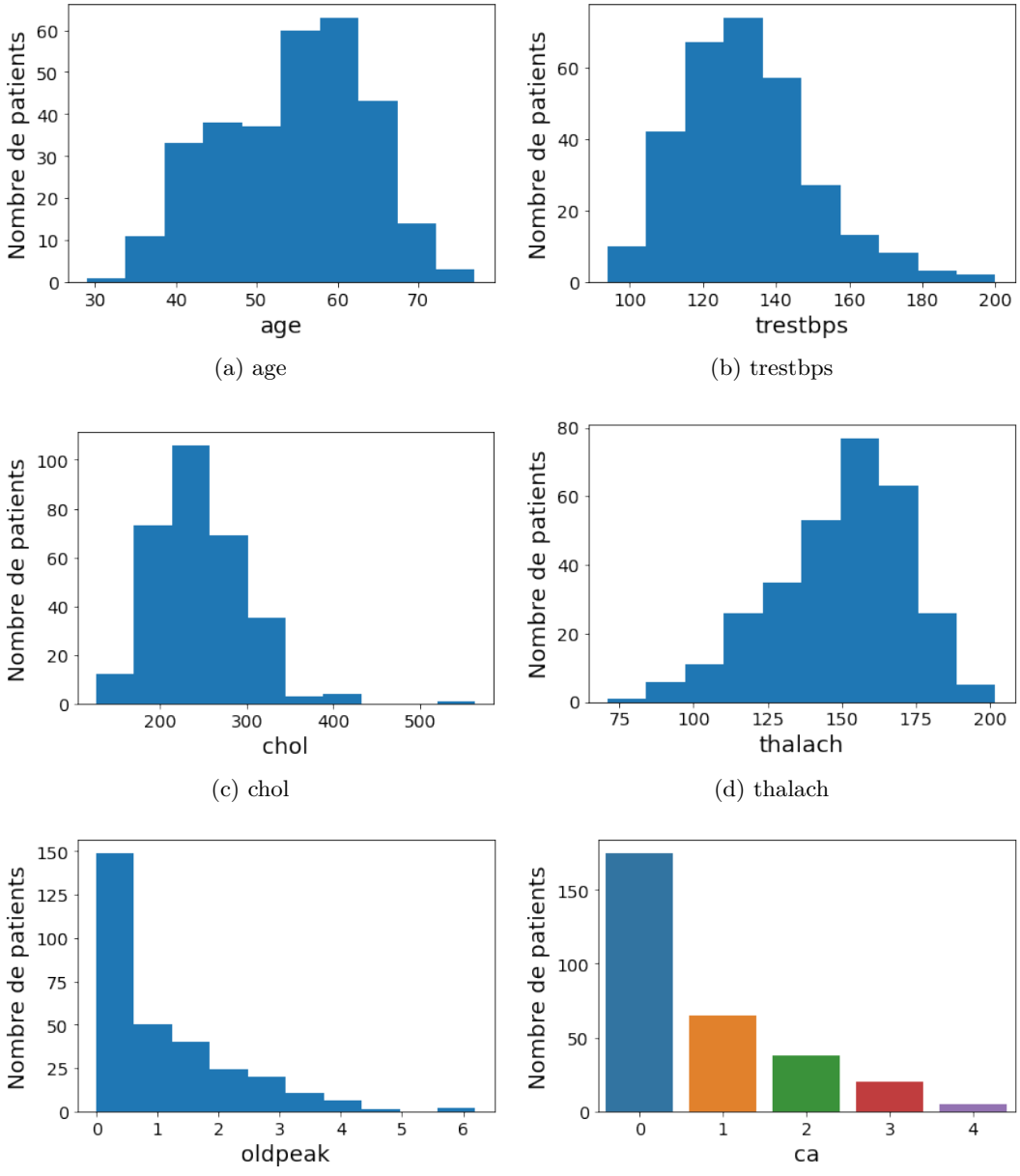
trestbps : Pression artérielle au repos (en mmHg à l'admission à l'hôpital)[94-200].

chol : Cholestérol sérique(en mg/dl)[126-564].

thalach : Fréquence cardiaque maximale atteinte[71-202].

oldpeak : Dépression du segment ST induite par l'exercice par rapport au repos[0-6.2].

ca : Nombre de vaisseaux principaux (0-3)[0, 1, 2, 3].



(a) age (b) trestbps

(c) chol (d) thalach

(e) oldpeak (f) ca

FIGURE 3.4 – Visualisation des variables numériques

Remarques sur les variables numériques

Variable	Remarques
age	Les tranches d'âge les plus représentées sont la tranche des 53-57,8 avec 60 patients et la tranche des 57,8-62,6 avec 63 patients et les tranches d'âge les moins représentées sont la tranche des 29-33,8 avec 1 patient et la tranche des 72,2-77 avec 3 patients.
trestbps	La pression artérielle au repos à l'admission à l'hôpital varie entre 104,6 mmHg et 157,6 mmHg pour la majorité des patients (267) avec un pique de 74 patients pour la tranche 125,8-136,4 mmHg.
chol	Le taux de cholestérol sérique varie entre 169,8 mg/dl et 345 mg/dl pour la majorité des patients (283) avec un pique de 106 patients pour la tranche 213,6-257,4 mg/dl.
thalach	La fréquence cardiaque maximale atteinte varie entre 110,3 battements par minute et 188,9 battements par minute pour la majorité des patients (280) avec un pique de 77 patients pour la tranche 149,6-162,7 battements par minute.
oldpeak	La dépression du segment ST induite par l'exercice par rapport au repos est entre 0 et 0,62 pour presque la moitié des patients (149) et ne dépasse pas 3,1 pour la majorité des patients (283).
ca	Le nombre de vaisseaux principaux colorés par fluoroscopie est 0 pour 175 patients avec une proportion de 57,76%, 1 pour 65 patients avec une proportion de 21,45%, 2 pour 38 patients avec une proportion de 12,54% et 3 pour 20 patients avec une proportion de 6,6%. On remarque que la majorité (plus que la moitié) des patients ont 0 vaisseaux principaux colorés par fluoroscopie.

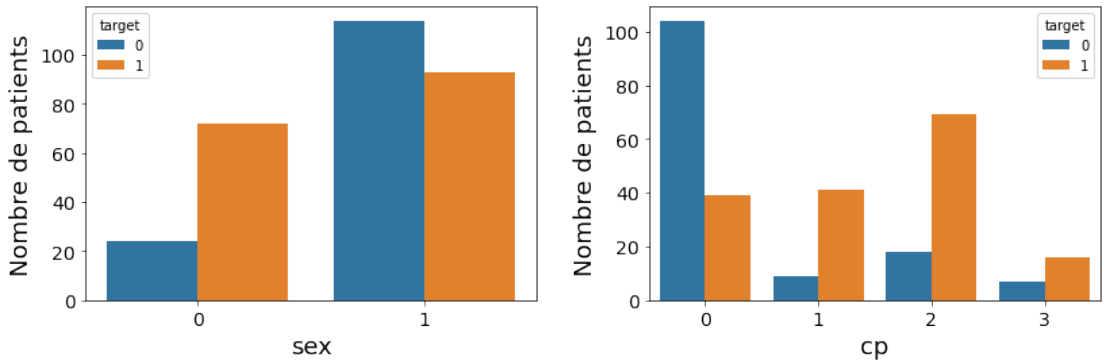
TABLE 3.3 – Remarques sur les variables numériques

3.3.3 Visualisation des relations entre les variables et la classe cible

Visualisation des relations entre les variables catégoriques et la classe cible

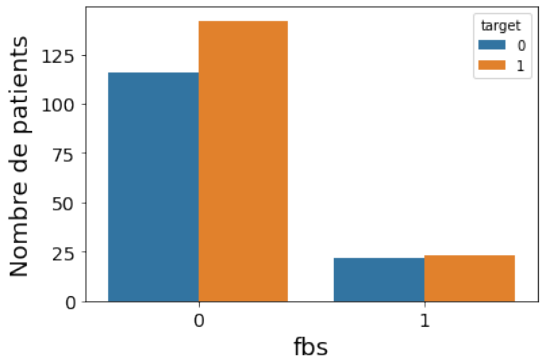
Nous représentons graphiquement les relations entre les variables catégoriques et la classe cible (target).

target : Diagnostic de maladie cardiaque ou statut de la maladie angiographique (0 : absence de maladie cardiovasculaire ou rétrécissement du diamètre $< 45\%$, 1 : présence de maladie cardiovasculaire ou rétrécissement du diamètre $> 45\%$).

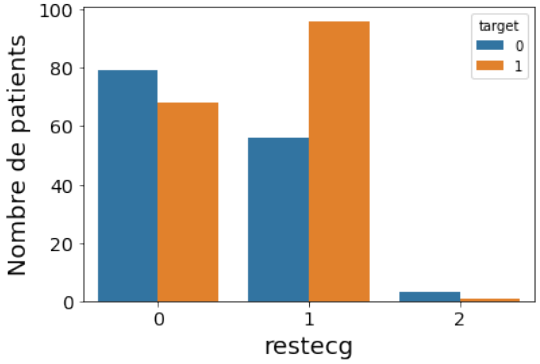


(a) sex-cible

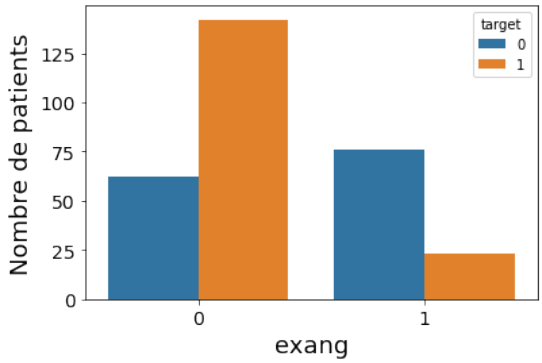
(b) cp-cible



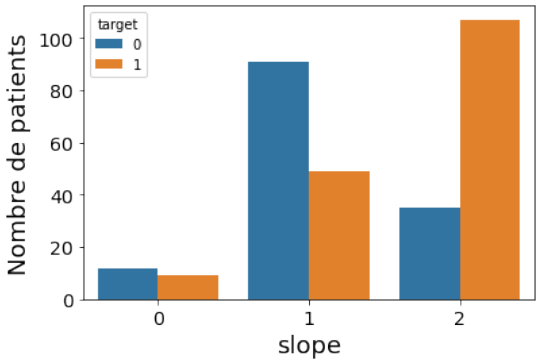
(c) fbs-cible



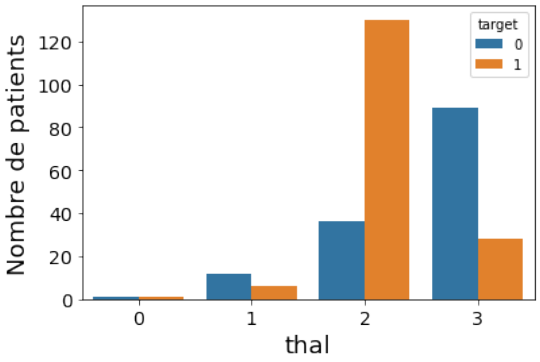
(d) restecg-cible



(e) exang-cible



(f) slope-cible



(g) thal-cible

FIGURE 3.5 – Visualisation des relations entre les variables catégoriques et la classe cible

Remarques sur les relations entre les variables catégoriques et la classe cible

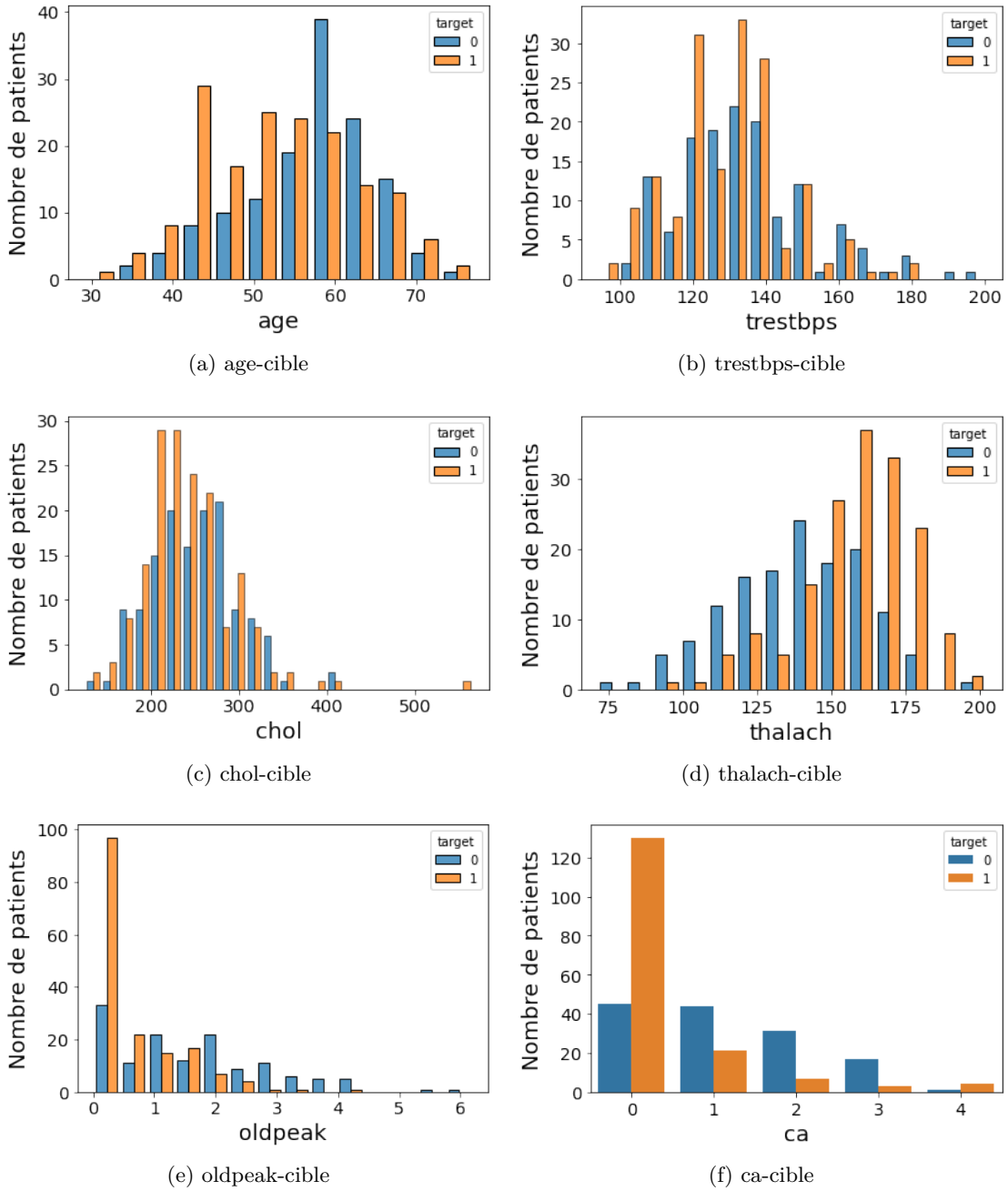
Relation	**Remarques**
sex-cible	La proportion des femmes atteintes de maladie cardiovasculaire est beaucoup plus élevée (75%) que celle des femmes non atteintes de maladie cardiovasculaire et la proportion des hommes atteints de maladie cardiovasculaire est moins élevée (environ 45%) que celle des hommes non atteints de maladie cardiovasculaire.
cp-cible	La probabilité de maladie cardiovasculaire chez les patients souffrant d'une Angor typique est relativement faible (environ 0,27/1), tandis que les patients souffrant d'une douleur thoracique de type 1,2 ou 3 la probabilité est relativement élevée (respectivement environ 0,82/1, 0,79/1 ou 0,7/1), ce qui indique une relation entre la maladie cardiovasculaire et le type des douleurs thoraciques.
fbs-cible	La probabilité de maladie cardiovasculaire est d'environ 0,51/1 chez les patients avec un taux de Glycémie à jeun $>$ 120 mg/dl et relativement un peu plus élevée (environ 0,55/1) chez les patients avec un taux de Glycémie à jeun $<$ 120 mg/dl.
restecg-cible	La proportion des patients présentant une anomalie de l'onde ST-T atteints de maladie cardiovasculaire est relativement beaucoup plus élevée (environ 63%) que celle des patients non atteints de maladie cardiovasculaire et la proportion des patients avec des résultats électrocardiographiques au repos normaux ou montrant une hypertrophie ventriculaire gauche probable ou certaine atteints de maladie cardiovasculaire est relativement moins élevée (respectivement environ 46% ou 25%) que celle des patients non atteints de maladie cardiovasculaire.
exang-cible	La probabilité de maladie cardiovasculaire est relativement faible (environ 0,23/1) chez les patients atteints d'une Angine de poitrine induite par l'exercice, tandis que chez les patients qui ne présentent pas une Angine de poitrine induite par l'exercice la probabilité est relativement élevée (environ 0,7/1), ce qui indique une relation entre la maladie cardiovasculaire et la présence ou non d'une Angine de poitrine induite par l'exercice.
slope-cible	La proportion des patients présentant une pente du segment ST d'exercice maximal descendante atteints de maladie cardiovasculaire est relativement beaucoup plus élevée (environ 75%) que celle des patients non atteints de maladie cardiovasculaire et la proportion des patients présentant une pente du segment ST d'exercice maximal ascendante ou plate atteints de maladie cardiovasculaire est relativement moins élevée (respectivement environ 43% ou 35%).
thal-cible	La proportion de patients atteints de maladie cardiovasculaire est relativement très élevée (environ 78%) chez les patients pour lesquels le résultat du test sur le trouble sanguin appelé thalassémie détecte un défaut corrigé, tandis que chez les patients pour lesquels le résultat du test est normal ou détecte un défaut réversible la proportion de patients atteints de maladie cardiovasculaire est relativement faible (respectivement environ 33% ou 24%).

TABLE 3.4 – Remarques sur les relations entre les variables catégoriques et la classe cible

Visualisation des relations entre les variables numériques et la classe cible

Nous représentons graphiquement les relations entre les variables numériques et la classe cible (target).

target : Diagnostic de maladie cardiaque ou statut de la maladie angiographique (0 : absence de maladie cardio-vasculaire ou rétrécissement du diamètre $< 45\%$, 1 : présence de maladie cardiovasculaire ou rétrécissement du diamètre $> 45\%$).



(a) age-cible

(b) trestbps-cible

(c) chol-cible

(d) thalach-cible

(e) oldpeak-cible

(f) ca-cible

FIGURE 3.6 – Visualisation des relations entre les variables numériques et la classe cible

Remarques sur les relations entre les variables numériques et la classe cible

Relation	Remarques
age-cible	La proportion des patients atteints de maladie cardiovasculaire est relativement moins élevée (environ 38%) pour la tranche d'âge 57,8-67,4 que celle des patients non atteints de maladie cardiovasculaire, relativement plus élevée que celle des patients non atteints de maladie cardiovasculaire pour les tranches d'âge supérieures et relativement beaucoup plus élevée que celle des patients non atteints de maladie cardiovasculaire pour les tranches d'âge inférieures.
trestbps-cible	La proportion des patients atteints de maladie cardiovasculaire est relativement plus élevée que celle des patients non atteints de maladie cardiovasculaire chez les patients avec une pression artérielle au repos à l'admission à l'hôpital inférieure à 157,6 mmHg et relativement moins élevée que celle des patients non atteints de maladie cardiovasculaire chez les patients avec une pression artérielle au repos à l'admission à l'hôpital supérieure à 157,6 mmHg.
chol-cible	La proportion des patients atteints de maladie cardiovasculaire est relativement plus élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec un taux de Cholestérol sérique entre 169,8 et 257,4 mg/dl, relativement moins élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec un taux entre 257,4 et 301,2 mg/dl et relativement plus élevée ou égale à celle des patients non atteints de maladie cardiovasculaire pour les taux supérieures à 301,2 mg/dl ou inférieures à 169,8 mg/dl.
thalach-cible	La proportion des patients atteints de maladie cardiovasculaire est relativement plus élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec une fréquence cardiaque maximale atteinte supérieure à 149,6 battements par minute et relativement moins élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec une fréquence cardiaque maximale atteinte inférieure à 149,6 battements par minute.
oldpeak-cible	La proportion des patients atteints de maladie cardiovasculaire est relativement plus élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec une dépression du segment ST induite par l'exercice par rapport au repos entre 0 et 0,62 ou entre 1,24 et 1,86 et relativement moins élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec une dépression du segment ST induite par l'exercice par rapport au repos hors de ces intervalles.
ca-cible	La proportion des patients atteints de maladie cardiovasculaire est relativement beaucoup plus élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec un nombre de vaisseaux principaux colorés par fluoroscopie égale à 0 et relativement moins élevée que celle des patients non atteints de maladie cardiovasculaire pour les patients avec un nombre de vaisseaux principaux colorés par fluoroscopie égale à 1, 2 ou 3.

TABLE 3.5 – Remarques sur les relations entre les variables numériques et la classe cible

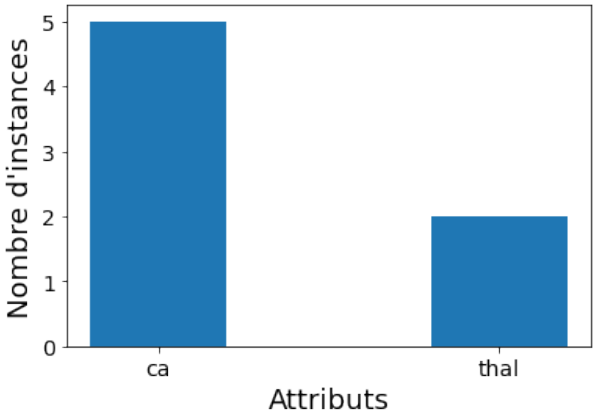
3.4 Prétraitement des données

3.4.1 Filtrage des données manquantes

D'après les graphiques de visualisation des données vus précédemment, les attributs « ca » et « thal » ont les valeurs invalides (bruit) 4 et 0 respectivement.

Visualisation des données manquantes

Les attributs et les instances avec des données manquantes sont indiqués dans la figure 3.7.



(a) Nombre d'instances par attribut

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
92	52	1	2	138	223	0	1	169	0	0.0	2	4	2	1
158	58	1	1	125	220	0	1	144	0	0.4	1	4	3	1
163	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1
164	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1
251	43	1	0	132	247	1	0	143	1	0.1	1	4	3	0

(b) Instances avec ca = 4

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
48	53	0	2	128	216	0	0	115	0	0.0	2	0	0	1
281	52	1	0	128	204	1	1	156	1	1.0	1	0	0	0

(c) Instances avec thal = 0

FIGURE 3.7 – Visualisation des données manquantes

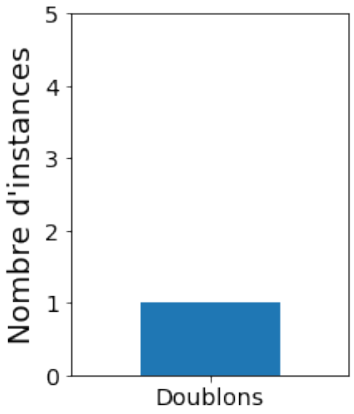
Suppression des données manquantes

Après la suppression des données manquantes, notre base de données diminue de 7 instances (Figure 3.7) et contient 296 instances.

3.4.2 Filtrage des doublons

Visualisation des doublons

Les doublons sont indiqués dans la figure 3.8.



(a) Nombre de doublons

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
164	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1

(b) Doublons

FIGURE 3.8 – Visualisation des doublons

Suppression des doublons

Nous constatons dans les figures 3.7b et 3.8b que le doublon (instance 164) fait partie des instances avec des données manquantes supprimées dans l'étape de filtrage des données manquantes. Notre base de données finale contient 296 instances.

Nous utiliserons dorénavant la base de données résultante après la suppression des données manquantes et des doublons.

3.4.3 Normalisation des données

La méthode de normalisation Min-Max est utilisée pour normaliser les données. Cette méthode met à l'échelle la plage de données à [0,1]. Dans la plupart des cas, la normalisation est également utilisée sur la base des caractéristiques.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

3.4.4 Evaluation de la dépendance (Matrice de corrélation)

Une matrice de corrélation est utilisée pour évaluer la dépendance entre plusieurs variables en même temps. Le résultat est une table contenant les coefficients de corrélation entre chaque variable et les autres.

Il existe différentes méthodes de tests de corrélation : Le test de corrélation de Pearson, la corrélation de Kendall et celle de Spearman qui sont des tests basés sur le rang.

Dans le graphique ci-dessous, nous avons une carte thermique de corrélation entre les variables, y compris notre cible. Les valeurs proches de 0 indiquent qu'il n'y a pas de corrélation, les valeurs négatives montrent une corrélation avec une tendance négative (plus elle est proche de -1) et positive, vice versa.

Pour calculer le rapport de corrélation entre deux variables différentes, nous avons utilisé le coefficient de corrélation de Pearson défini par la formule suivante :

$$\frac{\sum (x_i - \bar{x}) \times (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2}}$$

où x_i désigne la première variable, y_i la deuxième variable, $\bar{x}$ la moyenne de l'échantillon pour la première variable et $\bar{y}$ la moyenne de l'échantillon pour la deuxième variable.

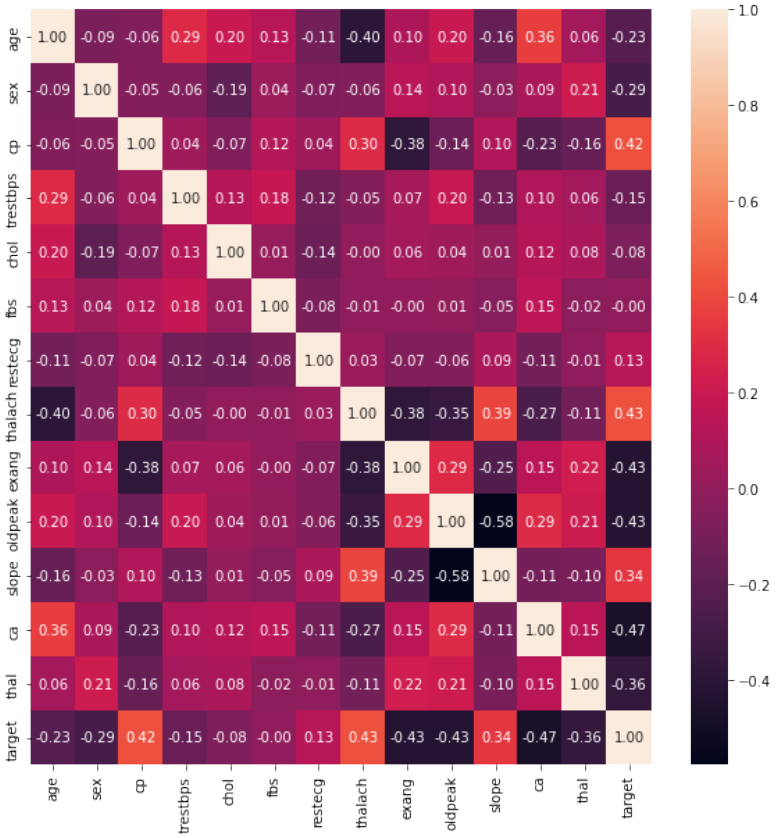


FIGURE 3.9 – Matrice de corrélation

Dans la carte de corrélation (Figure 3.9), nous voyons que :

— Les caractéristiques « chol » et « fbs » montrent une corrélation négative avec la cible proche de 0 (sans corrélation).

— « age », « sex », « trestbps » et « restecg » montrent une faible corrélation avec la cible.

— « cp », « thalach » et « slope » montrent une bonne corrélation positive avec la cible.

— « exang », « oldpeak », « ca » et « thal » ont une bonne corrélation négative avec la cible.

— La corrélation entre deux caractéristiques ne dépasse jamais 0.6, ce qui est acceptable. Une corrélation proche de 1 signifie que deux attributs sont redondants (entre eux).

3.4.5 Sélection d'attributs (Feature Selection) à l'aide d'un algorithme génétique

La sélection d'attributs est l'une des méthodes qui améliorent et optimisent les performances d'un algorithme d'apprentissage automatique. Dans la sélection des attributs, nous trouvons le sous-ensemble optimal d'attributs qui contribue le plus à notre variable prédite (cible).

La capacité de calcul des modèles d'apprentissage automatique dépend beaucoup de l'ensemble d'attributs. Retenir les attributs significatifs améliore considérablement le temps d'apprentissage, ainsi que l'accuracy.

Pourquoi avons-nous besoin de la sélection d'attributs ?

1. Améliorer la généralisation des modèles en réduisant le nombre des données.
2. Supprimez les données inutiles/redondantes.
3. Endiguer la malédiction de la dimensionnalité.
4. Optimiser le temps d'entraînement.

Algorithme génétique

Il existe de nombreux indicateurs d'examen physique pour les patients. S'ils sont tous introduits dans le modèle pour l'apprentissage, cela augmentera la charge de calcul du modèle. Dans la plupart des cas, ces indicateurs d'examen physique auront une grande corrélation et réduiront l'accuracy des résultats. Compte tenu de la charge de calcul et de l'accuracy, la réduction de la dimensionnalité des caractéristiques - à l'aide d'un algorithme génétique - est nécessaire pour préserver les caractéristiques les plus performantes. En utilisant cette méthode, le nombre optimal de caractéristiques dans différentes conditions peut être obtenu.

L'algorithme génétique est un algorithme de recherche et d'optimisation basé sur le principe de l'évolution naturelle. L'algorithme tente d'imiter le concept d'évolution humaine en modi-

fiant un ensemble d'individus appelé population, suivi d'une sélection aléatoire de parents dans cette population pour effectuer la reproduction sous forme de mutation et de croisement. Ce processus se poursuit jusqu'à ce que le critère d'arrêt soit atteint. Au final, il donne le meilleur individu/solution.

Cet algorithme est basé sur le fait que les "bons" parents produisent une "bonne" progéniture, ce qui fait converger l'algorithme vers une valeur optimale. Étant donné qu'il s'agit d'une méthode sans dérivation et que l'approche est purement aléatoire, on peut s'attendre à ce qu'elle converge vers un optimum global au fil du temps.

Etapes de l'algorithme génétique

Les étapes de l'algorithme génétique sont (Figure 3.10) :

1. Initialisation de la population
2. Calcul de la valeur fitness de chaque individu de la population
3. Sélection des parents
4. Croisement
5. Mutation

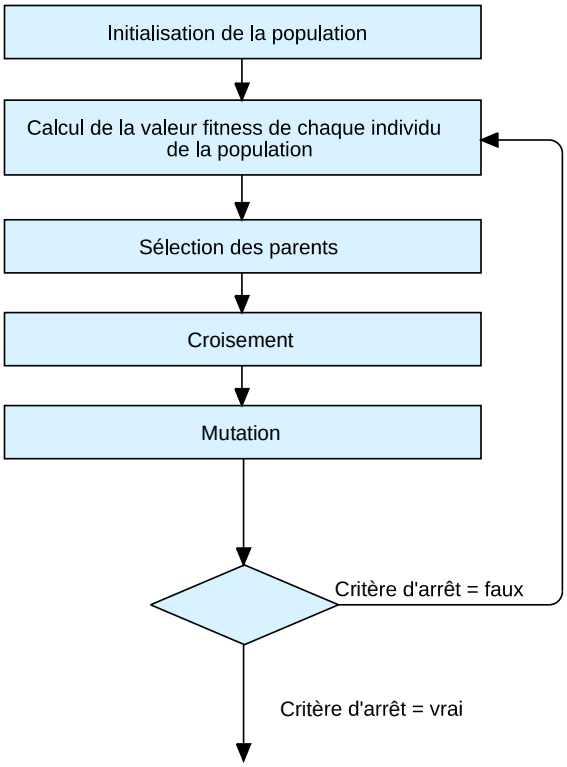


FIGURE 3.10 – Étapes de l'algorithme génétique

L'implémentation de l'algorithme génétique de sélection d'attributs est décrite dans le tableau 3.6 :

```
1.  Algorithme 1. Sélection d'attributs-Algorithme génétique
2.    # Récupérer et normaliser les données
3.    Fonction recuperation(donnees,attributs)
4.    Fonction normalisation(donneesDesAttributs)
5.    # Algorithme génétique
6.    Entrée
7.     donneesDesAttributs,donneesDeLaCible,tailleDeLaPopulation,
       critereDArret,nombreDAttributsASelectionner
8.    Sortie
9.     meilleursAttributs,meilleurF1Score
10.   Fonction algorithme_genetique(donneesDesAttributs,
      donneesDesAttributs,donneesDeLaCible,tailleDeLaPopulation,
      critereDArret,nombreDAttributsASelectionner)
11.    Début
12.     # Fonctions des différentes étapes de l'algorithme géné-
      tique
13.     Fonction initialiser_population(tailleDeLaPopulation,c,
        nombreDAttributsASelectionner)
14.     Fonction calculer_fitness(donneesDesAttributs,donneesDeLaCible)
15.     Fonction recuperer_fitness(population,donneesDesAttributs)
16.     Fonction selectionner_parents(population,valeursFitness)
17.     Fonction croisement_a_deux_points(parents,probabilite)
18.     Fonction mutation(croisementPopulation)
19.     c=donneesDesAttributs.shape[1] # Longueur du chromo-
      some
20.     population=initialiser_population(tailleDeLaPopulation,c,
        nombreDAttributsASelectionner)
21.     valeursFitness=recuperer_fitness(population,donneesDesAttributs)
22.     parents=selectionner_parents(population,valeursFitness)
23.     croisementPopulation=croisement_a_deux_points(parents,0.78)
24.     population=croisementPopulation
25.     p=random.uniform(0,1) # Probabilité de mutation
26.     Si(p<=0.001) :
27.      muteePopulation=mutation(croisementPopulation)
28.      population=muteePopulation
29.     Fin Si
30.     valeursFitness=recuperer_fitness(population,donneesDesAttributs)
31.     variancePopulation=statistics.variance(valeursFitness)
32.     print("variance is",variancePopulation)
33.     gen=1
34.     Tant que(variance>critereDArret) :
35.      print('generation-', gen)
36.      parents=selectionner_parents(population,valeursFitness)
37.      croisementPopulation=croisement_a_deux_points(parents,0.78)
38.      population=croisementPopulation
39.      p=random.uniform(0,1)
```

```
40.      Si(p<=0.001) :
41.       muteePopulation=mutation(croisementPopulation)
42.       population=muteePopulation
43.      Fin Si
44.      valeursFitness=recuperer_fitness(population,donneesDesAttributs)
45.      variancePopulation=statistics.variance(valeursFitness)
46.      print("variance is",variancePopulation)
47.      gen+=1
48.     Fin Tant que
49.     fitnessOptimale=sum(valeursFitness)/len(valeursFitness)
50.     print('avg fitness is : ',fitnessOptimale)
51.     Pour index,valeurFitness a enumerate(valeursFitness) :
52.      erreur=abs((valeurFitness-fitnessOptimale)/
         fitnessOptimale)
53.      Si(erreur<=0.01) :
54.       meilleursAttributs=population[index]
55.       meilleurF1Score=valeurFitness
56.      Fin Si
57.     Fin Pour
58.     print(meilleursAttributs)
59.     return meilleursAttributs,meilleurF1Score
60.    Fin Fonction
61.    Fonction affichage(meilleursAttributs,donneesDesAttributs,
        meilleurF1Score)
62.    donneesDesAttributs,donneesDeLaCible=recuperation(
        donnees,attributs)
63.    donneesDesAttributs=normalisation(donneesDesAttributs)
64.    meilleursAttributs,meilleurF1Score=algorithme_genetique(
       donneesDesAttributs,donneesDesAttributs,donneesDeLaCible,
       tailleDeLaPopulation,critereDArret,nombreDAttributsASelectionner)
65.    affichage(meilleursAttributs,donneesDesAttributs,meilleurF1Score)
66.  Fin Algorithme
```

TABLE 3.6 – Implémentation de l'algorithme génétique de sélection d'attributs

Population et codage :

La population est l'ensemble des individus qui forment une génération. C'est un sous-ensemble de solutions possibles qui subit la reproduction.

Un chromosome représente un individu dans une population. En termes de calcul, le chromosome est représenté par une chaîne binaire. Pour la sélection des caractéristiques, la longueur du chromosome est considérée comme le nombre de caractéristiques dans l'ensemble de données. 0/1 indique la présence/absence de la $i^{ème}$ caractéristique dans la solution.

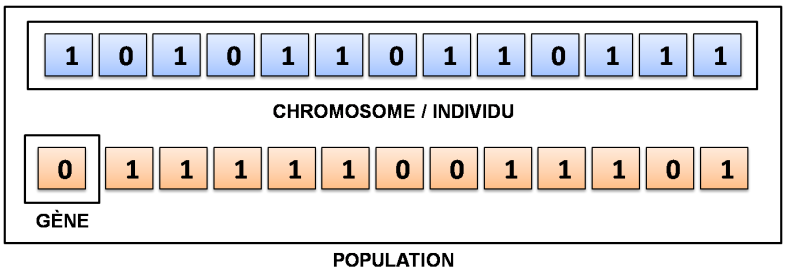


FIGURE 3.11 – Représentation de la population, du chromosome et du gène

Initialisation de la population :

Une population de grande taille ralentirait l'algorithme, tandis qu'une petite population ne présenterait pas de diversité. Il est donc important de choisir judicieusement la taille de la population. Ici, nous avons utilisé une taille de population de 20 ($\sim 1.5x$ (nombre de caractéristiques dans l'ensemble de données)).

init_population() initialise notre population pour trouver les N meilleures caractéristiques.

Calcul de la valeur fitness de chaque individu de la population :

La fonction Fitness évalue le degré d'adéquation d'un individu. Le terme "ajustement" indique dans quelle mesure la solution de l'individu est proche de la solution optimale. Un individu ayant une valeur de fitness élevée est considéré comme meilleur et a plus de chances d'être sélectionné pour la reproduction.

En général, la fonction de fitness est la même que la fonction d'optimisation. Par exemple, dans un problème de maximisation, la fonction d'aptitude sera la fonction à maximiser. Pour améliorer les performances du modèle, nous avons utilisé la mesure F de validation croisée du MLPClassifier() formé sur la solution de l'individu comme valeur de fitness.

La $F - mesure$ est utilisée comme métrique de précision car :

1. L'ensemble de données est assez déséquilibré, et l'utilisation de accuracy_score (mesure d'accuracy) pourrait indiquer un léger biais vers la classe majoritaire.
2. Une $F - mesure$ élevée indique une meilleure classification, c'est-à-dire une meilleure performance du modèle.

get_fitness() trouve les valeurs de fitness de la population en décodant chaque chromosome en un sous-ensemble de caractéristiques, et calculate_fitness() trouve la $F-mesure$ correspondante.

Sélection des parents pour la reproduction :

La sélection des parents, qui est l'une des étapes les plus cruciales, consiste à choisir des individus (parents) dans la population pour la reproduction, afin de produire la génération suivante.

La sélection des parents en fonction de leur aptitude est le critère largement accepté pour la sélection des parents. Il garantit que tous les individus ont une chance d'être sélectionnés comme parents avec une probabilité proportionnelle à leur valeur de fitness. De cette façon, l'idée sous-jacente à l'algorithme génétique serait justifiée.

Il existe plusieurs méthodes de sélection des parents, comme la sélection par tournoi, la sélection par roulette, l'échantillonnage universel stochastique, la sélection par rang, la sélection aléatoire, etc. Nous avons utilisé ici la sélection par roue de roulette.

Chromosome	**Valeur de fitness**	**Fitness normalisée**	**Fitness cumulée**
1	0.71	0.165120447	0.165120447
2	0.68	0.158150032	0.323270478
3	0.39	0.089372679	0.412643157
4	0.76	0.174795311	0.587438468
5	0.51	0.117126123	0.704564591
6	0.79	0.183458692	0.888023282
7	0.48	0.111976718	1

TABLE 3.7 – Choix de la roue de roulette

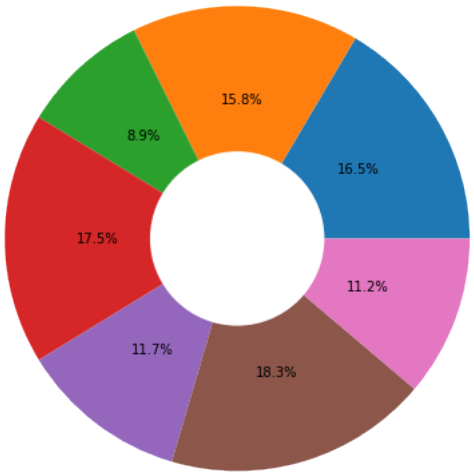


FIGURE 3.12 – Choix de la roue de roulette

Dans la sélection par roue à roulette, un point fixe est choisi sur le graphique circulaire préparé à l'aide des valeurs de fitness. À chaque rotation, l'individu qui se trouve devant le point est sélectionné pour la reproduction. Cela signifie qu'un individu ayant une plus grande surface sur le graphique circulaire (c'est-à-dire une plus grande valeur de fitness) a une forte probabilité d'être sélectionné.

Implémentation :

1. Trouver la somme de toutes les valeurs de fitness dans une population (S)
2. Trouver les valeurs de fitness normalisées ($= valeur\_de\_fitness/S$)

3. Trouver les valeurs de fitness cumulées (Tableau 3.7)
4. Générer un nombre aléatoire p entre $[0, 1]$
5. L'individu pour lequel p est juste inférieur à la somme cumulée est sélectionné. Par exemple, dans le tableau 3.7, considérez la colonne des valeurs de fitness normalisées. Si $p < 0.165$, l'individu 1 est sélectionné ; si $0.165 < p < 0.323$, l'individu 2 est sélectionné, et ainsi de suite.

select_parents() est l'implémentation de la sélection de la roue de la roulette.

Reproduction : croisement et mutation

La reproduction consiste à former une nouvelle génération par l'accouplement des parents. L'accouplement est mis en œuvre par le biais du processus de croisement. La mutation est utilisée pour ajouter un léger caractère aléatoire à l'individu afin d'introduire de la diversité dans la population.

Croisement :

Diverses opérations de croisement comme le croisement à un point, le croisement à deux points, le croisement uniforme sont utilisées. Ici, nous avons utilisé la technique du croisement à deux points qui consiste à échanger du matériel génétique entre deux points choisis au hasard sur les parents.

Habituellement, une probabilité est attribuée à ce processus, indiquant la chance de croisement pour une paire donnée. Le croisement est un événement à forte probabilité et on lui attribue une probabilité optimale comprise entre 0.65 et 0.80. Ici, nous avons utilisé une probabilité de 0.78.

two_point_crossover() est l'implémentation du croisement à deux points.

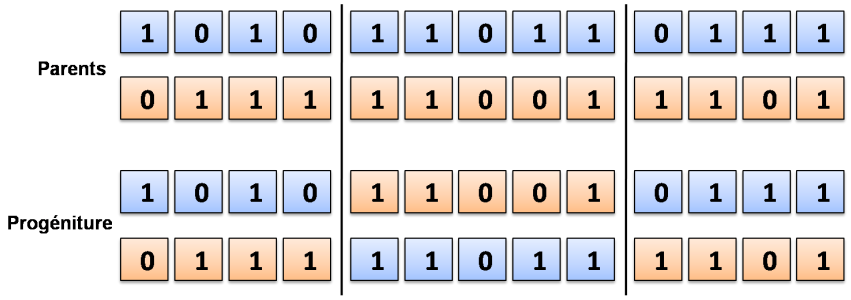


FIGURE 3.13 – Croisement à deux points

Mutation :

La mutation est utilisée pour introduire une légère variation dans le chromosome en modifiant l'un de ses gènes. Sa probabilité est maintenue très faible pour préserver l'intégrité de la population.

En général, la mutation est effectuée en échangeant de manière aléatoire n'importe quel bit d'un individu aléatoire de la population. En suivant le processus de mutation conventionnel, on a observé qu'après de nombreuses générations, le nombre de caractéristiques extraites s'écartait beaucoup de N. Pour réduire la déviation, nous avons échangé un bit '0' avec un bit '1'. De cette façon, la déviation a été réduite dans une bonne mesure.

mutation() est l'implémentation de la mutation.

Critère d'arrêt :

Après la reproduction, une nouvelle génération est formée, puis le critère d'arrêt est vérifié. Si la condition est satisfaite, l'algorithme se termine ; sinon, le processus est répété avec la population mutée comme population d'origine.

Quelques-uns des critères d'arrêt sont énumérés ci-dessous.

1. Fixer le nombre de générations : Ce n'est pas une très bonne méthode car nous pouvons manquer la meilleure génération à cause d'une limite supérieure du nombre de générations.
2. Absence de progrès dans la fitness du meilleur individu de la population : L'absence de progrès n'implique pas nécessairement une convergence car l'évolution se déroule avec des équilibres ponctués qui peuvent conduire ultérieurement à une amélioration.
3. Maintien d'une limite supérieure à la variance des valeurs de fitness dans une population : Lorsque les individus sont si semblables les uns aux autres que nous risquons de ne pas obtenir un meilleur individu dans les générations futures, l'algorithme indiquerait une convergence.

Nous avons utilisé le troisième critère d'arrêt.

Exécution de l'algorithme génétique

L'algorithme génétique est exécuté avec les valeurs des paramètres indiquées dans le tableau 3.8.

Nombre d'attributs	$13 + cible$
Taille de la population	20
Critère d'arrêt	0.00005
Nbr d'attributs à sélectionner	9

TABLE 3.8 – Valeurs des paramètres pour l'exécution de l'algorithme génétique

Résultats de l'exécution de l'algorithme génétique

Les résultats affichés par l'algorithme de sélection sont indiqués dans le tableau 3.9 et la figure 3.14.

Variance	$2.4978478825584272e-05$
Nombre de générations	48
AVG Fitness	0.7916631035136182
Attributs sélectionnés	['age', 'sex', 'cp', 'trestbps', 'fbs', 'thalach', 'exang', 'slope', 'thal']

TABLE 3.9 – Résultats de l'exécution de l'algorithme génétique

```
generation- 48
0.7922945481994915
0.7913485085835189
0.7872741490104381
0.7947611514008928
0.7882344673082449
0.7844603528241991
0.7845829225233523
0.7981609597626577
0.7881026541287616
0.8018983715883049
0.790905887152299
0.7878036867055409
0.7922523823667251
0.7943172571238424
0.7844549132431691
0.7944142976549189
0.7977349337345435
0.7938663612568495
0.7978335925686029
0.7885606731360129
variance is 2.4978478825584272e-05
avg fitness is:  0.7916631035136182
[1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1]
[1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1]
['age', 'sex', 'cp', 'trestbps', 'fbs', 'thalach', 'exang', 'slope', 'thal']
```

FIGURE 3.14 – Résultats de l'exécution de l'algorithme génétique

La liste d'attributs sélectionnés est la suivante : ['age', 'sex', 'cp', 'trestbps', 'fbs', 'thalach', 'exang', 'slope', 'thal']

3.5 Classification et validation

3.5.1 Technique utilisée pour améliorer l'évaluation des performances des modèles prédictifs

Nous décrivons la technique suivante qui amélioreraient l'évaluation des performances des modèles prédictifs sans sur-ajustement des modèles sur les données d'entraînement.

k-fold cross-validation : La validation croisée est une méthode de rééchantillonnage des données pour évaluer la capacité de généralisation des modèles prédictifs et pour éviter le surajustement [93, 94].

Dans la validation croisée k-fold, l'ensemble d'apprentissage disponible est divisé en k sous-ensembles disjoints de taille approximativement égale. Ici, « fold » fait référence au nombre de sous-ensembles résultants. Le modèle est entraîné à l'aide de $k-1$ sous-ensembles, qui,

ensemble, représentent l'ensemble d'apprentissage. Ensuite, le modèle est appliqué au sous-ensemble restant, qui est désigné comme l'ensemble de validation, et les performances sont mesurées. Cette procédure est répétée jusqu'à ce que chacun des k sous-ensembles ait servi d'ensemble de validation. La moyenne des k mesures de performance sur les k ensembles de validation est la performance validée croisée. La figure 3.15 illustre ce processus pour $k = 10$, c'est-à-dire une validation croisée de 10 fois. Dans le premier volet, le premier sous-ensemble sert d'ensemble de validation $D_{val,1}$ et les neuf sous-ensembles restants servent d'ensemble d'apprentissage $D_{train,1}$. Dans le deuxième volet, le deuxième sous-ensemble est l'ensemble de validation et les sous-ensembles restants sont l'ensemble d'apprentissage, et ainsi de suite.

La précision validée croisée, par exemple, est la moyenne des dix précisions obtenues sur les ensembles de validation. Plus généralement, notons $\hat{f} - k$ le modèle qui a été formé sur tout sauf le $k^{ième}$ sous-ensemble de l'ensemble d'apprentissage. La valeur $\hat{y}_i = \hat{f} - k(x_i)$ est la valeur prédite ou estimée pour l'étiquette de classe réelle, y_i, du cas x_i, qui est un élément du $k^{ième}$ sous-ensemble. L'estimation croisée de l'erreur de prédiction, $\hat{\epsilon}_{cv}$, est alors donnée par [95]

$$\hat{\epsilon}_{cv} = \frac{1}{n} \sum_{i=1}^{n} \mathfrak{L}(y_i, \hat{f} - k(x_i))$$

La validation croisée implique souvent un échantillonnage aléatoire stratifié, ce qui signifie que l'échantillonnage est effectué de telle manière que les proportions de classe dans les sous-ensembles individuels reflètent les proportions dans l'ensemble d'apprentissage.

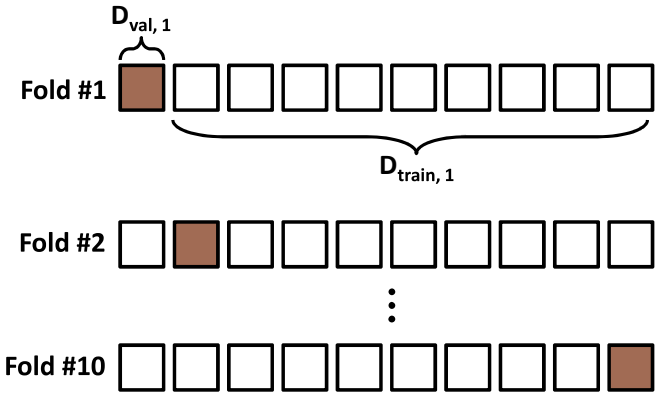


FIGURE 3.15 – 10-fold cross-validation

L'ensemble de données est divisé au hasard en dix sous-ensembles disjoints, chacun contenant (environ) 10% des données. Le modèle est entraîné sur l'ensemble d'apprentissage, puis appliqué à l'ensemble de validation.

3.5.2 Techniques utilisées pour améliorer les performances du modèle proposé

Nous énumérons les techniques suivantes qui amélioreraient les performances du modèle proposé.

Normalisation par lots : Nous avons ajouté une couche de normalisation par lots, après chaque couche, qui normalise les sorties de la couche précédente. Ceci est quelque peu similaire à la normalisation des données, sauf qu'il est appliqué aux sorties d'une couche, et que la moyenne et l'écart type sont des paramètres appris.

Formellement, l'algorithme de normalisation par lots [7] est défini comme indiqué dans la figure 3.16.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

FIGURE 3.16 – Algorithme : Transformation de normalisation par lots, appliquée à l'activation x sur un mini-lot [7]

Dans l'algorithme, B est utilisé pour désigner un mini-lot de taille m de l'ensemble d'apprentissage complet. La moyenne et la variance de B pourraient ainsi être calculées comme suit :

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i, and\ \sigma^2_B = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

Pour une couche avec une entrée d -dimensionnelle, chaque dimension de son entrée peut être normalisée (recentrée et remise à l'échelle) séparément. Ainsi, la normalisation pour une entrée d -dimensionnelle peut être calculée comme suit :

$x = (x_1, ..., x_d)$

$$\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{\sigma_B^{(k)^2} + \epsilon}}, where\ k \in [1, d] and\ i \in [1, m];$$

ϵ est ajouté au dénominateur de la stabilité numérique et est une constante arbitrairement petite.

Et enfin, pour restaurer la puissance de représentation du réseau, une étape de transformation est définie comme :

$$y_i^{(k)} = \gamma^{(k)} \hat{x}_i^{(k)} + \beta^{(k)}$$

où les paramètres β et γ sont ensuite appris dans le processus d'optimisation.

Optimiseur AdaBelief : Afin de comprendre AdaBelief (Adapting stepsizes by the Belief in observed gradients), nous devons d'abord comprendre les bases des optimiseurs stochastiques basés sur des gradients. Presque tous les optimiseurs d'apprentissage en profondeur appartiennent à cette catégorie et AdaBelief n'est pas différent [8, 96].

La descente de gradient stochastique (SGD) est l'optimiseur original basé sur le gradient. Il est facile à mettre en œuvre, fortement fondé sur la théorie, extrêmement stable pendant l'entraînement et donne des résultats compétitifs avec de nombreux autres optimiseurs avancés. L'idée est simple : calculez le gradient pour chaque paramètre, et faites un petit pas dans la direction du gradient. Si nous faisons cela plusieurs fois, en utilisant des lots d'échantillons sélectionnés au hasard (stochastiques) à partir des données d'apprentissage, notre modèle s'améliorera progressivement jusqu'à ce qu'il atteigne un minimum.

Il y a un problème majeur avec SGD : il converge trop lentement, en particulier dans les premières parties de la formation. Nous devons effectuer un grand nombre de mises à jour avant que le modèle ne commence à converger. Cela coûte un temps précieux et des ressources informatiques. De nombreuses variantes de SGD existent, qui tentent de résoudre ce problème. Parmi eux, l'optimiseur Adaptive moment estimation (Adam) est probablement le plus populaire et le plus connu (Figure 3.17).

Algorithm 1: Adam Optimizer

Initialize θ_0, $m_0 \leftarrow 0$, $v_0 \leftarrow 0$, $t \leftarrow 0$
While θ_t not converged
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
Update
$\theta_t \leftarrow \prod_{\mathcal{F}, \sqrt{v_t}} \left(\theta_{t-1} - \frac{\alpha m_t}{\sqrt{v_t} + \epsilon} \right)$

FIGURE 3.17 – Algorithme : Optimiseur Adam [8]

Adam introduit deux états internes pour chaque paramètre : le moment m et le carré v du gradient g. Avec chaque lot d'entraînement, chacun d'eux est mis à jour à l'aide de la moyenne pondérée exponentielle (EWA) :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

où les valeurs β sont fournies sous forme d'hyperparamètres. Ceux-ci sont ensuite utilisés pour mettre à jour les paramètres de chaque étape :

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}$$

où α est le taux d'apprentissage, et ϵ est ajouté pour améliorer la stabilité.

Les auteurs d'AdaBelief soulignent un problème important avec Adam. Lorsque le gradient est grand, mais que la variance est faible, Adam prédit une petite taille de pas. Cela n'a pas de sens intuitif - si l'ampleur et la direction du gradient sont cohérentes, nous nous attendons à faire des pas plus importants , car nous sommes plus convaincus que la direction du pas est correcte (Figure 3.18).

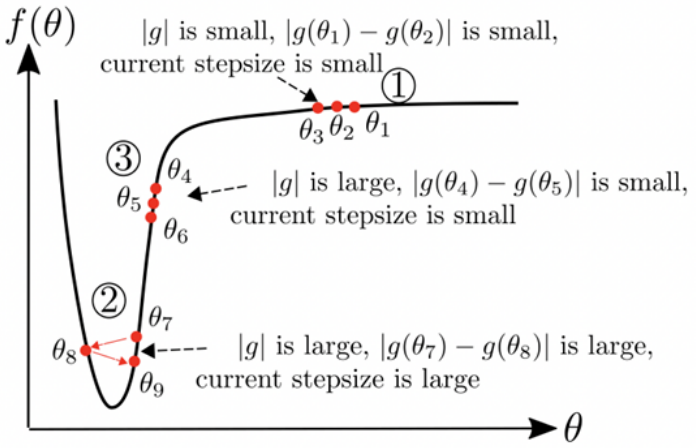


FIGURE 3.18 – Un optimiseur idéal considère la courbure de la fonction de perte, au lieu de prendre un grand (petit) pas où le gradient est grand (petit) [9]

AdaBelief corrige cela avec un changement très minime d'Adam (Figure 3.19).

Algorithm 2: AdaBelief Optimizer

Initialize θ_0, $m_0 \leftarrow 0$, $s_0 \leftarrow 0$, $t \leftarrow 0$
While θ_t not converged
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
$s_t \leftarrow \beta_2 s_{t-1} + (1 - \beta_2)(g_t - m_t)^2$
Update
$\theta_t \leftarrow \prod_{\mathcal{F}, \sqrt{s_t}} \left(\theta_{t-1} - \frac{\alpha m_t}{\sqrt{s_t} + \epsilon} \right)$

FIGURE 3.19 – Algorithme : Optimiseur AdaBelief [8]

Plutôt que de calculer l'élan au carré, AdaBelief calcule la variance du gradient dans le temps. Cette différence est subtile mais importante :

$$momentum\ squared: \ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$variance: \ s_t = \beta_2 s_{t-1} + (1 - \beta_2)(g_t - m_t)^2$$

C'est de là que vient le choix de AdaBelief, car la variance est calculée en utilisant notre momentum actuel estimé. La variance est essentiellement la distance au carré du gradient attendu (ou supposé). Et lorsque la variance est petite, la taille de notre pas reste grande !

3.5.3 Dense Deep Neural Network (Dense-DNN)

L'architecture du modèle proposé contient la couche dense[1] « Dense() » d'entrée avec « input_dim » pour définir la taille d'entrée et « units » pour définir le nombre de noyaux ainsi que « activation » pour définir la fonction d'activation « relu » (ReLU pour Rectified Linear Unit : Unité Linéaire Rectifiée).

Les trois couches denses suivantes sont les couches cachées de notre architecture, qui sont réalisées avec un nombre de noyaux ainsi qu'une fonction d'activation (ReLU).

Pour les calculs de probabilité de prédiction, ajout d'une couche de sortie. Cette dernière couche alimente une seule unité, qui est réalisée avec une fonction d'activation « sigmoid » (Sigmoïde).

Après chaque couche, nous implémentons la normalisation par lots « BatchNormalization() », qui normalise les sorties de la couche précédente.

Nous avons utilisé le modèle « Sequential() » pour enchaîner les couches du réseau neuronal profond.

Ensuite nous implémentons l'optimiseur « AdaBeliefOptimizer() ».

Le meilleur choix de fonction de perte « loss » pour une tâche de classification à deux classes est la fonction de perte d'entropie croisée binaire « binary_crossentropy », c'est donc ce que nous avons utilisé.

Le modèle proposé est enfin exécuté en utilisant :

— Une base de données nettoyée, avec des données normalisées, avec et sans réduction des données,

— un hyperparamètre « test_size » qui définit la taille de l'ensemble de test pour la validation du modèle avec la méthode split-validation,

— un hyperparamètre « validation_split » spécifiant la quantité de données d'entraînement qui sera utilisée pour la validation du modèle avec la méthode split-validation,

— un hyperparamètre « validation_data » spécifiant la quantité de données d'entraînement qui sera utilisée pour la validation du modèle avec la méthode cross-validation,

— une fonction d'échantillonnage « KFold » avec un hyperparamètre « n_splits » indiquant le nombre d'échantillons (blocs) sur lequel on divise l'échantillon original et un hyperparamètre « shuffle=True » pour mélanger les données utilisée avec la méthode cross-validation,

1. Dans tout réseau de neurones, une couche dense est une couche profondément connectée à sa couche précédente, ce qui signifie que les neurones de la couche sont connectés à chaque neurone de sa couche précédente. Le neurone de la couche dense dans un modèle reçoit la sortie de chaque neurone de sa couche précédente.

— la taille du lot « batch_size » qui définit le nombre d'échantillons à traiter avant de mettre à jour les paramètres du modèle interne,

— un nombre d'époques « epochs » qui définit le nombre de fois que l'algorithme d'apprentissage fonctionnera sur l'ensemble de données d'apprentissage,

— les métriques d'évaluation (Accuracy, Précision, Rappel et F-mesure), les métriques d'erreur (MAE et RMSE), la matrice de confusion, la courbe ROC et l'aire sous la courbe ROC (AUC) pour évaluer le modèle pendant les tests,

— et l'évaluation de la complexité pratique (Temporelle et Spatiale).

L'architecture du modèle proposé est illustrée dans la figure 3.20.

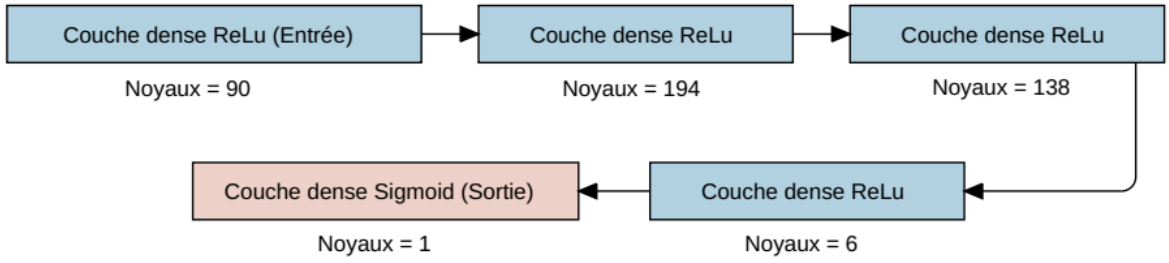


FIGURE 3.20 – Architecture du modèle proposé

L'implémentation de l'algorithme de classification proposé (Dense-DNN) avec split-validation est décrite dans le tableau 3.10 :

1.	**Algorithme** 2. Classification-Algorithme DNN Dense avec split-validation
2.	*# Récupérer et normaliser les données*
3.	Fonction recuperation(donnees,attributs)
4.	Fonction normalisation(donneesDesAttributs)
5.	*# Algorithme DNN Dense (Avec split-validation)*
6.	**Entrée**
7.	donneesDesAttributs,donneesDeLaCible
8.	**Sortie**
9.	modele,accuracy,precision,rappel,fscore,mae,rmse
10.	Fonction algo_dnn_dense(donneesDesAttributs,donneesDeLaCible)
11.	**Début**
12.	*# Fonctions de l'algorithme*
13.	Fonction train_test(donneesDesAttributs,donneesDeLaCible)
14.	Fonction modele_squentiel(donneesDesAttributs)
15.	Fonction optimisation()
16.	Fonction compilation(modele,optimiseur)
17.	Fonction entrainement(modele,donneesAttEntrain,donneesCibleEntrain)
18.	Fonction prediction(modele,donneesAttTest)
19.	Fonction resultats(donneesCibleTest,donneesCiblePred)
20.	donneesAttEntrain,donneesAttTest,donneesCibleEntrain, donneesCibleTest=train_test(donneesDesAttributs,donneesDeLaCible)
21.	modele=modele_squentiel(donneesDesAttributs)
22.	optimiseur=optimisation()
23.	compilation(modele,optimiseur)
24.	historique=entrainement(modele,donneesAttEntrain,donneesCibleEntrain)
25.	donneesCiblePred=prediction(modele,donneesAttTest)
26.	accuracy,precision,rappel,fscore,mae,rmse=resultats(donneesCibleTest, donneesCiblePred)
27.	return modele,accuracy,precision,rappel,fscore,mae,rmse
28.	**Fin Fonction**
29.	Fonction affichage(modele,donneesDesAttributs,accuracy,precision, rappel,fscore,mae,rmse)
30.	donneesDesAttributs,donneesDeLaCible=recuperation(donnees,attributs)
31.	donneesDesAttributs=normalisation(donneesDesAttributs)
32.	modele,accuracy,precision,rappel,fscore,mae,rmse= algo_dnn_dense(donneesDesAttributs,donneesDeLaCible)
33.	affichage(modele,donneesDesAttributs,accuracy,precision, rappel,fscore,mae,rmse)
34.	**Fin Algorithme**

TABLE 3.10 – Implémentation de l'algorithme de classification proposé (Dense-DNN) avec split-validation

L'implémentation de l'algorithme de classification proposé (Dense-DNN) avec cross-validation est décrite dans le tableau 3.11 :

```
1.  Algorithme 3. Classification-Algorithme DNN Dense avec cross-validation
2.    # Récupérer et normaliser les données
3.    Fonction recuperation(donnees,attributs)
4.    Fonction normalisation(donneesDesAttributs)
5.    # Algorithme DNN Dense (Avec cross-validation)
6.    Entrée
7.    donneesDesAttributs,donneesDeLaCible,nbrDEchnatillons
8.    Sortie
9.    modele,mAcc,mPre,mRap,mFs,mMAE,mRMSE
10.   Fonction algo_dnn_dense(donneesDesAttributs,donneesDeLaCible,
      nbrDEchnatillons)
11.   Début
12.     # Fonctions de l'algorithme
13.     Fonction validation_croisee(nbrDEchnatillons)
14.     Fonction modele_squentiel(donneesDesAttributs)
15.     Fonction optimisation()
16.     Fonction compilation(modele,optimiseur)
17.     Fonction entrainement(modele,donneesAttEntrain,donneesCibleEntrain,
        donneesAttTest,donneesCibleTest)
18.     Fonction prediction(modele,donneesAttTest)
19.     Fonction resultats(donneesCibleTest,donneesCiblePred)
20.     Fonction moyenne_resultats(accParEchant,preParEchant,rapParEchant,
        fsParEchant,maeParEchant,rmseParEchant)
21.     kEchant=validation_croisee(nbrDEchantillons)
22.     accParEchant=[ ] # Définir des conteneurs de résultat
      par échantillon
23.     preParEchant=[ ]
24.     rapParEchant=[ ]
25.     fsParEchant=[ ]
26.     maeParEchant=[ ]
27.     rmseParEchant=[ ]
28.     numEchant=1
29.     for train,test in kEchant.split(donneesDesAttributs,donneesDeLaCible) :
30.       donneesAttEntrain = donneesDesAttributs[train]
31.       donneesCibleEntrain = donneesDeLaCible[train]
32.       donneesAttTest = donneesDesAttributs[test]
33.       donneesCibleTest = donneesDeLaCible[test]
34.       modele=modele_squentiel(donneesDesAttributs)
35.       opt=optimisation()
36.       compilation(modele,opt)
37.       print("\n_____\n")
38.        print(f"Entrainement pour l'échantillon #numEchant
      ...")
39.       historique=entrainement(modele,donneesAttEntrain,
          donneesCibleEntrain,donneesAttTest,donneesCibleTest)
```

```
40.       donneesCiblePred=prediction(modele,donneesAttTest)
41.       accuracy,precision,rappel,fscore,mae,rmse=resultats(
          donneesCibleTest,donneesCiblePred)
42.       print(f"Accuracy de l'échantillon #numEchant : accu-
      racy")
43.       accParEchant.append(accuracy)
44.       preParEchant.append(precision)
45.       rapParEchant.append(rappel)
46.       fsParEchant.append(fscore)
47.       maeParEchant.append(mae)
48.       rmseParEchant.append(rmse)
49.       numEchant += 1
50.      mAcc,mPre,mRap,mFs,mMAE,mRMSE=moyenne_resultats(accParEchant,
          preParEchant,rapParEchant,fsParEchant,maeParEchant,rmseParEchant)
51.       return modele,mAcc,mPre,mRap,mFs,mMAE,mRMSE
52.     Fin Fonction
53.    Fonction affichage(modele,donneesDesAttributs,mAcc,mPre,mRec,
       mFs,mMAE,mRMSE)
54.    donneesDesAttributs,donneesDeLaCible=recuperation(donnees,attributs)
55.    donneesDesAttributs=normalisation(donneesDesAttributs)
56.    modele,mAcc,mPre,mRap,mFs,mMAE,mRMSE=algo_dnn_dense(
       donneesDesAttributs,donneesDeLaCible,5)
57.    affichage(modele,donneesDesAttributs,mAcc,mPre,mRap,mFs,mMAE,mRMSE)
58.  Fin Algorithme
```

TABLE 3.11 – Implémentation de l'algorithme de classification proposé (Dense-DNN) avec cross-validation

3.6 Conclusion

Dans ce troisième chapitre, nous avons d'abord présenté notre processus de construction du modèle prédictif puis nous avons visualisé nos données et effectué un prétraitement des données en vue de préparer les données pour les algorithmes, nous avons ensuite utilisé un algorithme génétique afin d'optimiser la recherche d'un sous-ensemble d'attributs pertinents et enfin, nous avons développé une nouvelle approche de classification basée sur l'apprentissage profond, expérimentée et évaluée sur une base de données pour la prédiction des maladies cardiovasculaires.

Dans le dernier chapitre nous allons évaluer les différents résultats obtenus et effectuer une étude comparative avec les différents algorithmes.

Chapitre 4

Résultats expérimentaux

4.1 Introduction

Dans ce chapitre, les résultats expérimentaux pour le modèle proposé et les sept autres classificateurs avec les deux méthodes de validation sont présentés, évalués, discutés et comparés sur la base de l'utilisation ou non de l'approche de sélection des caractéristiques et un test comparatif du modèle proposé avec les résultats réels de la cible est réalisé pour enfin sélectionner notre modèle.

4.2 Résultats et discussion

Une étude méthodologique comparative a été réalisée avec les algorithmes de classification supervisés les plus couramment utilisés en ML. Le même ensemble de données est testé avec différents classificateurs d'apprentissage automatique tels que la Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Gaussian Naive Bayes (GNB), K-Nearest Neighbors (KNN) et eXtreme Gradient Boosting (XGBoost). Dans ce travail, nous avons proposé un modèle de réseau profond dense (Dense-DNN) pour prédire avec précision si un patient est atteint d'une maladie cardiovasculaire ou non.

Les données nettoyées et normalisées sont divisées en 80% d'entraînement et 20% de test avec la méthode de validation « split-validation » et divisées en 5 échantillons (blocs) avec la méthode de validation « cross-validation » à des fins d'entraînement et de test des modèles ML et du modèle proposé, avec et sans réduction des données.

4.2.1 Métriques d'évaluation

Avant d'entamer l'étude comparative des métriques d'évaluation, nous allons effectuer une comparaison entre la méthode de validation croisée « cross-validation » et la méthode de validation « split-validation » de notre algorithme de classification proposé (Dense-DNN), avec et sans réduction des données (13/9 attributs). En fonction des résultats, nous ajusterons notre modèle jusqu'à ce qu'il fonctionne suffisamment bien pour le problème spécifique que nous essayons de résoudre.

Le tableau 4.1 indique les résultats obtenus en utilisant les deux méthodes de validation dans l'algorithme de classification proposé (Dense-DNN), avec et sans réduction des données.

Méthode	**Acc (13/9)**	**Pre (13/9)**	**Rap (13/9)**	**FM (13/9)**
split-validation	0.917/0.950	0.919/0.946	0.944/0.972	0.932/0.959
cross-validation	0.848/0.854	0.830/0.842	0.891/0.887	0.858/0.863

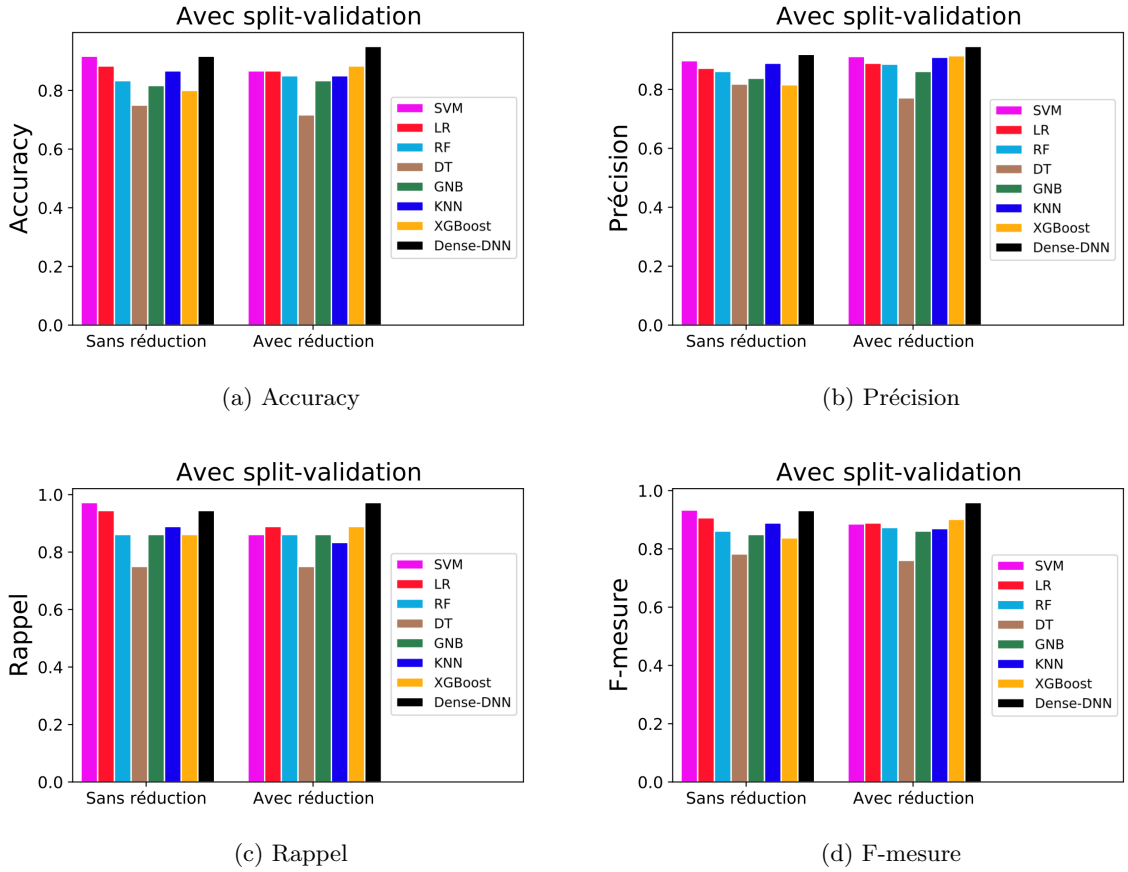
TABLE 4.1 – Comparaison entre les deux méthodes de validation

Métriques d'évaluation avec split-validation

Les résultats des métriques d'évaluation du modèle de réseau profond proposé sont comparés dans le tableau 4.2 et la figure 4.1 aux modèles ML avec la méthode de validation « split-validation », avec et sans réduction des données.

Modèle	**Acc (13/9)**	**Pre (13/9)**	**Rap (13/9)**	**FM (13/9)**
SVM	0.917/0.867	0.897/0.912	0.972/0.861	0.933/0.886
LR	0.883/0.867	0.872/0.889	0.944/0.889	0.907/0.889
RF	0.833/0.850	0.861/0.886	0.861/0.861	0.861/0.873
DT	0.750/0.717	0.818/0.771	0.750/0.750	0.783/0.761
GNB	0.817/0.833	0.838/0.861	0.861/0.861	0.849/0.861
KNN	0.867/0.850	0.889/0.909	0.889/0.833	0.889/0.870
XGBoost	0.800/0.883	0.816/0.914	0.861/0.889	0.838/0.901
Dense-DNN	**0.917/0.950**	**0.919/0.946**	**0.944/0.972**	**0.932/0.959**

TABLE 4.2 – Métriques d'évaluation des différents modèles avec split-validation



(a) Accuracy

(b) Précision

(c) Rappel

(d) F-mesure

FIGURE 4.1 – Représentation graphique des métriques d'évaluation avec split-validation

Comme nous pouvons le voir, le modèle d'apprentissage en profondeur proposé a obtenu les plus grands résultats en utilisant la méthode de sélection d'attributs.

En comparaison avec les résultats de la méthode sans réduction des données, les résultats de la SVM, de la régression logistique, de l'arbre de décision et du K-Plus Proche Voisin avec réduction des données ont diminué, et les résultats de la forêt aléatoire, de Bayes naïf gaussien, du XGBoost et du modèle proposé ont augmenté.

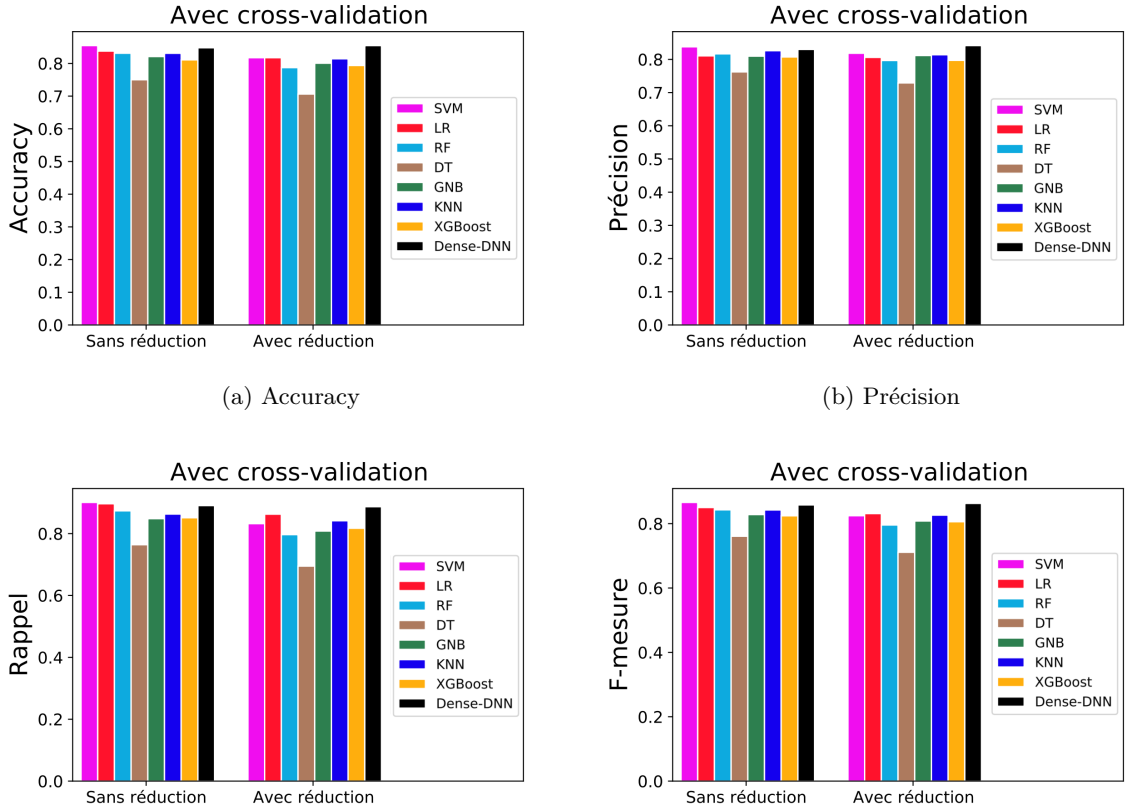
Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore et optimise les performances de l'algorithme en profondeur proposé.

Métriques d'évaluation avec cross-validation

Les résultats des métriques d'évaluation du modèle de réseau profond proposé sont comparés dans le tableau 4.3 et la figure 4.2 aux modèles ML avec la méthode de validation « cross-validation », avec et sans réduction des données.

Modèle	Acc (13/9)	Pre (13/9)	Rap (13/9)	FM (13/9)
SVM	0.855/0.817	0.838/0.818	0.901/0.832	0.866/0.825
LR	0.838/0.817	0.811/0.806	0.896/0.863	0.850/0.831
RF	0.831/0.787	0.817/0.797	0.874/0.797	0.843/0.796
DT	0.750/0.706	0.762/0.729	0.764/0.695	0.761/0.711
GNB	0.821/0.801	0.810/0.812	0.848/0.808	0.828/0.808
KNN	0.831/0.814	0.826/0.814	0.863/0.841	0.842/0.826
XGBoost	0.811/0.794	0.807/0.797	0.851/0.817	0.824/0.806
Dense-DNN	**0.848/0.854**	**0.830/0.842**	**0.891/0.887**	**0.858/0.863**

TABLE 4.3 – Métriques d'évaluation des différents modèles avec cross-validation



(a) Accuracy

(b) Précision

(c) Rappel

(d) F-mesure

FIGURE 4.2 – Représentation graphique des métriques d'évaluation avec cross-validation

Comme nous pouvons le voir, le modèle d'apprentissage en profondeur proposé a obtenu les plus grands résultats en utilisant la méthode de sélection d'attributs.

En comparaison avec les résultats de la méthode sans réduction des données, les résultats de la SVM, de la régression logistique, de la forêt aléatoire, de l'arbre de décision, de Bayes naïf gaussien, du K-Plus Proche Voisin et du XGBoost avec réduction des données ont diminué et les résultats du modèle proposé ont augmenté.

Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore et optimise

les performances de l'algorithme en profondeur proposé.

4.2.2 Métriques d'erreur

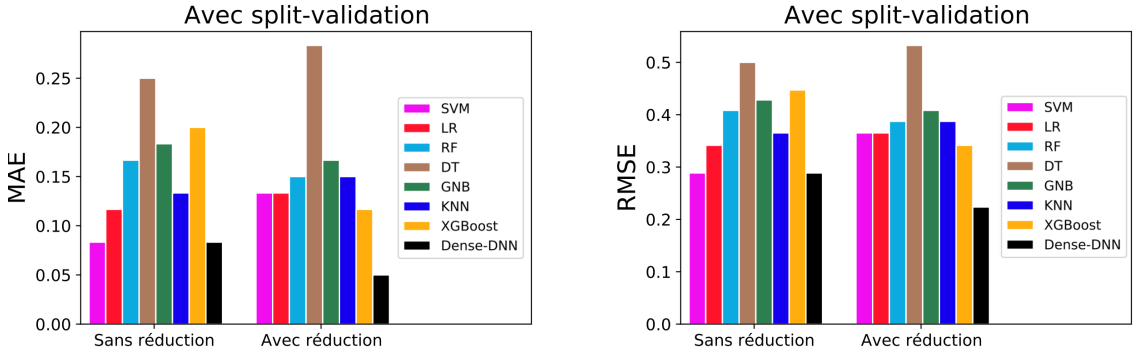
Nous avons utilisé les métriques d'erreur les plus utilisées dans la littérature avec les deux méthodes de validation, avec et sans réduction des données.

Métriques d'erreur avec split-validation

Les résultats des métriques d'erreur du modèle de réseau profond proposé et des modèles ML avec la méthode de validation « split-validation », avec et sans réduction des données sont indiqués dans le tableau 4.4 et la figure 4.3.

Modèle	MAE (13/9 attr)	RMSE (13/9 attr)
SVM	0.083/0.133	0.289/0.365
LR	0.117/0.133	0.342/0.365
RF	0.167/0.150	0.408/0.387
DT	0.250/0.283	0.500/0.532
GNB	0.183/0.167	0.428/0.408
KNN	0.133/0.150	0.365/0.387
XGBoost	0.200/0.117	0.447/0.342
Dense-DNN	**0.083/0.050**	**0.289/0.224**

TABLE 4.4 – Métriques d'erreur des différents modèles avec split-validation



(a) Erreur absolue moyenne (MAE)

(b) Racine de l'erreur quadratique moyenne (RMSE)

FIGURE 4.3 – Représentation graphique des métriques d'erreur avec split-validation

Comme nous pouvons le voir, le modèle d'apprentissage en profondeur proposé a obtenu les meilleurs résultats en utilisant la méthode de sélection d'attributs.

En comparaison avec les résultats de la méthode sans réduction des données, la MAE et la RMSE de la SVM, de la régression logistique, de l'arbre de décision et du K-Plus Proche Voisin

avec réduction des données ont augmenté et la MAE et la RMSE de la forêt aléatoire, de Bayes naïf gaussien, du XGBoost et du modèle proposé ont diminué.

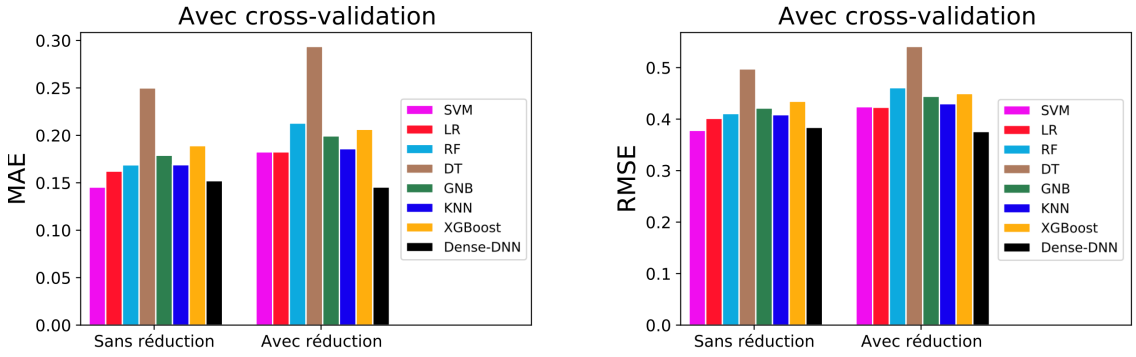
Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore et optimise les performances de l'algorithme en profondeur proposé.

Métriques d'erreur avec cross-validation

Les résultats des métriques d'erreur du modèle de réseau profond proposé et des modèles ML avec la méthode de validation « cross-validation », avec et sans réduction des données sont indiqués dans le tableau 4.5 et la figure 4.4.

Modèle	**MAE (13/9 attr)**	**RMSE (13/9 attr)**
SVM	0.145/0.183	0.378/0.424
LR	0.162/0.183	0.401/0.423
RF	0.169/0.213	0.411/0.461
DT	0.250/0.294	0.498/0.541
GNB	0.179/0.199	0.421/0.444
KNN	0.169/0.186	0.409/0.430
XGBoost	0.189/0.206	0.435/0.450
Dense-DNN	**0.152/0.146**	**0.384/0.376**

TABLE 4.5 – Métriques d'erreur des différents modèles avec cross-validation



(a) Erreur absolue moyenne (MAE)

(b) Racine de l'erreur quadratique moyenne (RMSE)

FIGURE 4.4 – Représentation graphique des métriques d'erreur avec cross-validation

Comme nous pouvons le voir, le modèle d'apprentissage en profondeur proposé a obtenu les meilleurs résultats en utilisant la méthode de sélection d'attributs.

En comparaison avec les résultats de la méthode sans réduction des données, la MAE et la RMSE de la SVM, de la régression logistique, de la forêt aléatoire, de l'arbre de décision, de Bayes naïf gaussien, du K-Plus Proche Voisin et du XGBoost avec réduction des données ont augmenté et la MAE et la RMSE du modèle proposé ont diminué.

Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore et optimise les performances de l'algorithme en profondeur proposé.

4.2.3 Evaluation de la complexité pratique

L'évaluation de la complexité pratique est divisée en deux types :

Complexité temporelle : La complexité temporelle décrit quantitativement le temps d'exécution de l'algorithme. Le temps qu'il faut pour exécuter un algorithme. Pour le faire on doit lancer le programme sur la machine. Le temps consacré à un algorithme est proportionnel au nombre de fois où les instructions sont exécutées et le nombre d'opérations de base effectuées dans l'algorithme.

Complexité spatiale : La complexité de l'espace est une mesure de la taille de l'espace de stockage temporairement occupé par un algorithme pendant qu'il est en cours d'exécution. La complexité spatiale compte le nombre de paramètres du modèle.

Les différents algorithmes de notre modèle sont exécutés sur la plateforme collaborative « Collab [1] » avec les caractéristiques suivantes :

RAM du système utilisée : 1.32 GB

Débit internet minimum : 17.83 Mbps

Evaluation de la complexité temporelle avec split-validation

Le tableau 4.6 indique les résultats des mesures de la complexité temporelle, obtenues en utilisant la méthode de validation « split-validation » dans l'algorithme de classification proposé (Dense-DNN), avec et sans réduction des données.

Données	**Sans réduction (13 attr)**	**Avec réduction (9 attr)**
Temps d'exécution (seconde)	24	20

TABLE 4.6 – Evaluation de la complexité temporelle « split-validation »

Comme nous pouvons le voir, le temps d'exécution de l'algorithme avec « split-validation » en utilisant la méthode de sélection d'attributs a diminué de 4 secondes.

Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore le temps d'exécution de l'algorithme en profondeur proposé.

1. https ://colab.research.google.com/

Evaluation de la complexité temporelle avec cross-validation

Le tableau 4.7 indique les résultats des mesures de la complexité temporelle, obtenues en utilisant la méthode de validation « cross-validation » dans l'algorithme de classification proposé (Dense-DNN), avec et sans réduction des données.

Données	**Sans réduction (13 attr)**	**Avec réduction (9 attr)**
Temps d'exécution (seconde)	65	60

TABLE 4.7 – Evaluation de la complexité temporelle « cross-validation »

Comme nous pouvons le voir, le temps d'exécution de l'algorithme avec « cross-validation » en utilisant la méthode de sélection d'attributs a diminué de 5 secondes.

Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore le temps d'exécution de l'algorithme en profondeur proposé.

Evaluation de la complexité spatiale avec split-validation

Le tableau 4.8 indique les résultats des mesures de la complexité spatiale, obtenues en utilisant la méthode de validation « split-validation » dans l'algorithme de classification proposé (Dense-DNN), avec et sans réduction des données.

Données	**Sans réduction (13 attr)**	**Avec réduction (9 attr)**
Total params	48377	48017
Trainable params	47521	47161
Non-trainable params	856	856

TABLE 4.8 – Evaluation de la complexité spatiale « split-validation »

Comme nous pouvons le voir, le nombre de paramètres total du modèle avec l'algorithme « split-validation » en utilisant la méthode de sélection d'attributs a diminué de 360 paramètres en diminuant le nombre de paramètres entrainables de 360 paramètres.

Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore la taille de l'espace de stockage temporairement occupé par l'algorithme en profondeur proposé pendant qu'il est en cours d'exécution.

Evaluation de la complexité spatiale avec cross-validation

Le tableau 4.9 indique les résultats des mesures de la complexité spatiale, obtenues en utilisant la méthode de validation « cross-validation » dans l'algorithme de classification proposé (Dense-DNN), avec et sans réduction des données.

Données	Sans réduction (13 attr)	Avec réduction (9 attr)
Total params	48377	48017
Trainable params	47521	47161
Non-trainable params	856	856

TABLE 4.9 – Evaluation de la complexité spatiale « cross-validation »

Comme nous pouvons le voir, le nombre de paramètres total du modèle avec l'algorithme « cross-validation » en utilisant la méthode de sélection d'attributs a diminué de 360 paramètres en diminuant le nombre de paramètres entrainables de 360 paramètres.

Les résultats obtenus indiquent que la méthode de sélection d'attributs améliore la taille de l'espace de stockage temporairement occupé par l'algorithme en profondeur proposé pendant qu'il est en cours d'exécution.

4.2.4 Matrice de confusion

Dans notre travail, la fonction fitness est prise comme maximum d'accuracy. La représentation graphique de la matrice de confusion du modèle proposé avec la méthode de validation « split-validation », avec réduction des données est montrée dans la figure 4.5.

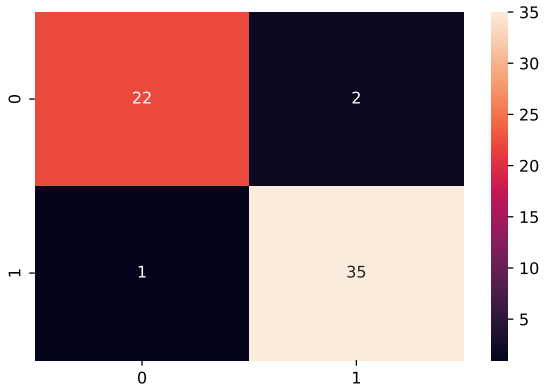


FIGURE 4.5 – Matrice de confusion

Les résultats montrent que la valeur des vrais positifs de la matrice de confusion est de 22 et celle des vrais négatifs de 35. Et le faux positif est de 1 et le faux négatif de 2.

Regardons maintenant plus en détail notre modèle. Nous pouvons voir qu'il a très bien fonctionné car il a un score d'accuracy très élevé. Sur les 24 personnes qui avaient une maladie cardiovasculaire, le modèle n'en a diagnostiqué que deux par erreur. Cependant, nous pouvons voir que le nombre de personnes qui n'avaient pas de maladie cardiovasculaire mais dont on prévoyait qu'elles auraient une maladie cardiovasculaire est égal à 1. Cela signifie que nous

sauvons de nombreuses personnes, sans avoir recours à beaucoup de tests, ce qui peut limiter le gaspillage de ressources.

4.2.5 Courbe ROC et AUC

La courbe ROC du modèle proposé avec la méthode de validation « split-validation », avec réduction des données résumant le compromis entre le taux de vrais positifs et le taux de faux positifs pour le modèle prédictif en utilisant différents seuils de probabilité est montrée dans la figure 4.6.

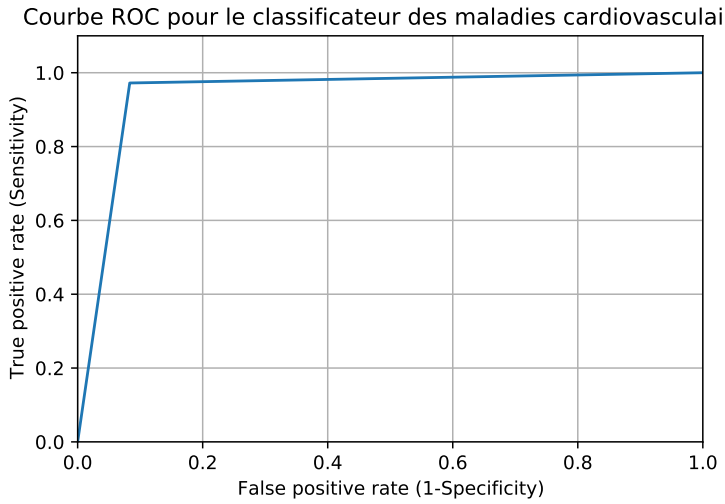


FIGURE 4.6 – Courbe ROC

L'aire sous la courbe ROC du modèle proposé avec la méthode de validation « split-validation » et avec réduction des données est de 0.9444444444444445 (94%). Une augmentation de l'AUC indique une amélioration des capacités discriminatoires.

Le modèle prédictif proposé avec la méthode de validation « split-validation » et avec réduction des données est sélectionné pour le test qui va suivre.

4.3 Test du modèle prédictif proposé

Le tableau 4.10 indique les résultats des prédictions, obtenus en utilisant le modèle prédictif proposé (avec split-validation et réduction des données).

Pour réaliser ce test, nous avons choisi aléatoirement cinq patients malades et cinq patients non malades dans notre base de données.

Données des patients	Valeur réelle	Valeur prédite	Gravité de la maladie
P1 : [49,1,1,130,0,171,0,2,2]	1	1	94%
P2 : [64,1,3,110,0,144,1,1,2]	1	1	70%
P3 : [58,0,3,150,1,162,0,2,2]	1	1	97%
P4 : [50,0,2,120,0,158,0,1,2]	1	1	94%
P5 : [58,0,2,120,0,172,0,2,2]	1	1	92%
P6 : [68,1,2,180,1,150,1,1,3]	0	0	27%
P7 : [62,0,0,160,0,145,0,0,3]	0	0	19%
P8 : [52,1,0,128,0,161,1,2,3]	0	0	16%
P9 : [59,1,0,110,0,142,1,1,3]	0	0	20%
P10 : [60,0,0,150,0,157,0,1,3]	0	0	4%

TABLE 4.10 – Test du modèle prédictif proposé

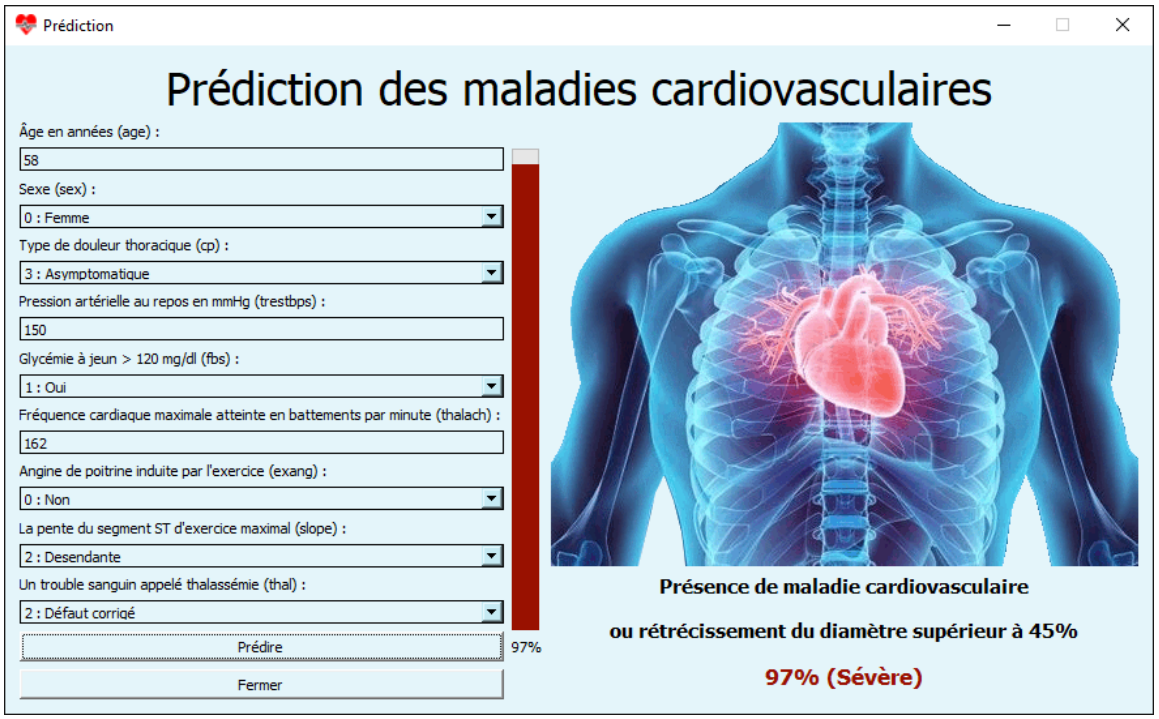
Nous constatons un total succès de la prédiction de la présence comme de l'absence de la maladie chez les 10 patients testés.

Notre modèle prédictif est parvenu à classer correctement 100% des patients selon leurs données médicales en deux classes et a fournit également la gravité de la maladie pour chacun des patients, ce qui permet aux praticiens de prendre les mesures nécessaires pour traiter ou prévenir des risques potentiels ou effectifs.

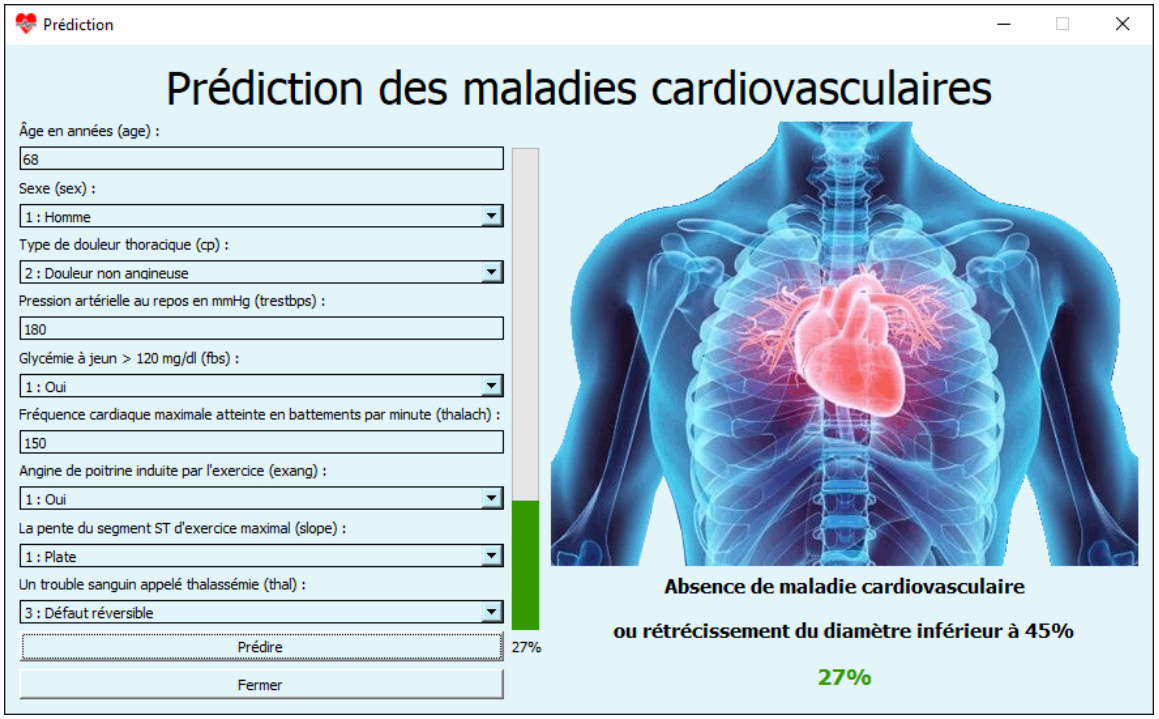
4.3.1 Test sur application de bureau

La figure 4.7 montre les résultats des prédictions sur une application de bureau réalisée avec le langage de programmation interprété « Python » pour les besoins de ce test, en utilisant le modèle prédictif proposé (avec split-validation et réduction des données).

Nous avons choisi deux patients (P3 et P6) parmi les dix patients du tableau 4.10 pour montrer l'affichage des résultats sur notre application.



(a) Patient 3 (P3)



(b) Patient 6 (P6)

FIGURE 4.7 – Test sur application de bureau

Le modèle prédictif proposé est efficace et fournit des résultats justes et précis.

4.4 Conclusion

Dans ce quatrième et dernier chapitre, nous avons d'abord représenté graphiquement, évalué la performance et discuté les résultats de plusieurs modèles ML et du modèle proposé avec les deux méthodes de validation, avec et sans réduction des données, représenté graphiquement, évalué et discuté les résultats des métriques d'erreur les plus utilisées dans la littérature du modèle proposé et des modèles ML avec les deux méthodes de validation, avec et sans réduction des données et évalué et discuté la complexité pratique du modèle proposé avec les deux méthodes de validation, avec et sans réduction des données. Nous avons ensuite représenté graphiquement et discuté les résultats de la matrice de confusion, de la courbe ROC et de l'aire sous la courbe ROC (AUC) du modèle proposé avec la méthode de validation « split-validation » avec réduction des données. Nous avons enfin sélectionné notre modèle prédictif proposé et effectué un test sur 10 patients choisis aléatoirement dans notre base de données et les résultats des prédictions de deux patients parmi eux sont affichés sur une application de bureau réalisée pour les besoins de ce test.

Conclusion générale et perspectives

Dans ce travail, nous avons présenté un système de soutien pour aider le personnel soignant au diagnostique rapide et à la prédiction des maladies cardiovasculaires en utilisant un modèle d'apprentissage en profondeur pour améliorer l'accuracy de la prédiction des maladies cardiovasculaires.

De nombreux problèmes raisonnables sont discutés, y compris la collecte de données physiologiques, la gestion des valeurs manquantes et des doublons pour obtenir des résultats efficaces, la normalisation des données, l'évaluation de la dépendance à l'aide de la matrice de corrélation, la sélection de caractéristiques importantes en utilisant un algorithme génétique, la prédiction des maladies cardiovasculaires à l'aide d'un modèle d'apprentissage en profondeur avec les techniques d'amélioration des performances les plus fiables, notamment : la normalisation par lots et l'optimiseur AdaBelief, des méthodes de validation différentes (split-validation et cross-validation), l'évaluation du modèle à l'aide des métriques d'évaluation et des métriques d'erreur telles que l'accuracy, la précision, le rappel, la F mesure, l'erreur absolue moyenne et la racine de l'erreur quadratique moyenne, la complexité pratique, la matrice de confusion, la courbe ROC et l'aire sous la courbe ROC.

La méthode proposée offre un système de prédiction qui détecte les facteurs de risque les plus importants dans les données de santé de grande dimension et les analyse de manière critique pour prédire avec précision les maladies cardiovasculaires avant qu'une crise cardiaque ou un accident vasculaire cérébral ne se produisent.

En effet, le modèle d'apprentissage en profondeur proposé gère efficacement les données et améliore les performances des diagnostics de maladies cardiovasculaires, ce qui augmente l'accuracy des prédictions. En outre, cette méthode peut prédire le risque de maladie, car elle peut extraire des caractéristiques précieuses à partir de données structurées, et représente ces caractéristiques extraites efficacement avec un faible poids dimensionnel et spécifique afin d'améliorer les performances de prédiction des maladies cardiovasculaire.

Au cours des dernières années, les objets connectés de santé ont progressé rapidement et il est devenu possible de surveiller à distance et en temps réel les signaux physiologiques des patients comme la tension artérielle, l'ECG, le taux de graisse sanguin, etc.

Dans les travaux futurs, les performances de la sélection de caractéristiques seront améliorées en utilisant des méthodes et des techniques de sélection plus précises telles que les méthodes enveloppes (Wrapper methods) qui effectuent une recherche dans l'espace des sous-ensembles de variables, guidée par le résultat du modèle, et les méthodes embarquées (Embedded methods) qui utilisent l'information interne du modèle de classification, pour produire des résultats optimaux. De plus, de nouvelles méthodes seront conçues dans nos travaux futurs pour la réduction des caractéristiques afin de gérer un grand nombre de caractéristiques et qui seront plus particulièrement adaptées aux exigences de la santé intelligente. Une méthode plus sophistiquée sera étudiée pour améliorer les performances de notre modèle et utiliser les méthodes de validation les plus fiables. Enfin, une application ergonomique et adaptée au contexte de la santé publique sera réalisée et mise à disposition des praticiens et du public afin d'améliorer le diagnostique et la prédiction des maladies les plus dangereuses telles que les maladies contagieuses et les maladies les plus mortelles en impliquant la technologie de l'internet des objets dans le secteur de la santé.

Bibliographie

[1] C. Latrémouille* and F. Lintz. Anatomie du coeur, 2005. *(Professeur des Universités, praticien hospitalier).

[2] Le service de cardiologie du Centre Hospitalier Universitaire Vaudois (CHUV) Lausanne (Suisse). https ://www.chuv.ch/fr/cardiologie/car-home/patients-et-famille/fonctionnement-du-coeur (Centre de transplantation d'organes) [Dernière consultation : 15/06/2022].

[3] BruceBlaus. Multipolar neuron, sep 2013. https ://upload.wikimedia.org/wikipedia/commons/thumb/1/10/Blausen_0657_MultipolarNeuron.png/350px-Blausen_0657_MultipolarNeuron.png [Dernière consultation : 15/06/2022].

[4] Lakshmi Bhargav Jetti. User authentication based on keystroke dynamics using multilayer perceptron, aug 2021. National College of Ireland, MSc Project Submission Sheet, School of Computing.

[5] Mustafa Salam Kadhm AL-Shammari. Arabic handwritten text recognition and writer identification, 2016. A Thesis Submitted to the Department of Computer Science of the University of Technology, Iraq.

[6] Gopi Battineni, Getu Gamo Sagaro, Chintalapudi Nalini, Francesco Amenta, and Seyed Khosrow Tayebati. Comparative machine-learning approach : A follow-up study on type 2 diabetes predictions by cross-validation methods, dec 2019. Machines.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift, 2015. arXiv :1502.03167.

[8] Juntang Zhuang, Tommy Tang, Sekhar Tatikonda, Nicha Dvornek, et al. Adabelief optimizer : Adapting stepsizes by the belief in observed gradients, 2020. 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

[9] Marc Toussaint. Lecture notes : Some notes on gradient descent, may 2012. Machine Learning and Robotics lab, FU Berlin, Germany.

[10] Organisation mondiale de la santé (OMS). https ://www.who.int/fr/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) [Dernière consultation : 15/06/2022].

[11] M.Akhil jabbar, Priti Chandra, and B.L Deekshatulu. Prediction of risk score for heart disease using associative classification and hybrid feature subset selection, nov 2012. International Conference on Intelligent Systems Design and Applications (ISDA).

[12] Zhu Xiaojin. Semi-supervised learning literature survey, sep 2005. Computer Sciences Department, University of Wisconsin, Madison. Technical Report 1530.

[13] Hebb D. O. The organization of behavior, a neuropsychological theory, 1949. McGill University, Montréal, Canada.

[14] Li I. Zhang, Huizhong W. Tao, Christine E. Holt, William A. Harris, and Mu ming Poo. A critical window for cooperation and competition among developing retinotectal synapses, sep 1998. Department of Biology, University of California at San Diego, La Jolla, California 92093-0357, USA.

[15] Droniou Alain. Apprentissage de représentations et robotique développementale : quelques apports de l'apprentissage profond pour la robotique autonome, apr 2015.

[16] Werbos Paul. Beyond regression : New tools for prediction and analysis in the behavioral sciences, 1974. Thèse de Doctorat.

[17] Parker David B. Learning logic, 1985.

[18] Le Cun Yann. Learning process in an asymmetric threshold network, 1986. Disordered systems and biological organization. Springer.

[19] D.E Rumelhart, G.E Hintont, and R.J Williams. Learning representations by back-propagating errors, 1986. Nature 323.6088.

[20] Nemissi Mohamed. Classification et reconnaissance des formes par algorithmes hybrides, 2009. Thèse de Doctorat.

[21] Preux Philippe. Fouille de données, notes de cours, may 2011. Université de Lille 3.

[22] Nguyen Cong Long, Phayung Meesad, and Herwig Unger. A highly accurate firefly based algorithm for heart disease prediction, 2015. Expert Systems with Applications.

[23] Subhashini Narayan and E. Sathiyamoorthy. A novel recommender system based on fft with machine learning for predicting and identifying heart diseases, 2019. Neural Computing and Applications.

[24] Boudheb Tarik. Privacy preserving classification of biomedical data, jul 2019. Thèse de Doctorat.

[25] Xiaohui Yuan, Lijun Xie, and Mohamed Abouelenien. A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data, dec 2017.

[26] Yong Liu, Feng Tang, and Zhiyong Zeng. Feature selection based on dependency margin, 2015. IEEE TRANSACTIONS ON CYBERNETICS.

[27] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection, mar 2003. Journal of Machine Learning Research 1157-1182.

[28] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection : A data perspective, dec 2017.

[29] Seppo Puuronen, Alexey Tsymbal, and Iryna Skrypnyk. Advanced local feature selection in medical diagnostics, 2000. IEEE.

[30] K.Rajeswari, V.Vaithiyanathan, and Shailaja V.Pede. Feature selection for classification in medical data mining, 2013. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 2, March-April 2013.

[31] H. Hannah Inbarani, Ahmad Taher Azar, and G. Jothi. Supervised hybrid feature selection based on pso and rough sets for medical diagnosis, 2014. Computer Methods and Programs in Biomedicine 113 (2014) 175-185.

[32] Lai Po Hung, Rayner Alfred, and Mohd Hanafi Ahmad Hijazi. A review on feature selection methods for sentiment analysis, oct 2015. Journal of Computational and Theoretical Nanoscience.

[33] RL Babu and S Vijayan. Wrapper based feature selection in semantic medical information retrieval, jun 2016. American Scientific Publishers.

[34] George Forman. An extensive empirical study of feature selection metrics for text classification, mar 2003. Journal of Machine Learning Research.

[35] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval : An experimental comparison, nov 2007.

[36] Kesari Verma, Bikesh Kumar Singh, Priyanka Tripathi, and A.S. Thoke. Review of feature selection algorithms for breast cancer ultrasound image, 2015. Springer International Publishing Switzerland.

[37] Steven L. Salzberg, Arthur L. Delcher, Simon Kasif, and Owen White. Microbial gene identification using interpolated markov models, jan 1998. Nucleic Acids Research, Volume 26, Issue 2.

[38] Yvan Saeys, Pierre Rouzé, and Yves Van de Peer. In search of the small ones : improved prediction of short exons in vertebrates, plants, fungi and protists, jan 2007. Bioinformatics.

[39] Ali Keles and Ayturk Keles. Expert system for thyroid diseases diagnosis (estdd), 2008. Expert Systems with Applications.

[40] Mahlet G. Tadesse, Marina Vannucci, and Pietro Liò. Identification of dna regulatory motifs using bayesian variable selection, apr 2004. Bioinformatics.

[41] Huiqing Liu, Hao Han, Jinyan Li, and Limsoon Wong. Using amino acid patterns to accurately predict translation initiation sites, 2004. In Silico Biology.

[42] R. L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data : curses, caveats, cautions, 2003. Bioinformatics.

[43] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer : Class discovery and class prediction by gene expression monitoring, 1999. Science.

[44] U. Alon, N.Barkai, D.A.Notterman, K. Gish, S.Ybarra, D.Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, jun 1999. Natl. Acad. Sci. USA.

[45] Amir Ben-Dor, Laurakay Bruhn, Nir Friedman, Iftach Nachman, Michel Schummer, and Zohar Yakhini. Tissue classification with gene expression profiles, 2000. J. Comput. Biol.

[46] Douglas T. Ross, Uwe Scherf, Jeffrey C.F. Lee, John N. Weinstein, Patrick O. Brown, et al. Systematic variation in gene expression patterns in human cancer cell lines, mar 2000. Nature Genetics.

[47] Sushilkumar Kalmegh. Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news, feb 2015. International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 2.

[48] Ron Kohavi. A study of crossvalidation and bootstrap for accuracy estimation and model selection, 1995. International Joint Conference on Articial Intelligence (IJCAI).

[49] Ali Farman, El-Sappagh Shaker, Islam S.M. Riazul, Kwak Daehan, Ali Amjad, Imran Muhammad, and Kwak K.S. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion, 2020. Information Fusion.

[50] Gopi Battineni, Getu Gamo Sagaro, Chintalapudi Nalini, Francesco Amenta, and Seyed Khosrow Tayebati. Comparative machine-learning approach : A follow-up study on type 2 diabetes predictions by cross-validation methods, dec 2019.

[51] N. Harun, S. S. Dlay, and W. L. Woo. Performance of keystroke biometrics authentication system using multilayer perceptron neural network (mlp nn), 2010. in 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010), pp. 711-714.

[52] Indrastanti R. Widiasari, Lukito Edi Nugroho, and Widyawan. Deep learning multilayer perceptron (mlp) for flood prediction model using wireless sensor network based hydrology time series data mining, nov 2017. International Conference on Innovative and Creative Information Technology (ICITech). IEEE, 17634194.

[53] Djeffal Abdelhamid. Utilisation des méthodes support vector machine (svm) dans l'analyse des bases de données, 2012. Thèse de Doctorat.

[54] Zaiz Faouzi. Les supports vecteurs machines (svm) pour la reconnaissance des caractères manuscrits arabes, jul 2010. Mémoire de Magister.

[55] Andrew I. Schein and Lyle H. Ungar. Active learning for logistic regression : an evaluation, aug 2007. Mach Learn (2007) 68 : 235–265.

[56] Eric Bauer and Ron Kohavi. An empirical comparison of voting classi cation algorithms : Bagging, boosting, and variants, 1998. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

[57] Wenbing Chang, Yinglai Liu, Yiyong Xiao, Xinglong Yuan, Xingxing Xu, Siyue Zhang, and Shenghan Zhou. A machine-learning-based prediction method for hypertension outcomes based on medical data, nov 2019. School of Reliability and Systems Engineering, Beihang University, Beijing 100191 China.

[58] Mahesh Pal. Random forest classifier for remote sensing classification, 2005. International Journal of Remote Sensing, 26 :1, 217-222.

[59] Katherine R. Gray, Paul Aljabar, Rolf A. Heckemann, Alexander Hammers, and Daniel Rueckert. Random forest-based similarity measures for multi-modal classification of alzheimer's disease, jan 2013. Neuroimage, 65C : 167–175.

[60] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. Gene selection and classification of microarray data using random forest, jan 2006. BMC Bioinformatics, 7 :3.

[61] Santos Frédéric. Arbres de décision, mar 2015. CNRS, UMR 5199 PACEA.

[62] Taleb Zouggar S. Contribution à l'apprentissage automatique par automate d'arbre et mesure de sélection, dec 2014. Thèse de Doctorat.

[63] Tina R. Patil and S. S. Sherekar. Performance analysis of naive bayes and j48 classification algorithm for data classification, apr 2013. Sant Gadgebaba Amravati University, Amravati.

[64] Hua Tang, Yueting Xu, Aiju Lin, Ali Asghar Heidari, Mingjing Wang, Huiling Chen, Yungang Luo, and Chengye Li. Predicting green consumption behaviors of students using efficient firefly grey wolf-assisted k-nearest neighbor classifiers, feb 2020.

[65] Tianqi Chen and Carlos Guestrin. Xgboost : A scalable tree boosting system, aug 2016. in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA.

[66] Norma Latif Fitriyani, Muhammad Syafrudin, Ganjar Alfian, Member IEEE, and Jongtae Rhee. An effective heart disease prediction model for a clinical decision support system (hdpm), jul 2020.

[67] Md Rafiul Hassan, Sadiq Al-Insaif, M. Imtiaz Hossain, and Joarder Kamruzzaman. A machine learning approach for prediction of pregnancy outcome following ivf treatment, sep 2018. Neural Computing and Applications.

[68] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems, jan 2004. ACM Transactions on Information Systems, Vol. 22, No. 1, Pages 5–53.

[69] Cai-Nicolas Ziegler. Towards decentralized recommender systems, jun 2005. Albert-Ludwigs-Universität Freiburg - Fakultät für Angewandte Wissenschaften, Institut für Informatik.

[70] Jonathan Lee Herlocker. Understanding and improving automated collaborative filtering systems, sep 2000. University of Minnesota.

[71] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification, may 2005. International World Wide Web Conference, Chiba, Japan.

[72] Gediminas Adomavicius and Alexander Tuzhilin. Towards the next generation of recommender systems : A survey of the state-of-the-art and possible extensions, 2005. Transactions on Knowledge and Data Engineering.

[73] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce, 2000. Proceedings of the 2nd ACM Conference on Electronic Commerce, Minneapolis, Minnesota, United States.

[74] John Canny. Collaborative filtering with privacy, 2002. Proceedings of the 2002 IEEE Symposium on Security and Privacy : IEEE Computer Society.

[75] Bradley N. Miller, Joseph A. Konstan, and John Riedl. Pocketlens : Toward a personal recommender system, jul 2004. ACM Transactions on Information Systems, Vol. 22, No. 3, Pages 437–476.

[76] Zied Zaier, Robert Godin, and Luc Faucher. Recommendation quality evolution based on neighborhood size, dec 2007. Third International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution. AXMEDIS '07 Barcelona, Spain, pp. 33-36.

[77] Zied Zaier, Robert Godin, and Luc Faucher. Recommendation quality evolution based on neighbors discrimination, 2008. International MCETECH Conference on e-Technologies Montreal, pp. 148-153.

[78] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering, jan 2005. SIAM Data Mining (SDM'05) Newport Beach, California, USA, pp. 21-23.

[79] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge, 2007. SIGKDD Explor. Newsl., vol. 9, pp. 75-79.

[80] Robert M. Bell and Yehuda Koren. Improved neighborhood-based collaborative filtering, aug 2007. KDDCup'07, San Jose, California, USA.

[81] Robert M. Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights, 2007. Data Mining. ICDM. Seventh IEEE International Conference, pp. 43-52.

[82] Robert M. Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems, aug 2007. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining San Jose, California, USA : ACM.

[83] Robert M. Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the netflix prize, 2007.

[84] Amma N G Bhuvaneswari. An intelligent approach based on principal component analysis and adaptive neuro fuzzy inference system for predicting the risk of cardiovascular diseases, dec 2013. Fifth International Conference on Advanced Computing (ICoAC).

[85] R. Saravana Kumar and G. Tholkappia Arasu. Rough set theory and fuzzy logic based warehousing of heterogeneous clinical databases, 2017. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems Vol. 25, No. 3, 385-408.

[86] Thanh Nguyen, Abbas Khosravi, Douglas Creighton, and Saeid Nahavandi. Classification of healthcare data using genetic fuzzy logic system and wavelets, 2014. Expert Systems with Applications.

[87] G. Thippa Reddy and Neelu Khare. An efficient system for heart disease prediction using hybrid ofbat with rule-based fuzzy logic model, 2017. Journal of Circuits, Systems, and Computers Vol. 26, No. 4, 1750061.

[88] Aanshi Gupta, Shubham Yadav, Shaik Shahid, and Venkanna U. Heartcare : Iot based heart disease prediction system, dec 2019. International Conference on Information Technology (ICIT).

[89] Senthilkumar Mohan, Chandrasegar Thirumalai, and Gautam Srivastava. Effective heart disease prediction using hybrid machine learning techniques, jun 2019. IEEE Access.

[90] Tulasi Krishna Sajja and Hemantha Kumar Kalluri. A deep learning method for prediction of cardiovascular disease using convolutional neural network, dec 2020. Revue D Intelligence Artificielle.

[91] Dhiraj Dahiwade, Gajanan Patle, and Ektaa Meshram. Designing disease prediction model using machine learning approach, mar 2019. Computer Science, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC).

[92] Halima El Hamdaoui, Saïd Boujraf, Nour El Houda Chaoui, and Mustapha Maaroufi. A clinical support system for prediction of heart disease using machine learning techniques, sep 2020. 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP).

[93] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning : Data mining, inference, and prediction, 2008. Springer.

[94] R. O. Duda, P. E. Hart, D. G. Stork, John Wiley, and Sons. Pattern classification, 2001.

[95] Daniel Berrar. Cross-validation, jan 2018. Data Science Laboratory, Tokyo Institute of Technology.

[96] Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization, 2015. Published as a conference paper at ICLR.

[97] Daphne Koller and Mehran Sahami. Toward optimal feature selection, 1996.

[98] Lei Yu and Huan Liu. Feature selection for high-dimensional data : A fast correlation-based filter solution, 2003. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC.

[99] Eric P. Xing, Michael I. Jordan, and Richard M. Karp. Feature selection for high-dimensional genomic microarray data, jun 2001. Proceedings of the Eighteenth International Conference on Machine Learning (ICML).