الجمهورية الجزائرية الديمقراطية الشعبية

République algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة عين تموشنت بلحاج بوشعيب

Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie

قسم الرياضيات والإعلام الآلي

Département des Mathématiques et Informatique

Projet de fin d'études

Pour l'obtention du diplôme de Master

Spécialité : Réseaux et Ingénierie des Données

Thème

Etude comparative des algorithmes dédiés à la classification

Présenté Par :

- M BELARBI Boucif
- M TEMMOUN Alaa Eddine

Devant le jury composé de :

Dr BENARIBI.F	MAA	UAT.B.B (Ain Temouchent)	Président
Dr MERAD.D	MAA	UAT.B.B (Ain Temouchent)	Examinateur
Dr BOUAFIA.Z	MAA	UAT.B.B (Ain Temouchent)	Encadrant

Année Universitaire 2021/2022

Dédicace

Je dédie ce travail, en signe de reconnaissance et de gratitude, à ma grand-mère et mes chers parents qui ont sacrifié leur vie pour ma réussite et m'ont éclairé le chemin par leurs conseils judicieux. J'espère qu'un jour, je pourrais leur rendre un peu de ce qu'ils ont fait pour moi. Que Dieu leur prête bonheur et longue vie.

A ma tante Nassira, mes chers frère et sœurs.

Ce travail est également dédié à tous mes enseignants et à tous mes amis.

Boucif.

Dédicace

Tout d'abord, je tiens à remercier DIEU de m'avoir donné la force et le courage de mener à bien ce modeste travail. Je tiens à dédier cet humble travail à ma mère Nadia et mon père Houari, dieu ait son âme, à ma grand-mère Merieme, à mes frères SIDAHMED et NIDAL, à mon binôme Boucif et tous ceux qui m'aiment et que j 'aime.

Alaa.

Remerciements

Avant tout, nous remercions Allah le tout puissant de nous avoir donné la force et la volonté d'accomplir ce travail

C'est avec une immense reconnaissance que nous adressons nos remerciements les plus sincères à notre encadrant, Dr ZOUHEIR BOUAFIA d'avoir accepté de diriger ce travail, ses aides précieuses, son encouragement et ses conseils constructifs ont été d'une grande utilité pour bien mener ce travail.

Nous tenons à remercier Dr F. BENARIBI Maître Assistant Classe "A" à Université Ain Temouchent Belhadj Bouchaib. D'avoir accepté de présider le jury et d'évaluer ce projet de fin d'études.

Nous remercions Dr D. MERADE Maître Assistant Classe "A" à Université Ain Temouchent Belhadj Bouchaib. D'avoir accepté d'examiner notre travail.

Nous remercions « TOUS » les Messieurs et dames, nos professeurs qui nous ont enseigné durant deux ans de formation master en informatique.

Finalement, nous adressons nos plus sincères remerciements à nos familles et nos proches qui nous ont soutenu et encouragé, et ce durant toute la durée de ce projet.

Résumé

La classification est la technique de data mining la plus populaire, elle sert à catégoriser ou à classer des informations issues d'un grand jeu de données dans le but d'établir des prédictions. Il existe de nombreux algorithmes utilisés pour la classification. Mais la plupart des algorithmes existants sont instables en termes de précision et basés sur le contenu des données.

Ce projet vise à présenter une étude comparative entre les algorithmes de classification, et de proposer une approche ensembliste afin d'essayer de résoudre le problème de précision. Les expérimentations et les évaluations ont été réalisées sur des bases de données médicales et à l'aide de la bibliothèque WEKA. Les résultats obtenus sont très satisfaisants par rapport aux algorithmes de classification.

Mots clés : Fouille de données, techniques de classification, méthodes ensemblistes, Weka.

Abstract

Classification is the most popular data mining technique; it is used to categorize or classify information from a large data set in order to make predictions. There are many algorithms used for classification. But most of the existing algorithms are unstable in terms of accuracy and based on data content.

This project aims to present a comparative study between classification algorithms, and to propose an ensemble approach in order to try to solve the problem of precision. The experiments and evaluations were carried out on medical databases and using the WEKA library. The results obtained are very satisfactory compared to the classification algorithms.

Keywords : Data mining, classification techniques, ensembles methods, Weka.

ملخص

التصنيف هو أكثر تقنيات التنقيب عن البيانات شيوعًا، ويستخدم لتصنيف المعلومات المأخوذة من مجموعة بيانات كبيرة من أجل عمل تنبؤات. هناك العديد من الخوارزميات المستخدمة في التصنيف. لكن معظم الخوارزميات الموجودة غير مستقرة من حيث الدقة وتعتمد على محتوى البيانات.

يهدف هذا المشروع إلى تقديم دراسة مقارنة بين خوارزميات التصنيف، واقتراح نهج جماعي لمحاولة حل مشكلة الدقة. أجريت التجارب والتقييمات على قواعد البيانات الطبية وباستخدام مكتبة WEKA, النتائج التي تم الحصول عليها مرضية للغاية مقارنة بخوارزميات التصنيف.

الكلمات المفتاحية :التنقيب عن البيانات، تقنيات التصنيف، طرق التجميع، Weka.

Table des matières

Liste des Figures

Liste des algorithmes :

Liste des tableaux :

Liste d’abréviations :

ECD : Extraction de connaissances à partir de données.

KNN: k plus porches voisin (k-nearst-neighbors).

ML: machine Learning.

SVM: machine a vecteurs de support

MLLIB: bibliothèque de machine Learning (machine learning library)

JAVAML: java machine Learning

IBL : apprentissage basé sur l’instance (instance based learning)

TdK : type de kernel

TAUX APP : taux d’apprentissage.

NBCC : nombre de couches cachées.

CL NV2 : classifieur de niveau 2.

NB B : nombre de blocs.

AD : arbre de décisions

NB : naïve Bayes.

RN : réseaux de neurones.

Introduction générale

Nous vivons dans un monde où des grands volumes de données sont collectées quotidiennement, Ces données provenaient de la société commerciale, de la science, de la médecine et de presque tous les autres aspects de la vie quotidienne. Ces données sont importantes pour développer le domaine dont elles sont issues, Mais les plus importantes, ce sont les informations cachées dans ces données, nous avons donc besoin des méthodes puissantes pour extraire ces informations. Cette nécessité a conduit à la naissance du data mining.

Data mining a plusieurs techniques pour extraire les informations à partir des grandes bases de données, la technique la plus célèbre est la classification. Cette technique consiste à appliquer des algorithmes de classification sur des grandes bases de données pour construire un modèle qui aide à classer des nouvelles données dans le futur. Mais les algorithmes de classification ne donnent pas toujours des bons résultats, en particulier dans la précision.

L'objectif de ce projet est de présenter une étude comparative entre les algorithmes de classification et d'implémenter une méthode ensembliste pour augmenter la précision des résultats sur des données médicales.

Notre mémoire est organisé en trois chapitres :

Le premier chapitre est consacré pour une introduction au data mining tel que la définition de data mining, le processus général d'extraction des connaissances, les techniques de data mining et la différence entre les deux termes populaires data mining et machine learning.

Dans le deuxième chapitre nous allons voir quelques algorithmes de classification afin de présenter une étude comparative entre ces derniers sur plusieurs critères de performances. Nous allons voir aussi quelques algorithmes ensemblistes populaires.

Dans le troisième chapitre, nous allons implémenter la méthode ensembliste (stacking) pour essayer de résoudre le problème de précision des résultats trouvé par les algorithmes de classification dans certains types de données. Nous allons expérimenter la méthode implémentée sur des bases de données médicales afin de comparer les résultats de cette méthode ensembliste avec les résultats des algorithmes de classification sur plusieurs mesures de performance.

Nous terminons ce rapport par une conclusion générale dans laquelle nous rappelons la problématique, la méthode implémentée, et les principaux résultats qu'on a eus ainsi que les améliorations qu'on peut effectuer sur notre méthode.

Chapitre 1 : Introduction au Data Mining

1.1 Introduction

Le data mining n'est pas né à l'ère numérique. Ce concept est vieux d'un siècle mais il s'est révélé dans les années 1980. De nos jours, l'extraction de données est utilisée dans de nombreux les domaines d'activité tels que la recherche, la commercialisation, la mise au point de produits, la santé ou l'éducation.

L'objectif de ce premier chapitre est de présenter les principales notions de data mining : sa définition, le processus d'extraction des connaissances, les techniques de data mining supervisées et non supervisées ainsi que la différence entre data mining et machine Learning.

1.2 Définition de data mining

Selon Charu C. Aggarwal data mining est l'étude de la collecte, du nettoyage, du traitement, de l'analyse et de l'obtention d'informations utiles à partir de données. Une grande variation existe en termes de domaines problématiques, d'applications, de formulations et de représentations de données rencontrées dans les applications réelles. Par conséquent, «data mining» est un terme générique utilisé pour décrire ces différents aspects du traitement des données.[21] [1].

1.3 Notions de base

Attribut : dans data mining, un attribut est une caractéristique qui est mesurée pour chaque observation et peut varier d'une observation à l'autre. Il peut être mesuré en valeurs continues ou en valeurs catégorielles. [2].

Enregistrement : Un enregistrement (en anglais record) est un élément d'une base de données. L'enregistrement contient habituellement plusieurs informations (entrées) qui se rapportent au même objet.

Discrétisation : la discrétisation est la transposition d'un état continu en un équivalent discret.

Attribut classe : l'attribut de classe est l'attribut discret dont vous souhaitez prédire la valeur en fonction des valeurs d'autres attributs.

Un point de données : est une unité d'information discrète. Dans un contexte statistique ou analytique, un point de données est généralement dérivé d'une mesure ou d'une recherche et peut être représenté numériquement et/ou graphiquement.

Variable Indépendante : Une variable indépendante est la variable qui est modifiée ou contrôlée dans le cadre d'une expérience scientifique afin de tester les effets sur la variable dépendante.

Variable dépendante : Une variable dépendante est la variable testée et mesurée dans le cadre d'une expérience scientifique.

Centroïde : Un centroïde représente le centre d'un groupe de données.

Big data : ensembles de données extrêmement volumineux qui peuvent être analysés par calcul pour révéler des modèles, des tendances et des associations.

1.4 Extraction des connaissances à partir de données (ECD)

Il faut noter que le data mining n'est qu'une étape d'un processus plus global appelé extraction de connaissance à partir de donne (ECD). Il définit le vaste processus de découverte de connaissances dans les données et met l'accent sur les applications de haut niveau des techniques d'extraction des connaissances définies. C'est un domaine d'intérêt pour les chercheurs dans plusieurs domaines. L'objectif global du processus ECD est de convertir les données brutes en informations utiles. [3][4].

1.5 PROCESSUS ECD

Selon le schéma illustré dans la figure 1.1 le processus d'ECD comporte essentiellement cinq étapes comme explicitées ci-dessous :

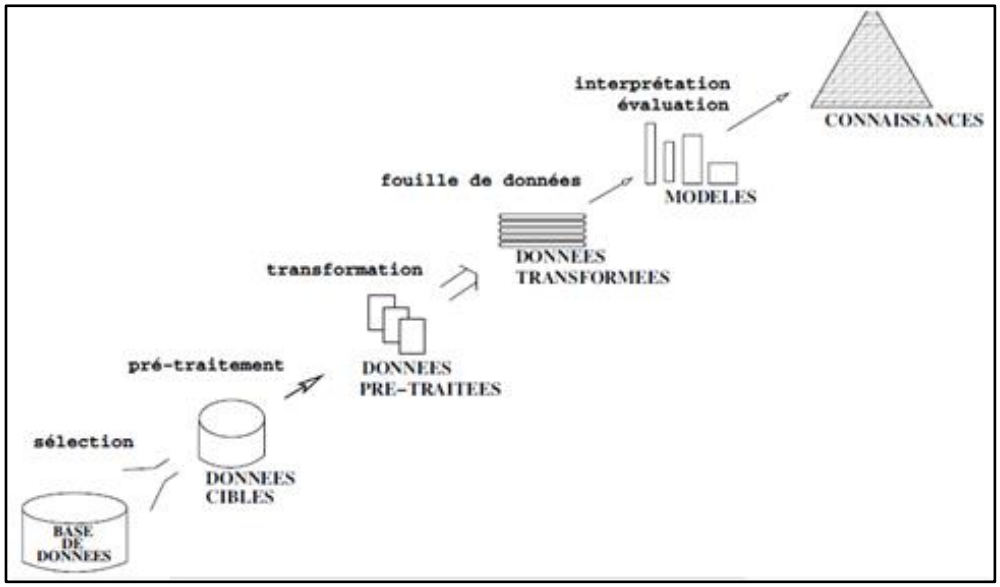

Figure 1. 1: le processus de découverte des connaissances dans les bases de données (ECD).[61]

1.5.1 Sélection des données

Les données qui seront utilisées pour la découverte de connaissances devraient être déterminées selon l'objectif visé. Cela comprend la recherche des données disponibles, l'obtention des données supplémentaires nécessaires, puis l'intégration de toutes les données pour la découverte des connaissances dans un seul ensemble de données.

1.5.2 Prétraitement

Les données peuvent contenir plusieurs types d'anomalies des données à cause des erreurs de frappe ou à causes des erreurs dues au système lui-même ce qui nécessite un nettoyage. C'est-à-dire, l'élimination de la répétition, ainsi que le traitement des valeurs manquantes ou erronées et même supprimer les enregistrements qui contient beaucoup de valeurs manquantes, cette étape est importante pour la suite de processus d'extraction de connaissances.

1.5.3 Transformation des données.

Dans certains cas les données exigent des transformations, ce dernier se fait par attribut, c'est-à-dire toutes les valeurs d'un attribut doivent être transformées en un format unique. Certaines tâches de data mining donne des bons résultats avec des attributs qui contiennent des valeurs discrètes. Donc, il est mieux d'appliquer la méthode de discrétisation sur les attributs continus, il y a d'autres transformations selon les besoins comme le changement de type (ex : date de naissance à âge) et le regroupement.

1.5.4 Data mining

Data mining est le cœur du processus ECD, elle consiste l'extraction des connaissances, dans cette étape une méthode de data mining doit être appliquée sur les données prétraitées, il faut bien choisir la méthode de l'extraction selon l'objectif de ce processus, il est possible de combiner plusieurs méthodes pour avoir plus de connaissances.

1.5.5 Evaluation et interpretation

Les modèles extraits par les méthodes de data mining ne peuvent être utilisés qu'après une étape d'évaluation, il existe plusieurs méthodes d'évaluation des modèles. Ces méthodes peuvent aider à corriger les modèles. [4]

1.6 Techniques de data mining

1.6.1 Techniques Supervisées

Data mining supervisé fait référence à des algorithmes d'apprentissage qui sont utilisés dans la classification et dans la prédiction. L'algorithme supervisé apprend à partir des données d'apprentissage qui sont étiquetées et la tâche est contrôlée par l'ingénieur des connaissances et le concepteur du système. Avec des données supervisées, nous devons avoir des entrées connues correspondant à des sorties connues, telles que déterminées par les experts du domaine. Cette tâche de data mining est souvent appelée apprentissage supervisé, car les classes sont déterminées avant l'examen des données. La technique supervisée tente d'identifier les relations entre les variables indépendantes et les variables dépendantes, identifier le degré de corrélation pour chaque ensemble de variables et de construire un modèle montrant le réseau de dépendances. Le modèle est ensuite appliqué aux données dont la variable dépendante est inconnue. [5][6]

1.6.1.1 Modèles de classification

Les modèles de classification sont un sous-ensemble de data mining supervisé. Un modèle de classification lit certaines entrées et génère une sortie qui classe l'entrée dans une certaine catégorie. La classification est la technique de data mining la plus couramment appliquée, qui utilise un ensemble d'enregistrement pour développer un modèle qui peut classer les nouveaux enregistrements. Le processus de classification des données implique apprentissage et le classement. Dans l'apprentissage, les données d'entraînement sont analysées par un algorithme de classification. Puis un nouvel ensemble de données de test utilisées pour estimer la précision de model de classification. Si la précision est acceptable, les modèles peuvent être appliquer sur des nouveaux enregistrements. [7][8][9]

➢ Arbre de décision :

Les algorithmes d'arbre de décision sont les algorithmes les plus couramment utilisés en classification et en estimations. L'arbre de décision fournit une technique de modélisation facilement compréhensible et simplifie également le processus de classification. L'arbre de décision est un mécanisme transparent qui permet aux utilisateurs de suivre facilement une structure arborescente, les arbres de décision fonctionnent en séparant de façon récursive les enregistrements initiaux. Pour chaque groupe, ils sélectionnent automatiquement l'attribut le plus significatif qui donne la meilleure séparation par rapport à l'attribut classe à travers des divisions successives, leur objectif est de produire des sous-segments purs qui aident à la classification. [11] [12]

Les algorithmes d'arbre de décision disponible sont :

- ID3 (IterativeDichotomiser 3)[12]
- C4.5, C5 (successeurs d'ID3)[13]
- CHAID (CHi-squaredAutomatic Interaction Detector) [14]
- CART (Classification And Regression Tree))[13]

➢ K plus proches voisins (KNN) :

C'est un algorithme de data mining qui sert à la classification et la régression, l'algorithme KNN ne construire pas un modèle à partir d'un ensemble de données d'entrainement, pour pouvoir effectuer une classification ou une régression, KNN se base sur l'ensemble de données pour produire un résultat. Pour un enregistrement qui ne fait pas partie de l'ensemble de données initiales et qu'on souhaite classer, l'algorithme va chercher les K points de données les plus proches de notre enregistrement. Ensuite, l'algorithme se base sur les valeurs dépendantes de ces K voisins, pour calculer ou estimer la valeur de la variable de sortie (dépendante) de l'enregistrement qu'on souhaite prédire. [15]

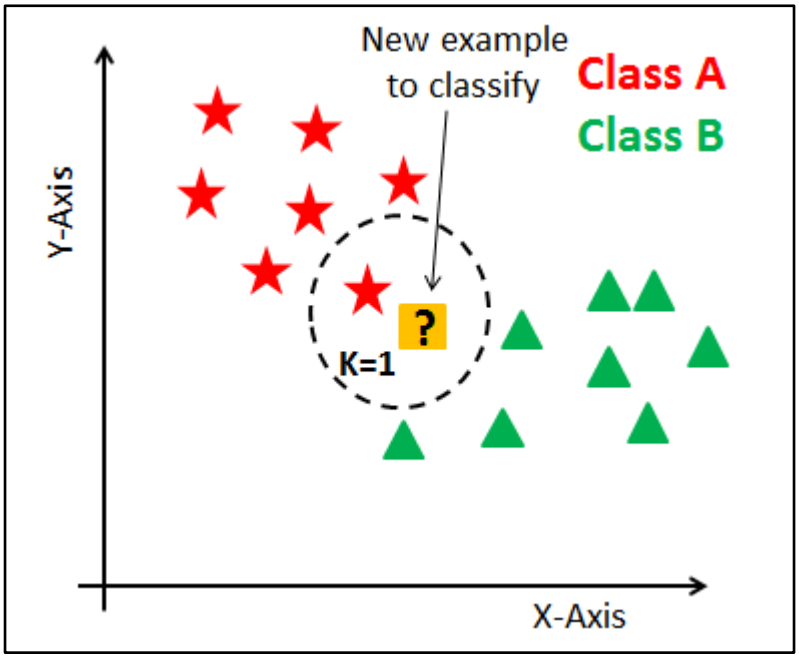

Figure 1. 2: un exemple de classification avec l'algorithme KNN [62]

1.6.1.2 Modèles de régression

La régression est similaire à la classification, sauf que les variables cibles (dépendantes) sont numériques plutôt que catégoriques. Les modèles sont construits à partir d'enregistrements qui fournissent les valeurs des variables dépendantes ainsi que les prédicteurs. Ensuite, pour les nouveaux enregistrements, des estimations des valeurs des variable cibles (dépendantes) sont faites à l'aide de modèle construit. Nous pouvons ensuite appliquer ce modèle sur des nouveaux enregistrements. [10]

- **Régression linéaire :**

La régression linéaire est une technique de régression dans laquelle les valeurs des attributs dans la base de données à une relation linéaire avec les valeurs des attributs classes. La ligne droite dans la figure 1.3 est la ligne de meilleur ajustement. L'objectif principal de la régression linéaire est de considérer les points de données et de tracer la ligne de meilleur ajustement pour ajuster le modèle de la meilleure façon possible. [16]

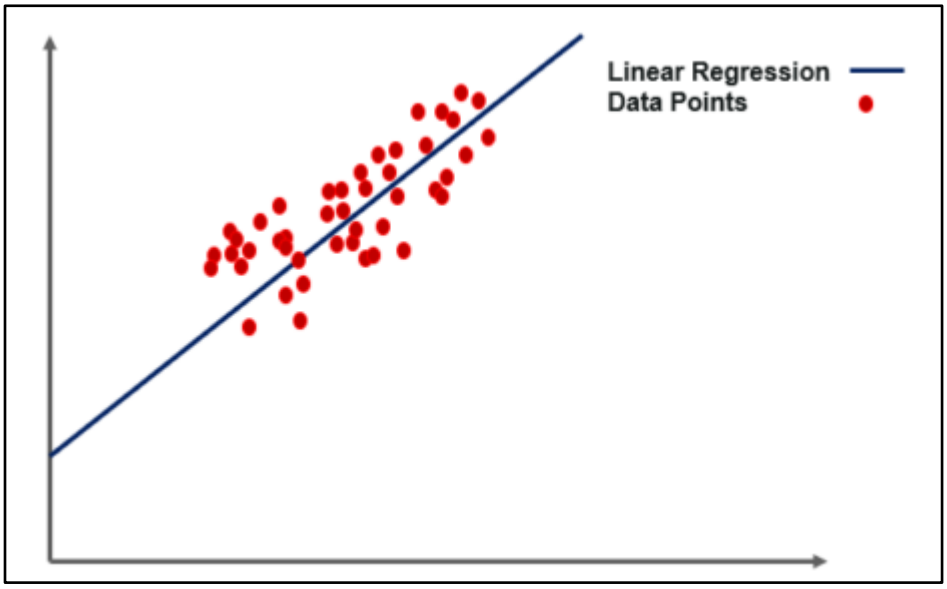

Figure 1. 3: exemple démonstratif d'une régression linaire.[63]

1.6.2 Techniques non Supervisées

Contrairement à la technique supervisée, les techniques de data mining non supervisé sont celles où il n'y a pas des variables dépendantes à prédire ou à classer. Par conséquent, il n'y a pas d'apprentissage à partir des enregistrements où une telle variable dépendante est connue. Il aide à identifier toutes sortes de modèles inconnus dans les données. Le modèle non supervisé est également appelé modèle descriptif, car il cherche des modèles inconnus dans un ensemble de données sans étiquettes prédéterminées et sans supervision humaine. Ce type de technique d'apprentissage est utilisée lorsqu'un objectif spécifique n'est pas disponible ou lorsque l'utilisateur cherche à trouver des relations cachées dans les données [5][6].

1.6.2.1 Modèles de regroupement (clustring) :

Le clustering est une méthode de regroupement des enregistrements en clusters de sorte que les enregistrements présentant le plus de similitudes restent dans un groupe et présentent moins ou pas de similitudes avec les enregistrements d'un autre groupe. L'analyse de cluster trouve les points communs entre les enregistrements et les catégorise selon la présence et l'absence de ces points communs.

- **Algorithme k-means :**

K-Means est un algorithme d'apprentissage non supervisé, c'est la méthode de regroupement la plus connue et la plus utilisée qui regroupe l'ensemble de données non étiquetées en différents

groupes. K définit le nombre de groupes prédéfinis qui doivent être créés dans le processus, comme si K=2, il y aura deux clusters, et pour K=3, il y aura trois clusters ou groupes, et ainsi de suite. C'est un moyen pratique de découvrir les catégories de groupes dans l'ensemble de données non étiqueté par lui-même. Il s'agit d'un algorithme basé sur les centroïdes, où chaque groupe est associé à un centroïde. L'objectif principal de cet algorithme est de minimiser la somme des distances entre les points de données et leurs clusters correspondants, l'algorithme de clustering k-means effectue principalement deux tâches :[17]

- Déterminer les k centroides.
- Affectations des points de données aux centroides selon la distance la plus proche.

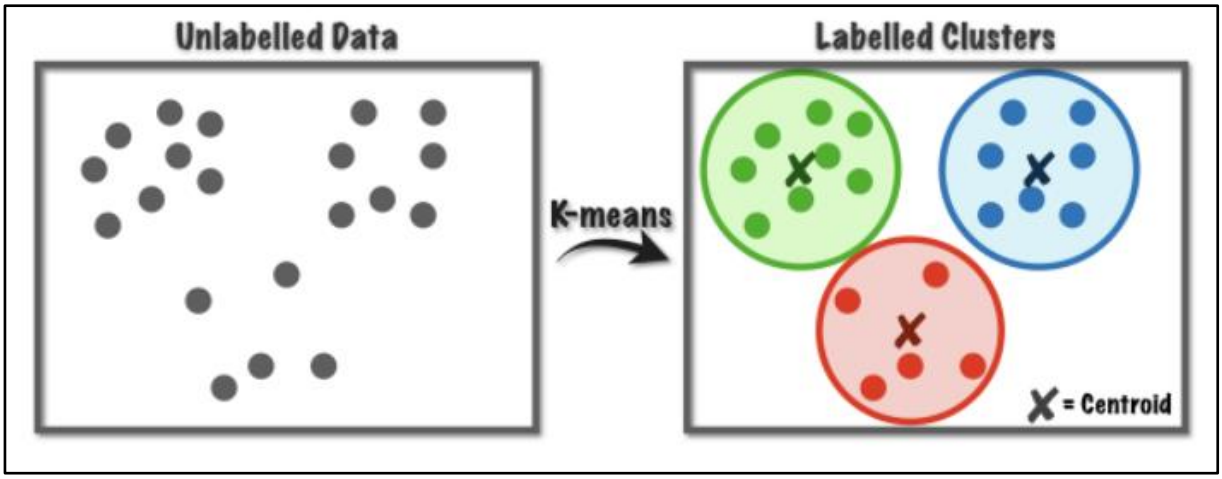

Figure 1. 4:exemple démonstratif d'un ensemble de données avant et après l'application de la technique k-means.[64].

1.6.2.2 Modèles d'association :

Un modèle d'association est une méthode d'apprentissage non supervisée qui est utilisée pour trouver les relations entre les attributs dans des bases de données volumineuses. Il détermine l'ensemble d'éléments qui se produisent ensemble dans l'ensemble de données. Le modèle d'association rend la stratégie de marketing plus efficace. Par exemple, les personnes qui achètent un article X (supposons un pain.) ont également tendance à acheter un article Y (beurre/confiture). Un exemple typique de règle d'association est l'analyse du panier de consommation.

- **Algorithme Apriori :**

L'algorithme apriori utilise des ensembles d'éléments fréquents pour générer des règles d'association, et il est conçu pour fonctionner sur les bases de données qui contiennent des transactions. À l'aide de ces règles d'association, il détermine la force ou la faiblesse de la connexion de deux objets. Cet algorithme utilise une recherche étendue pour calculer efficacement les règles d'associations. C'est le processus itératif pour trouver les ensembles d'éléments fréquents à partir d'un grand ensemble de données. [18].

1.7 Data mining vs machine Learning

1.7.1 Définition de machine Learning

Kevin P. Murphy a défini la machine learning comme un ensemble de méthodes capables de détecter automatiquement des modèles dans les données, puis d'utiliser les modèles non couverts pour prédire les données futures ou pour effectuer d'autres types de prise de décision dans des conditions d'incertitude [73].

1.7.2 Les similitudes entre data mining et machine Learning

Data mining et machine learning relèvent tous deux de la science des données, ce qui est logique puisqu'ils utilisent tous les deux des données. Les deux processus sont utilisés pour résoudre des problèmes complexes, par conséquent, de nombreuses personnes utilisent les deux termes de manière interchangeable. Ce n'est pas si surprenant, étant donné que machine learning est parfois utilisé comme un moyen de mener une tâche de data mining utile. Alors que les données recueillies à partir de data mining peuvent être utilisées pour enseigner aux machines, les frontières entre les deux concepts deviennent un peu floues. De plus, les deux processus utilisent les mêmes algorithmes critiques pour découvrir les modèles de données. Bien que leurs résultats souhaités différents finalement [19].

1.7.3 Différences entre Data mining et Machine Learning

Même si data mining et machine learning se intersectés, ils ont une part équitable de différences quant à la façon dont ils sont utilisés. Voici quelques différences principales entre les deux :

1.7.3.1 Leur Age :

Data mining est antérieure de deux décennies à la machine learning, data mining étant initialement appelé extraction de connaissances dans les bases de données (ECD). [19] [20].

1.7.3.2 Leur but :

Data mining est conçue pour extraire les règles de grandes quantités de données, tandis que machine learning enseigne à un ordinateur comment apprendre et comprendre les paramètres donnés. Ou pour le dire autrement, data mining est simplement une méthode de recherche pour déterminer un résultat particulier basé sur le total des données recueillies. De l'autre côté, nous avons machine learning, qui entraîne un système à effectuer des tâches complexes et utilise les données et l'expérience récoltées pour devenir plus intelligent. [19] [20]

1.7.3.3 Utilisation des données :

La principale différence entre data mining et machine learning réside dans la manière dont chacun utilise les données et les applique à diverses applications. Alors que le data mining s'appuie sur des vastes référentiels de big data à partir desquels il extrait des modèles significatifs, le machine learning fonctionne principalement avec des algorithmes plutôt qu'avec des données brutes. Data mining peut également être utilisée pour parcourir des sites web, des profils de médias sociaux, et même des actifs numériques afin d'obtenir des informations sur les prospects potentiels d'une marque ou d'une entreprise. Bien que le machine learning intègre les principes du data mining, il cherche à établir des corrélations automatiques pour en tirer des leçons et appliquer les résultats à de nouveaux algorithmes de machine learning. Étant donné que les algorithmes sont programmés pour apprendre de l'expérience, ils s'améliorent continuellement, fournissant ainsi des résultats plus précis au fil du temps. [19] [20]

1.8 Types de données

Aujourd'hui, presque tous les systèmes automatisés génèrent des formes de données à des fins de diagnostic ou d'analyse. Cela a entraîné un flot de données, qui a atteint les limites du pétaoctet ou de l'exaoctet. [21] Il est Presque impossible de l'analyser sans utiliser des méthodes telles que data mining. En principe, data mining n'est pas spécifique à un type de données. Data mining devrait être applicable à tous types de données que ce soit structurés ou non-structurés. [22]

Voici quelques ressources de données auxquelles data mining peut être appliquée :

1. Bases de données relationnelles.
2. Bases de données transactionnelles.
3. Bases de données multimédia.
4. World Wide Web(WWW).

1.9 Conclusion

Ce chapitre a été consacré à la présentation du data mining, le domaine qui représente le contexte de travail de notre projet de fin d'études. Nous avons présenté donc les notions fondamentales et nécessaires à la compréhension de ce domaine, en particulier le processus ECD et les techniques de data mining. Le deuxième chapitre de notre mémoire sera consacré à la classification.

Chapitre 2 : Etude comparative entre les algorithmes de classification

2.1 Introduction

Depuis l'aube de l'histoire, l'homme a pratiqué la classification dans sa vie quotidienne, en essayant de répondre aux problèmes et aux questions sur la classe des choses, c'est-à-dire en attribuant un objet à sa classe d'origine, mais avec le développement de la technologie, l'homme est confronté à des problèmes plus complexes. Problèmes qui nécessitent des classifications intelligentes pour faire des prédictions et extraire des informations à partir d'un ensemble de données brutes.

Dans ce chapitre, nous allons présenter une étude comparative entre certains algorithmes de classification tels que l'arbre de décision (AD), naïve bayes (NB), K-plus proches voisins (KNN), le réseau de neurones (RN) et la machine à vecteurs de support (SVM). Cette étude mettre en évidence les avantages et les inconvénients de chaque algorithme de classification, et nous présenterons également quelques méthodes ensemblistes populaires comme le bagging, le boosting et le stacking.

2.2 Notions de base

Sur ajustement : (overfitting en anglais) est l'un des pires ennemis des scientistes des données. Il s'agit d'un problème fréquemment rencontré en machine learning. Il survient lorsque le modèle essaie de trop s'adapter aux données d'apprentissage.

Produit scalaire : En mathématiques, et plus précisément en algèbre et en géométrie vectorielle, le produit scalaire est une opération algébrique s'ajoutant aux lois s'appliquant aux vecteurs. C'est une forme bilinéaire, symétrique, définie positive.

Hyperplans : Un hyperplan est une frontière de décision qui différencie les deux classes dans SVM. Un point de données tombant de chaque côté de l'hyperplan peut être attribué à différentes classes. La dimension de l'hyperplan dépend du nombre d'entités en entrée dans le jeu de données.

Les apprenants paresseux : (lazy learner en anglais) stockent simplement les données d'apprentissage et attendent qu'une donnée de test apparaisse. Lorsque c'est le cas, la classification est effectuée sur la base des données d'apprentissage stockée. Par rapport aux autres apprenants, les apprenants paresseux ont moins de temps d'apprentissage mais plus de temps pour prédire.

2.3 Les algorithmes de classification

2.3.1 Arbre de décision

L'arbre de décision classe les données à l'aide d'algorithmes de structure arborescente [12]. L'objectif principal des arbres de décisions est d'exposer les informations structurelles contenues dans les données. La méthode de l'arbre de décision est une technique d'apprentissage automatique supervisée qui construit un arbre de décision à partir d'un ensemble de données étiquetés par classe au cours du processus d'apprentissage automatique [24].

L'algorithme d'arbre de décision commence par les échantillons d'apprentissage et leurs étiquettes de classe associées. Cet ensemble d'apprentissage est partitionnée de manière récursive en fonction de la valeur de la caractéristique en sous-ensemble afin que les données de chacun des sous-ensembles soient plus pures que les données de l'ensemble parent. Chaque nœud interne dans un arbre de décision représente un test sur un attribut (fonctionnalité), chaque branche représente un résultat du test et chaque nœud feuille représente l'étiquette de classe. En tant qu'arbre de décision est utiliser pour identifier l'étiquette de classe d'enregistrement inconnu, en traçant le chemin de la racine au nœud feuille, qui contient l'étiquette de classe pour cet enregistrement [24] [25].

Le nœud racine de l'arbre est la caractéristique qui divise le mieux les données d'apprentissage. Il existe plusieurs mesures pour trouver la caractéristique qui divise mieux les données d'apprentissage, comme le gain d'information, le rapport de gain, l'indice de gini, [24].

La complexité de l'arbre de décision augmente avec la hauteur de l'arbre. Par conséquent, les mesures qui ont tendance à produire un arbre avec plusieurs voies et qui favorisent des fractionnements plus équilibrés peuvent être préférées, et dépond de l'ensemble de données.

L'algorithme d'induction de l'arbre de décision de base adopte des stratégies de division et de conquête sans retour en arrière, gourmandes, descendantes et récursives. L'algorithme se résume comme suit :

Algorithme : Generate_decision_tree

Entrée :

a. Partition de données, D, un ensemble de tuples d'apprentissage et leurs étiquettes de classe associées

b. liste_attribut, l'ensemble des attributs candidats

c. méthode sélection d'attribut, une procédure pour déterminer le critère de fractionnement qui partitionne le mieux les tuples de données en classes individuelles. Ce critère se compose d'un attribut de fractionnement et, soit d'un point de partage, soit d'un sous-ensemble de fractionnement

Sortie : un arbre de décision

Méthode :

1. créer un nœud N

2. si les tuples dans D sont tous de la même classe, C, alors

3. retourner N comme un nœud feuille étiqueté avec la classe C

4. si la liste_attribut est vide, alors

5. retourner N comme un nœud feuille étiqueté avec la classe majoritaire dans D

6. appliquez la méthode sélection d'attribut (D, liste_attribut) pour trouver le meilleur critère_de_partage

7. étiquetez le nœud N avec critère_de_partage

8. si l'attribut_partage est à valeur discrète et que les fractionnements multiples sont autorisés, alors

9. liste_attribut ← liste_attribut – attribut de fractionnement

10. pour chaque résultat j du critère_splittage

    i. soit Dj l'ensemble des tuples de données dans D satisfaisant le résultat j

    ii. si Di est vide alors attachez une feuille étiquetée avec la classe majoritaire dans D au nœud N

    iii. sinon, attachez le nœud retourné par Generate_decision_tree (Dj, liste_attribut) au nœud N

11. retourner N

Algorithme 2.1 : algorithme de l'arbre de décision. [70]

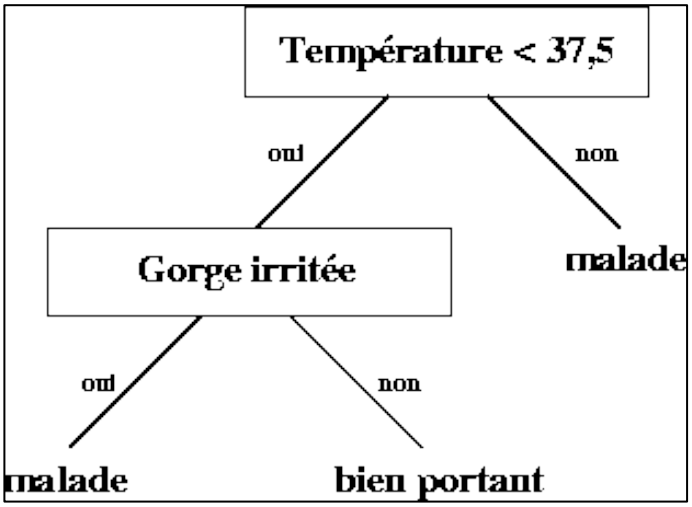

Figure 2. 1 : exemple d'arbre de décision répondant à "Le patient est-il malade ?".[67]

2.3.1.1 Avantages

- Les arbres de décision sont très simples et rapides.
- Il ne nécessite aucune connaissance du domaine ou paramétrage et il est capable de gérer des données de grande dimension.
- La représentation est facile à comprendre.
- Avoir une bonne précision (peut dépendre des données disponibles).
- Il prend en charge l'apprentissage progressif.
- Les arbres de décision sont invariables car ils utilisent une seule caractéristique à chaque nœud interne[28]

2.3.1.2 Inconvénients

- Il a un long temps d'apprentissage, car il nécessite un passage sur les enregistrements d'apprentissage en D pour chaque niveau d'arbre.
- Manque de mémoire disponible, lorsqu'il s'agit de grandes bases de données.
- La division de l'espace des instances est orthogonale à l'axe d'une variable et parallèle à tous les autres axes. Les régions résultantes après partitionnement sont toutes des hypers rectangles.
- La plupart des algorithmes d'arbre de décision ne peuvent pas bien fonctionner avec des problèmes qui nécessitent un partitionnement diagonal.
- Les arbres de décision peuvent être une représentation beaucoup plus complexe pour certains concepts en raison du problème de réplication.
- Les ordres d'attributs dans les nœuds d'arbre ont un effet négatif sur les performances. [28]

2.3.2 Naïve Bayes

Le classifieur Naïve Bayes est un classifieur bayésien statistique simple [29]. Il est appelé naïve car il suppose que toutes les variables contribuent à la classification et sont mutuellement corrélées.

Cette hypothèse est appelée indépendance conditionnelle de classe [30]. Il s'agit d'une hypothèse irréaliste pour la plupart des ensembles de données, mais elle conduit à un cadre de prédiction simple qui donne des résultats étonnamment bons dans de nombreux cas pratiques. Le classifieur naïve bayes est basé sur le théorème de bayes. Le théorème de bayes est

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)}$$

Où,

H- une hypothèse, telle que le tuple de données X appartient à la classe spécifiée C

X – une preuve, décrit par mesure sur un ensemble d'attributs

$P(H|X)$– la probabilité a posteriori que l'hypothèse H tient compte tenu de la preuve X

P(H) – probabilité a priori de H, indépendante de X

$P(X|H)$– la probabilité a posteriori de X conditionnée sur H.

2.3.2.1 Avantages

- Il nécessite un temps de calcul court pour l'apprentissage et très facile à construire.
- Le modèle a la forme d'un produit, qui peut être facilement converti en une somme grâce à l'utilisation de logarithmes - avec des avantages de calcul conséquents significatifs.
- Ne nécessitant aucun schéma d'estimation de paramètres itératifs compliqué, il peut donc être appliqué à un grand ensemble de données.
- Interprétation facile de la représentation des connaissances
- Peut ne pas être le meilleur classifieur dans une application particulière, mais il fonctionne bien et robuste. [28]

2.3.2.2 Inconvénients

- Théoriquement, le classifieur naïve bayes à un taux d'erreur minimal par rapport à un autre classifieur, mais pratiquement ce n'est pas toujours vrai, en raison de l'hypothèse d'indépendance conditionnelle de classe et du manque de données de probabilité disponibles.
- Moins précis que les autres classifieurs. [28]

2.3.3 K plus proches voisins

K plus proches voisins est une méthode d'apprentissage non paramétrique basée sur l'ensemble d'enregistrements. Les classifieurs basés sur les enregistrements sont également appelés Les apprenants paresseux car ils stockent tous les échantillons d'apprentissage et ne créent pas de modèle tant qu'un nouvel échantillon non étiqueté n'a pas besoin d'être classifié. Les algorithmes d'apprentissage paresseux nécessitent moins de temps de calcul pendant la phase d'apprentissage que les algorithmes d'apprentissage (tels que les arbres de décision, les réseaux de neurones et les réseaux de bayes), mais plus de temps de calcul pendant le processus de classification [24] [31] [25].

L'algorithme de k plus proches voisins est l'un des plus simples de tous les algorithmes d'apprentissage automatique. Il est basé sur le principe que les échantillons qui sont similaires sont situés à proximité [32]. Étant donné un échantillon non étiqueté, le classifieur K-plus proche voisin recherche dans l'espace de modèle les k-objets qui en sont les plus proches et attribue la classe en identifiant l'étiquette de classe la plus fréquente. Si la valeur de k = 1, attribuez la classe de l'enregistrement d'apprentissage qui est la plus proche de l'enregistrement inconnu dans l'espace des enregistrements d'apprentissages. L'algorithme KNN est résumé comme suit :

Algorithme KNN ;

Entrée : Instances de test.

1. Pour chaque instance de test {

2. Étant donné une instance de test, trouver le k voisin le plus proche de l'ensemble d'apprentissage selon une métrique de distance.

3. L'étiquette de classe la plus fréquente du k plus proche voisin est l'étiquette de classe de l'instance de test.

}

Algorithme 2.2 : algorithme de K plus proches voisins

Généralement, des attributs à n dimensions sont utilisés pour représenter les enregistrements d'apprentissage. Chaque enregistrement d'apprentissage est représenté par un point dans un espace à n dimensions. Les principaux éléments de ce processus sont :

1) un ensemble d'enregistrements stockés.

2) une mesure de similarité ou de distance pour calculer la distance entre deux échantillons.

3) la valeur de k, le nombre de voisins les plus proches.

Il existe plusieurs mesures de distance. Idéalement, la mesure doit être choisie de manière à minimiser la distance entre les enregistrements similaires et à maximiser la distance entre les enregistrements dissemblables.

2.3.3.1 Avantages

- Technique de classification facile à comprendre et facile à mettre en œuvre.
- On s'attend à ce que les méthodes d'apprentissage paresseux soient plus rapides lors de la phase d'apprentissage.
- Effectuez bien sur une application dans laquelle un enregistrement peut avoir de nombreuses étiquettes de classe. [28]

2.3.3.2 Inconvénients

- Les apprenants paresseux encourent des coûts de calcul coûteux lorsque le nombre de voisins potentiels à comparer à un enregistrement non étiqueté donné est important.
- Plus lent à la classification puisque tous les calculs sont retardés à ce moment.
- Sensible à la structure locale des données.
- Ils ont de grandes exigences de stockage.
- Ils sont sensibles au choix de la fonction de similarité utilisée pour comparer les instances.
- Ils n'ont pas de méthode fondée sur des principes pour choisir k, sauf par validation croisée ou technique similaire coûteuse en calculs. [28]

2.3.4 Réseaux de neurones

Un réseau de neurones est comme son nom l'indique, constitué de composants élémentaires appelés neurones. Modélisant un neurone biologique, le composant de base du réseau est une cellule à n entrées E1, E2 … En, et une sortie. Chaque entrée Ei possède un poids Wi. Le neurone combine les n entrées sous la forme d'une fonction linéaire. $\sum Wi * Ei$ Puis applique une fonction de transfert f appeler fonctionne d'activation aux résultats afin d'obtenir la sortie. La fonction f est, généralement, une fonction de seuil, qui change complètement la sortie si une petite modification est appliquée aux entrées. [34]

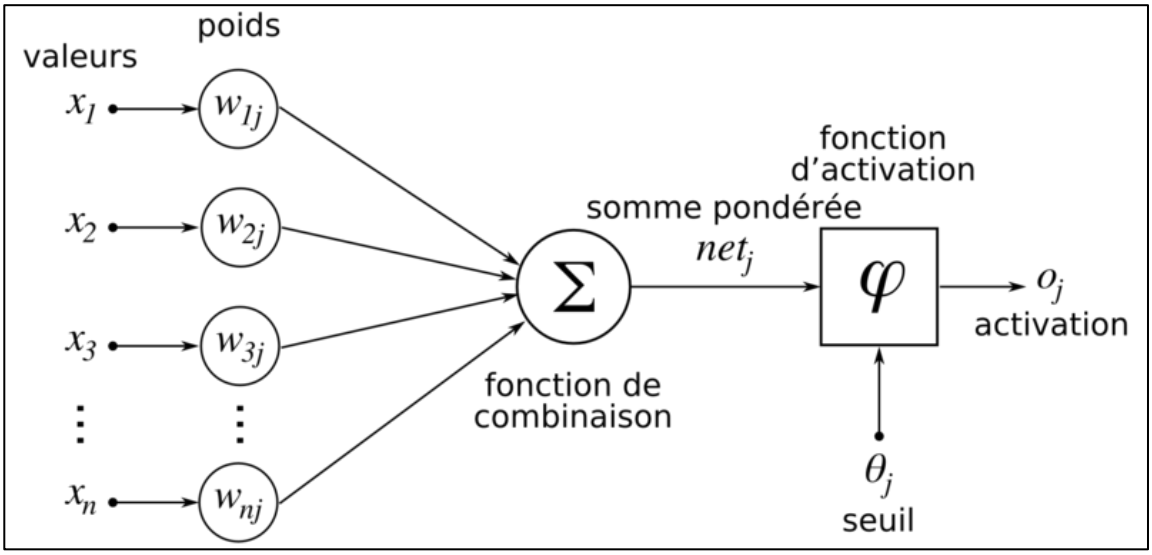

Figure 2. 2: Représentation d'un neurone.[69]

Un réseau se compose de multiples neurones interconnectés. Il est en général, organisé en couche, où chaque neurone de la couche i recevant en entrée certaines sorties des neurones de la couche i-1. Trois couches sont généralement suffisantes, l'une servant au codage, l'autre à la modélisation de l'intelligence et la troisième à la préparation de la sortie. [34]

Lors de l'apprentissage, le réseau commence par chercher un modèle en analysant un ensemble de données. Pour cela il injecte des entrées à sorties connues, il fait une propagation vers l'avant dans le réseau, puis il calcule la différence entre la sortie souhaitée et celle obtenue, puis fait une propagation arrière afin de corriger les poids en entrées en augmentant les poids des neurones qui donnent de bons résultats et en diminuant ceux des neurones qui donnent de mauvais résultats. Il existe plusieurs types de réseaux de neurones, les uns spécialisés en classification et prédiction et d'autres en groupage des données. Leurs apprentissages sont toujours difficiles à faire mais les résultats qu'ils produisent sont généralement supérieurs aux autres modèles de data mining.[34]

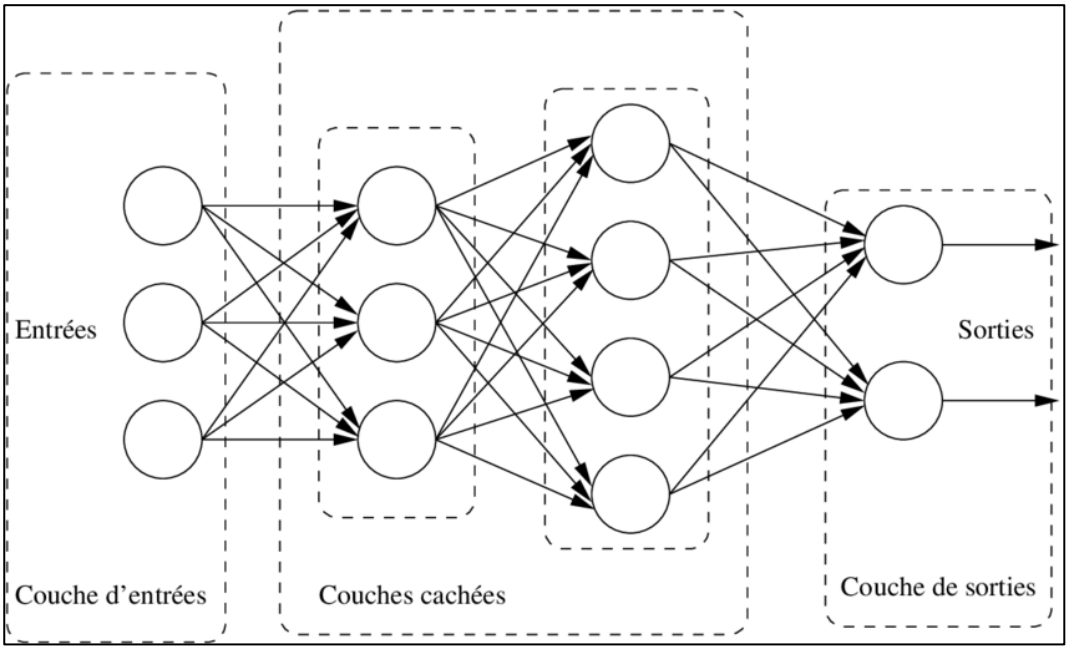

Figure 2. 3: exemple d'un réseau de neurones. [68]

2.3.4.1 Avantages :

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe.
- Résistance au bruit ou au manque de fiabilité des données.
- Comportement moins mauvais en cas de faible quantité de données.

2.3.4.2 Inconvénients

- Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage, qui jouent un rôle important dans la vitesse de convergence.
- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s).

2.3.5 Machine à vecteurs de support

Machine à vecteurs de support ou SVM est l'un des algorithmes d'apprentissage supervisé les plus populaires, qui est utilisé pour les problèmes de classification et de régression. Cependant, il est principalement utilisé pour les problèmes de classification dans l'apprentissage automatique.

L'objectif de l'algorithme SVM est de créer la meilleure ligne ou limite de décision capable de séparer l'espace à n dimensions en classes afin que nous puissions facilement placer le nouveau point de données dans la bonne catégorie à l'avenir. Cette frontière de meilleure décision est appelée un hyperplan.

SVM choisit les points/vecteurs extrêmes qui aident à créer l'hyperplan. Ces cas extrêmes sont appelés vecteurs de support et, par conséquent, l'algorithme est appelé machine à vecteur de

support. Considérez le diagramme ci-dessous dans lequel deux catégories différentes sont classées à l'aide d'une limite de décision ou d'un hyperplan : [33]

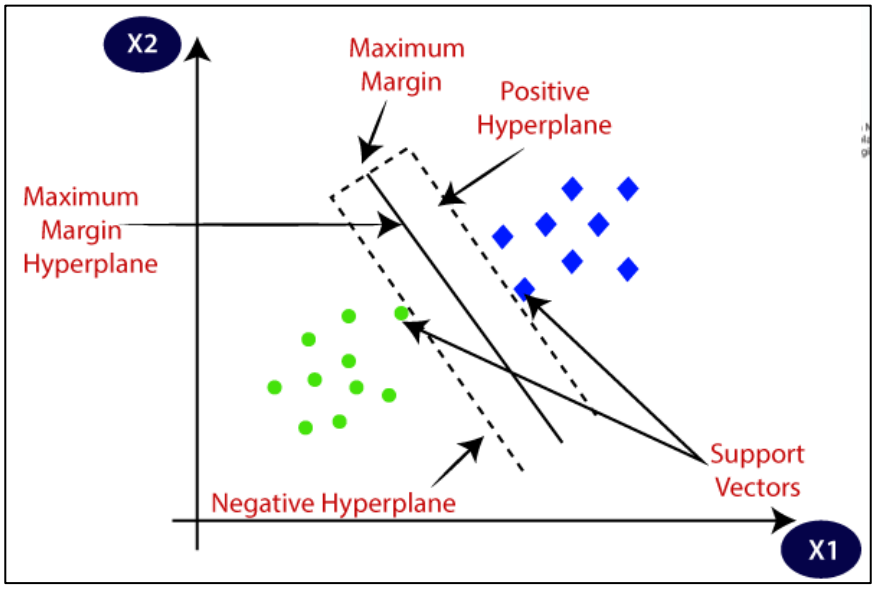

Figure 2. 4 : exemple démonstrative de la classification avec SVM.[33]

2.3.5.1 Hyperplan et vecteurs de support dans l'algorithme SVM

Hyperplan : il peut y avoir plusieurs lignes/limites de décision pour séparer les classes dans un espace à n dimensions, mais nous devons trouver la meilleure limite de décision qui aide à classer les points de données. Cette meilleure frontière est connue sous le nom d'hyperplan de SVM.

Les dimensions de l'hyperplan dépendent des entités présentes dans le jeu de données, ce qui signifie que s'il y a 2 entités, alors l'hyperplan sera une ligne droite. Et s'il y a 3 caractéristiques, alors l'hyperplan sera un plan à 2 dimensions.

Nous créons toujours un hyperplan qui a une marge maximale, c'est-à-dire la distance maximale entre les points de données. [33]

Vecteurs de soutien : Les points de données ou vecteurs les plus proches de l'hyperplan et qui affectent la position de l'hyperplan sont appelés vecteurs de support. Puisque ces vecteurs supportent l'hyperplan, donc appelé vecteur de support. [33]

2.3.5.2 Comment fonctionne SVM ?

SVM linéaire : Le fonctionnement de l'algorithme SVM peut être compris à l'aide d'un exemple. Supposons que nous ayons un jeu de données qui a deux balises (vert et bleu) et que le

jeu de données à deux caractéristiques x1 et x2. Nous voulons un classifieur capable de classer la paire (x1, x2) de coordonnées en vert ou en bleu.

Donc, comme il s'agit d'un espace à 2 dimensions, en utilisant simplement une ligne droite, nous pouvons facilement séparer ces deux classes. Mais plusieurs lignes peuvent séparer ces classes.

Par conséquent, l'algorithme SVM aide à trouver la meilleure ligne ou limite de décision ; cette meilleure limite ou région est appelée hyperplan. L'algorithme SVM trouve le point le plus proche des lignes des deux classes. Ces points sont appelés vecteurs supports. La distance entre les vecteurs et l'hyperplan est appelée marge. Et le but de SVM est de maximiser cette marge. L'hyperplan avec une marge maximale est appelé l'hyperplan optimal. [33]

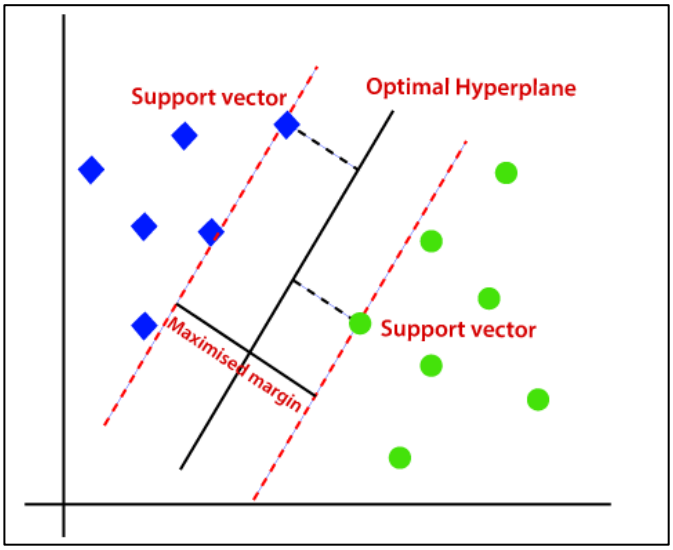

Figure 2. 5: SVM linéaire. [33]

SVM non linéaire : Si les données sont disposées linéairement, nous pouvons les séparer en utilisant une ligne droite, mais pour les données non linéaires, nous ne pouvons pas tracer une seule ligne droite, donc, pour séparer ces points de données, nous devons ajouter une dimension supplémentaire. Pour les données linéaires, nous avons utilisé deux dimensions x et y, donc pour les données non linéaires, nous ajouterons une troisième dimension z. Il peut être calculé comme suit :

$$Z=x^2+y^2$$

En ajoutant la troisième dimension, l'espace échantillon deviendra comme l'image ci-dessous :

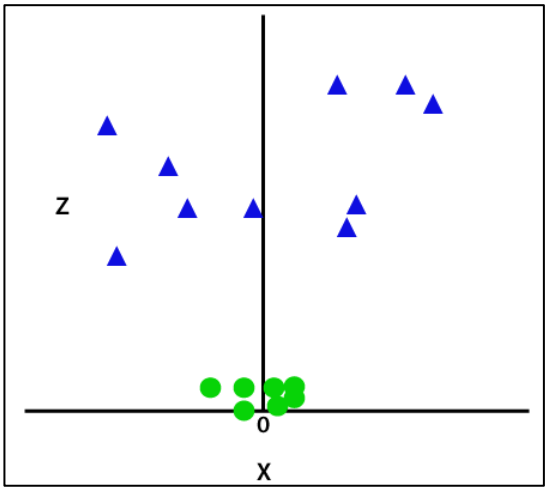

Figure 2. 6: l'ajout de la 3emedimension.[33]

Alors maintenant, SVM divisera les ensembles de données en classes. Puisque nous sommes dans l'espace 3D, il ressemble donc à un plan parallèle à l'axe des x. Si nous le convertissons dans un espace 2d avec z=1, il deviendra alors : [33]

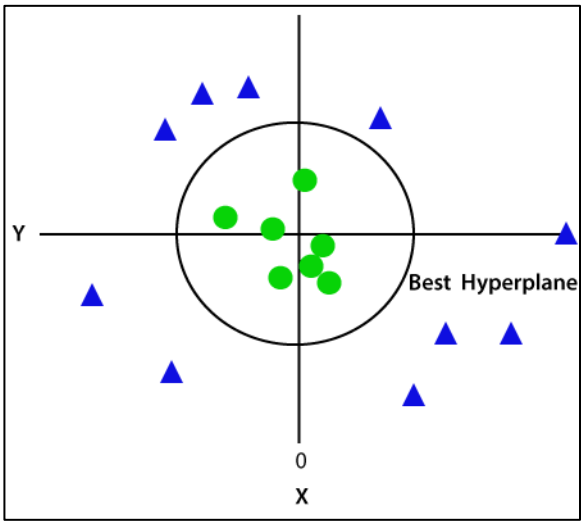

Figure 2. 7: la conversion en 2 Dimension.[33]

2.3.5.3 Fonctions du SVM kernel

Les algorithmes SVM utilisent un groupe de fonctions mathématiques appelées noyaux(kernels). La fonction d'un noyau est d'exiger des données en entrée et de les transformer dans la forme souhaitée.

Différents algorithmes SVM utilisent différents types de fonctions du noyau. Ces fonctions sont de différents types, par exemple, linéaire, non linéaire, polynomiale, fonction de base radiale (RBF) et sigmoïde.

Les fonctions du kernel renvoient le produit scalaire entre deux points dans un espace de fonctions extrêmement approprié. Ainsi en définissant une notion de ressemblance, avec un petit coût de calcul même dans le cas d'espaces de très grande dimension.

2.3.5.4 Fonctions populaires du SVM kernel

- **Kernel linéaire** : C'est le type de kernel le plus basique, généralement de nature unidimensionnelle. Cela s'avère être la meilleure fonction lorsqu'il y a beaucoup de fonctionnalités. Le noyau linéaire est principalement préféré pour les problèmes de classification de texte car la plupart de ces types de problèmes de classification peuvent être séparés linéairement. Les fonctions linéaires du kernel sont plus rapides que les autres fonctions.
- **Kernel polynomial** : C'est une représentation plus généralisée du noyau linéaire. Ce n'est pas aussi préféré que les autres fonctions du noyau car il est moins efficace et précis.
- **Kernel RBF :** C'est l'une des fonctions du noyau les plus préférées et les plus utilisées dans SVM. Il est généralement choisi pour les données non linéaires. Cela aide à faire une séparation appropriée lorsqu'il n'y a aucune connaissance préalable des données.
- **Kernel sigmoïde :** Il est surtout préféré pour les réseaux de neurones. Cette fonction du noyau est similaire à un modèle de perceptron à deux couches du réseau neurones, qui fonctionne comme une fonction d'activation pour les neurones.

2.3.5.5 Avantages :

- Une des méthodes les plus robustes et précises parmi toutes algorithmes bien connus.
- Il a une base théorique solide, ne nécessite qu'une douzaine d'exemples pour l'apprentissage, insensible au nombre de dimensions.
- Trouver la meilleure fonction de classification pour distinguer entre les membres des deux classes dans les données d'apprentissage.
- SVM est moins sujet au sur ajustement que les autres méthodes. [28]

2.3.5.6 Inconvénients

- C'est coûteux en calcul, car la résolution de cette méthode nécessite de grandes opérations matricielles et prend beaucoup du temps de calcul.
- Les SVM sont extrêmement lentes à apprendre.
- Le besoin en mémoire croît avec le carré du nombre d'exemples d'apprentissage.
- Mauvaise interprétabilité des résultats. [28]

2.4 Étude comparative

En générale, la machine a vecteurs de support et le réseau de neurones ont tendance à être beaucoup plus performants lorsqu'ils traitent des attributs continues. D'un autre côté, naïve bayes à tendance à mieux fonctionner lorsqu'ils traitent des caractéristiques discrètes/catégorielles. Pour le réseau neurones et la machine a vecteurs de support, une grande base de données est nécessaire pour atteindre sa précision de prédiction maximale, tandis que naïve bayes peut avoir besoin d'un ensemble de données relativement petit.

Il est généralement admis que k plus proches voisins est très sensible aux attributs non pertinentes cette sensibilité peut s'expliquer par le fonctionnement de l'algorithme. De plus, la présence des attributs non pertinents peut rendre l'apprentissage du réseau de neurones très inefficace, voire peu pratique. Le réseau de neurones et la machine a vecteurs de support fonctionnent bien lorsqu'une relation non linéaire existe entre les entités d'entrée et de sortie.

La méthode d'apprentissage k plus proches voisins ne nécessitent aucun temps dans la phase d'apprentissage car l'ensemble d'apprentissage est simplement stocké. La naïve bayes s'entraînent également très rapidement puisqu'elles ne nécessitent qu'un seul passage sur les données pour compter les fréquences, les arbres de décision sont également rapides dans les étapes d'apprentissage et de test. Tandis que le réseau de neurones et la machine a vecteurs de support prennent plus de temps par rapport aux arbres de décision, k plus proches voisins et naïve bayes.

Naïve bayes nécessite peu d'espace de stockage pendant les phases d'apprentissage et de classification le strict minimum est la mémoire nécessaire pour stocker les probabilités a priori et conditionnelles. L'algorithme k plus proches voisins de base utilise beaucoup d'espace de stockage pour la phase d'apprentissage, et son espace d'exécution est aussi grand que son espace d'apprentissage. Au contraire, pour tous les autres classifieurs de base mentionnée dans cette étude, l'espace d'exécution est généralement beaucoup plus petit que l'espace d'entraînement.

Naïve Bayes est naturellement robuste aux valeurs manquantes puisque celles-ci sont simplement ignorées dans le calcul des probabilités et n'ont donc aucun impact sur la décision finale. Au contraire, k plus proches voisins et le réseau de neurones nécessitent des enregistrements complets pour faire leur travail. De plus, k plus proches voisins est généralement considéré comme intolérant au bruit les erreurs dans les valeurs d'attribut, conduisant le classifieur à mal classer un nouvel enregistrement. Contrairement à l'algorithme k plus proches voisins, la plupart des arbres de décision sont considérés comme résistants aux bruits.

Les paramètres à régler par l'utilisateur sont un indicateur de la facilité d'utilisation d'un algorithme. Le réseau de neurones et la machine a vecteurs de support ont plus de paramètres que les autres techniques. L'algorithme k plus proche voisins n'a généralement qu'un seul paramètre (k) qui est relativement facile à régler.

Les algorithmes basés sur la logique sont tous considérés comme très faciles à interpréter, alors que le réseau de neurones et la machine a vecteurs de support ont une interprétabilité notoirement médiocre. K plus proches voisins est également considéré comme ayant une interprétabilité très médiocre car une collection non structurée des enregistrements d'apprentissage est loin d'être lisible.

Enfin, aucun algorithme de classification ne peut surpasser uniformément les autres algorithmes sur tous les ensembles de données [28].

Les caractéristiques	Arbre de decision	Réseau de neurones	Naïve Bayes	K Plus Proche Voisin	Machine a Vecteurs de Support
Précision en général	**	***	*	**	****
Vitesse d'apprentissage par rapport au nombre des attributs et le nombre d'instances	***	*	****	****	*
Vitesse de classement	****	****	****	*	****
Tolérance aux valeurs manquantes	***	*	****	*	**
Tolérance aux attributs non pertinents	***	*	**	**	****
Tolérance aux attributs redondants	**	**	*	**	***
Tolérance aux attributs interdépendants (ex. problèmes de parité)	**	***	*	*	***
Traitement des attributs discret/binaire/continu	****	***(non discrète)	***(non continu)	***(non discret)	**(non discret)
Tolérance au bruit	**	**	***	*	**
Faire face au danger de sure ajustements	**	*	***	***	**
Explication capacité/transparence des connaissances/classifications	****	*	****	**	*
Gestion des paramètres du modèle	***	*	****	***	*

Tableau 2.1: comparaison entre les 5 algorithmes de classification [28]

2.5 Les méthodes ensemblistes

L'idée de combinaison de modèles reprend le processus naturel de prise de décision. Dans un cas complexe pour lequel il existe plus d'un avis, le décideur a souvent recours à plusieurs experts pour se forger une conviction et prendre une décision. C'est exactement ce que fait la combinaison de modèles en exploitant et évaluant les résultats de plusieurs modèles pour en déduire un résultat. Les méthodes les plus connues de combinaison de modèles sont le bagging, le boosting et le stacking. La combinaison de plusieurs modèles peut augmenter les performances de classification

ou de prédiction mais elle a l'inconvénient de rendre le processus de prise de décision plus lourd et plus complexe et parfois difficile interpréter. Les trois méthodes suscitées sont des techniques générales qui peuvent être utilisées pour la prédiction numérique et la classification en association avec différents algorithmes de datamining. [34]

2.5.1 Bagging

Le nom bagging est dérivé de l'expression *"Bootstrap aggregating"*. L'idée est d'avoir plusieurs échantillons avec lesquelles sont construits un nombre équivalent de modèles de datamining. Pour chaque instance (enregistrement) des données initiales est calculé sa classe respective en utilisant chacun des modèles. Pour une instance donnée, la classe la plus fréquemment prédite par les différents modèles est choisie comme classe de l'instance. Pour avoir plusieurs échantillons d'apprentissage et de test, les données de départ sont transformées en supprimant aléatoirement un nombre d'instances de départ et en dupliquant d'autres pour remplacer ceux supprimées et garder la même taille des données de départ. Les ensembles de données ainsi produits sont utilisés pour l'apprentissage et les instances supprimées son utilisées pour les tests. La combinaison de modèles produits est généralement plus performante que l'utilisation d'un seul modèle obtenu en utilisant la totalité de l'échantillon initial.

Pour la prédiction des valeurs numériques, il suffit de prendre la moyenne des valeurs prédites par les différents modèles comme la valeur à prédire de l'instance considérée. Il est aussi possible d'utiliser un vote pondéré ou de calculer une moyenne pondérée au lieu d'un vote et une moyenne simple. Si les données initiales sont stables, l'apport du bagging est insignifiant puisque les modèles générés seront sensiblement les mêmes. [34]

L'algorithme de l'approche bagging est résumé comme suit :

Génération du modèle

1. Pour chacune des itérations Faire
2. Créer un nouvel ensemble de données en faisant des remplacements dans l'ensemble de données initial
3. Créer un modèle en appliquant l'algorithme de data mining

Classification

4. Pour chacun des modèles générés
5. Prédire la classe de chaque instance
6. Assigner à chaque instance la classe la plus fréquemment prédite par les modelés

Algorithme 2.3 : algorithme du bagging. [34]

2.5.2 Boosting

L'idée est la même que pour le bagging. Le boosting utilise le vote multiple pour la classification et le calcul de la moyenne pour la prédiction des valeurs numériques en combinant des modèles de même type, générés à partir de différents échantillons. Sauf que le boosting est un processus itératif, où chaque modèle généré est directement influencé par le modèle précédemment produit. Le nouveau modèle favorise les instances mal classées par le modèle précédent. Aussi, le boosting détermine la contribution du modèle en fonction de sa performance plutôt que de considérer les modèles comme égaux. [34]

L'algorithme du boosting est comme suit :

Génération du modèle : Un même poids p est attribué à chaque instance
7. Pour chacune des t itérations Faire
8. Créer un modèle en appliquant l'algorithme de data mining
9. Calculer l'erreur e du modèle en utilisant les poids des instances
10. Si e = 0 ou e >= 1/2 : Terminer la génération du modèle
11. Pour chaque instance des données initiales
12. Si l'instance est bien classée par le modèle : p = p * e /(1-e)
13. Normaliser pour que la somme des poids des instances reste la même
14. Normaliser le poids de toutes les instances : p = p * (∑ anciens p) 1 (∑ nouveaux p)

Classification : Un poids de valeur zéro est attribué à toutes les classes
15. Pour chacun des t (ou moins) modèles Faire
16. Pondérer le poids de l'instance pour un modèle en fonction de l'erreur total du modèle
17. Ajouter -Iog(e /(1 - e)) au poids de chaque classe prédite par le modèle
18. Retourner la classe avec le plus grand poids

Algorithme 2.4 : algorithme du boosting.[34]

Le boosting peut être adapté aux algorithmes de datamining qui n'acceptent pas les valeurs avec poids.

Pour cela, il suffit de générer à chaque nouvelle itération un échantillon à partir du précédent en utilisant les poids des instances. Le même principe que le bagging est utilisé, sauf que les probabilités de choix des instances à supprimer et à dupliquer ne seront plus égales pour toutes les instances mais dépendront du poids de chaque enregistrement.[34]

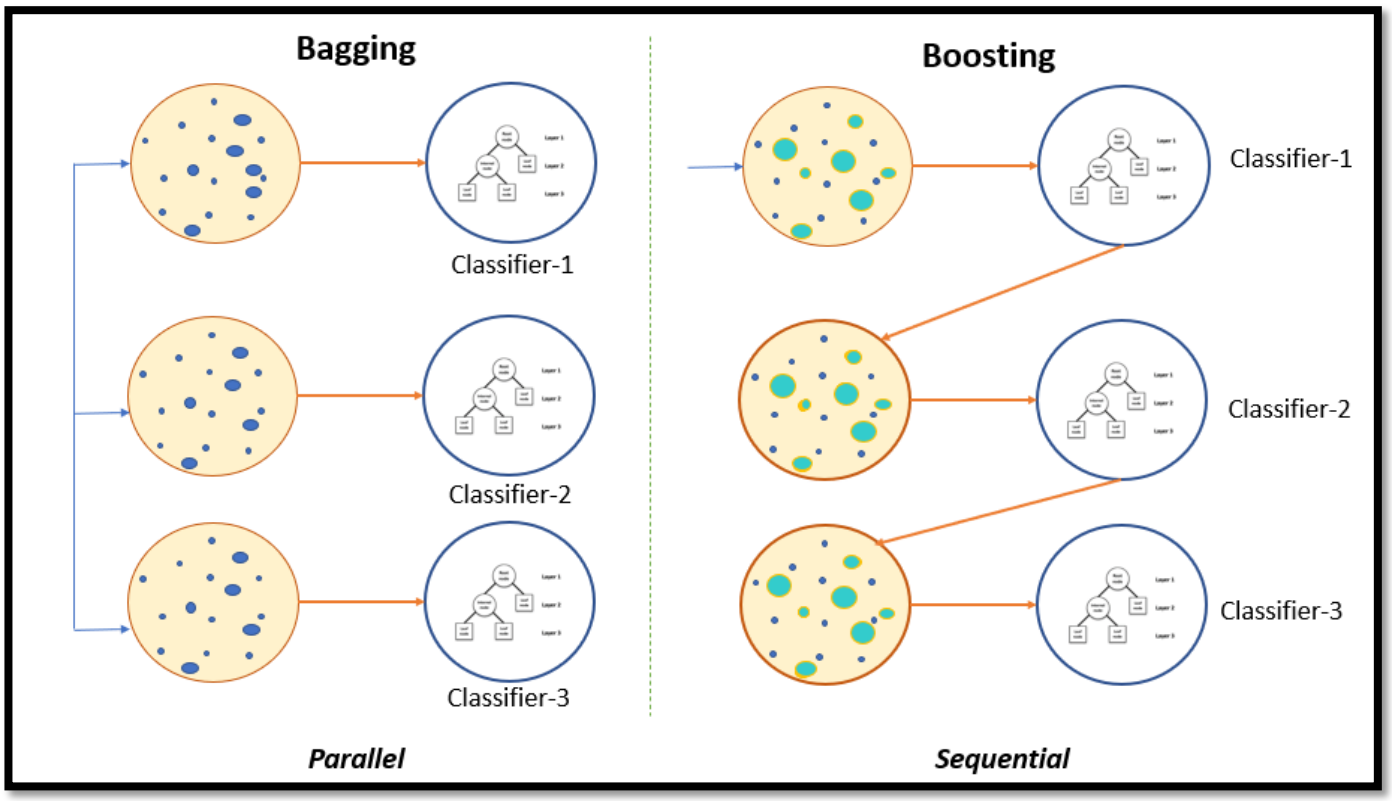

Figure 2. 8: la différence entre le bagging et le boosting.[65]

2.5.3 Stacking

Inventé par David Wolpert, le stacking est un méta-modèle qui combine des modèles de différents types plutôt que des modèles de même type comme dans le cas du bagging et boosting. Combiner plusieurs modèles de types différents en utilisant le vote majoritaire ou en choisissant tout simplement le modèle le plus performant sont les deux méthodes les plus triviales de méta-modèle. Le problème avec le vote est qu'il peut générer de très mauvais résultats si la majorité des modèles ne sont pas performants, voilant ainsi les bons résultats des modèles restants. Le stacking essaye de corriger cette lacune en utilisant un algorithme ou méta-modèle qui pourrait estimer les performances des modèles de base et ainsi combiner leurs résultats intelligemment.

Pour son propre apprentissage, le méta-modèle utilise les résultats (classes) générés par les modèles de base. Malheureusement, la simple combinaison des classes prédites par les modèles de base ne permet pas un bon apprentissage du méta-modèle. Cela conduit à la domination du modèle de base le plus performant et le plus surentraîné et de ce fait ne garantit pas une bonne performance sur de nouvelles données. Une première solution est de diviser les données initiales en deux. La première partie de données est utilisée pour l'apprentissage des modèles de base et la deuxième partie est utilisée pour l'apprentissage du méta-modèle. Comme la deuxième partie des données n'a pas été utilisée pour l'apprentissage des modèles de base, leurs performances vont diminuer lors de leurs utilisations pour l'apprentissage du méta-modèle évitant ainsi la domination d'un seul modèle. Une fois l'apprentissage du méta-modèle terminé, les modèles de base peuvent être réentraînés sur la totalité des données augmentant ainsi leurs performances. L'inconvénient d'une telle approche est que le méta-modèle n'est pas entraîné sur la totalité des données.

Une deuxième solution est d'utiliser la même technique décrite plus haut pour le bagging, utilisée aussi dans la validation croisée des modèles. C'est à dire que le méta-modèle utilisera la technique du bagging, cela lui permettra de s'entraîner sur toutes les données de départ. L'utilisation du bagging augmente considérablement le temps d'apprentissage total, parce que les modèles de base doivent s'entraîner et être testés sur chaque échantillon généré par le bagging.

Pour la génération du méta-modèle, il est recommandé d'utiliser un algorithme simple car la classification se fait au niveau des modèles de base, le méta-modèle ne fait que combiner les résultats.

L'extension du stacking à la prédiction des valeurs numériques est possible et triviale, il suffit de se limiter pour la génération des modèles de base à des algorithmes qui supportent les valeurs numériques et remplacer les attributs représentant les classes dans les modèles de base et le méta-modèle par des attributs de type numérique. [34]

L'algorithme du stacking est comme suit :

Entrée : Base de données initiale D $= \{x_i, y_i\}_{i=1}^{m}\{x_i \in R_n, y_i \in Y\}$.

Sortie : un classifieur d'ensemble H

1- Étape 1 : construire les classifieurs du premier niveau sur l'ensemble initiaux complet

2- Pour t ← 1 à T faire

3- Construire un classifieur h_t basé sur D.

4- Fin pour

5- Etape 2 : Construire de nouveaux ensembles de données à partir de D

6 -pour i← 1 to m faire

7 - Obtenir un enregistrement$\{x_i', y_i\}$ ou $x_i' = \{h_{k1}(x_1), h_{k2}(x_2), \ldots, h_{kT}(x_i)\}$.

8- Fin pour

9- Etape 3 : construire un classifieur de second niveau.

10- Construire un classifieur h' à partir de l'ensemble $\{x_i', y_i\}$

11- Retourner H(x) $= h'(h_1(x), h_2(x), \ldots, h_T(x))$

Algorithme 2.5 : algorithme du stacking.[71]

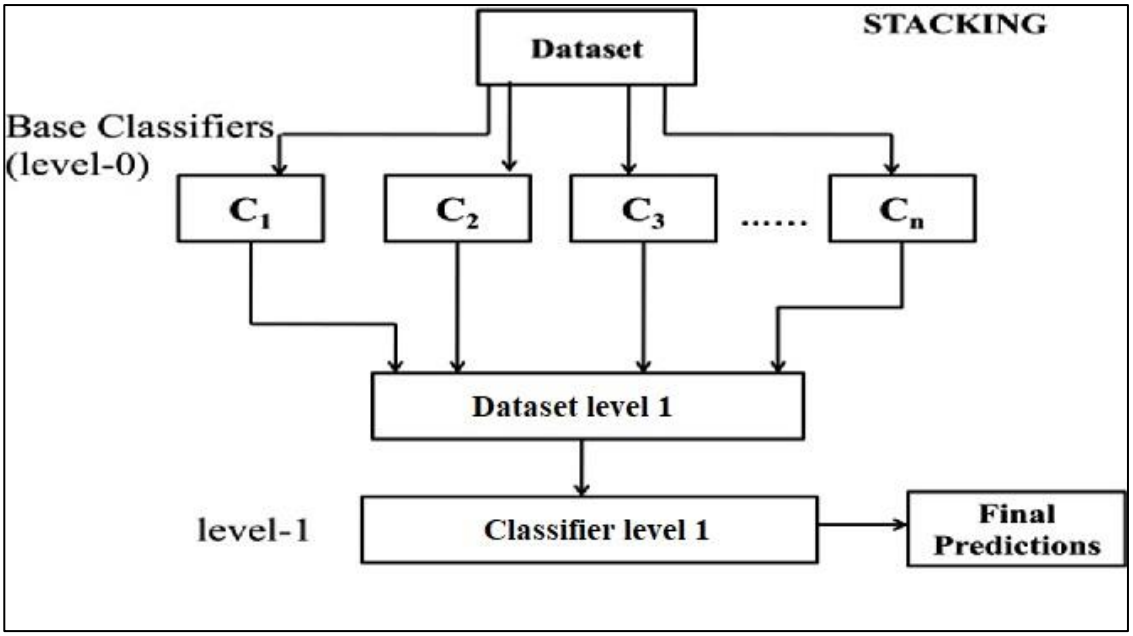

Figure 2. 9: schéma de la méthode Stacking.[66]

2.6 Conclusion

Dans ce chapitre, nous avons présenté une étude comparative entre les algorithmes de classification, à savoir l'arbre de décision, naïve bayes, K plus proches voisins, réseau de neurone et machine a vecteurs de support, ensuite, nous avons présenté quelques méthodes ensemblistes pour accroître les performances du modèle d'apprentissage, et parvenir à un niveau de précision supérieur à celui qui serait réalisé si on utilisait un de ces algorithmes pris séparément. L'étude comparative avait montré que chaque algorithme avait son propre ensemble d'avantages et d'inconvénients ainsi que son propre domaine d'implémentation. Aucun algorithme ne peut satisfaire tous les critères. On peut étudier un classifieur qui peut être construit par une intégration de deux classifieurs ou plus en combinant leur force.

Dans le cadre de notre PFE, nous nous intéressons dans le chapitre suivant à l'application de la méthode ensembliste stacking et plusieurs algorithmes de classification sur des données médicales pour augmenter la précision des résultats.

Chapitre 3 : Implémentation et évaluation des résultats

3.1 Introduction

Dans le domaine de machine Learning et data mining, les chercheurs essaient toujours d'améliorer leurs modèles, ils essaient de générer des modèles robustes et fiables, en particulier dans la précision. Il existe de nombreuses façons d'améliorer les résultats des modèles de classification, l'une des méthodes est la méthode ensembliste. L'idée principale de la méthode d'ensemble est de combiner un ensemble de modèles, chacun résolvant la même tâche d'origine, afin d'obtenir un meilleur modèle global composite, avec des estimations ou des décisions plus précises et plus fiables que celles qui peuvent être obtenues à partir d'un modèle unique.

Dans ce chapitre, nous allons implémenter une méthode ensembliste pour résoudre les problèmes des algorithmes de classification de base à savoir la précision et la stabilité des résultats. Pour cela, nous allons combiner 5 classifieurs de base (k plus proche voisin, arbre de décision, réseaux de neurones, support vecteur machine, naïve bayes) pour construire la méthode ensembliste stacking, les paramètres des algorithmes de base sont ajustés manuellement ou à l'aide de la méthode grid search. Nous nous intéressons au domaine médical donc nous allons appliquer cette méthode ensembliste stacking sur 5 bases de données médicales réelles et synthétiques pour évaluer la performance et la stabilité de cette méthode par rapport aux classifieurs de base. Pour valider cette méthode, nous utiliserons la validation croisée, dont nous parlerons plus en détail dans ce chapitre.

3.2 Notions de base

Données bruyantes : Les données bruyantes sont des données contenant une grande quantité d'informations supplémentaires sans signification appelées bruit. Cela inclut la corruption des données et le terme est souvent utilisé comme synonyme de données corrompues.

Distance euclidienne : En mathématiques, la distance euclidienne entre deux points dans l'espace euclidien est la longueur d'un segment de ligne entre les deux points. Elle peut être calculée à partir des coordonnées cartésiennes des points.

Distance Manhattan : La distance de Manhattan, appelée aussi taxi-distance, est la distance entre deux points parcourus par un taxi lorsqu'il se déplace dans une ville où les rues sont agencées selon un réseau quadrillage.

Elagage : élagage dans les arbres de décisions c'est supprimer les branches peu représentatives pour garder de bonnes performances prédictives (généralisation)

Sur-ajustement : (overfitting en anglais) est l'un des pires ennemis du Data Scientistes. Il s'agit d'un problème fréquemment rencontré en Machine Learning. Il survient lorsque le modèle essaie de trop s'adapter aux données d'entraînement. Il est trop flexible et trop complexe et s'adapte à des données qui ne sont pas forcément à prendre en compte.

3.3 Bibliothèque et langage utilisé

3.3.1 Java

Nous avons utilisé le langage java dans notre projet, java est un langage de haut niveau. Langage de programmation orienté objet basé sur les classes et conçu pour avoir le moins de dépendances d'implémentation possible. Il s'agit d'un langage de programmation à usage général, de s'exécuter n'importe où (WORA), ce qui signifie que le code java compilé peut s'exécuter sur toutes les plates-formes prenant en charge java sans qu'il soit nécessaire de le recompiler. Les applications java sont généralement compilées en byte code pouvant s'exécuter sur n'importe quelle machine virtuelle Java (JVM), quelle que soit l'architecture informatique sous-jacente. Le runtime java fournit des fonctionnalités dynamiques (telles que la réflexion et la modification du code d'exécution) qui ne sont généralement pas disponibles dans les langages compilés traditionnels. [35].

L'une des principales raisons du choix de ce langage est que java est plus rapide que n'importe quel langage. Il est raclé pour être 25 fois plus rapide que python [36] [37] et excellent lorsqu'il s'agit de mettre à l'échelle des applications, ce qui en fait le meilleur choix pour créer des applications ML et IA volumineuses et plus complexes.

Java possède un grand nombre de bibliothèques et d'outils pour le ML et Data Mining, parmi les plus populaires étant Weka, Java-ML, MLlib et Deeplearning4j, [37] qui sont exploités pour résoudre la plupart des problèmes d'apprentissage automatique, dans notre étude. On a utilisé weka comme bibliothèque.

3.3.2 Weka

Weka (acronyme pour Waikato environment for knowledgeanalysis,) en français : « environnement Waikato pour l'analyse de connaissances ») est une suite de logiciels d'apprentissage automatique écrite en java et développée à l'université de Waikato en Nouvelle-Zélande. L'espace de travail Weka contient une collection d'outils de visualisation et d'algorithmes pour l'analyse des données et la modélisation prédictive la version plus récente entièrement basée sur java (Weka 3), pour laquelle le développement a débuté en 1997, est désormais utilisée dans

beaucoup de domaines d'application différents en particulier pour l'éducation et la recherche, les principaux points forts de Weka sont :

- Libre et gratuit, distribué selon les termes de la licence publique générale GNU .
- Portable car il est entièrement implémenté en java et donc fonctionne sur quasiment toutes les plateformes modernes, et en particulier sur quasiment tous les systèmes d'exploitation actuels.
- Contient une collection complète de préprocesseurs de données et de techniques de modélisation.
- Facile à utiliser par un novice en raison de l'interface graphique qu'il contient

Weka supporte plusieurs outils d'exploration de données standards en particulier, des préprocesseurs de données, des agrégateurs de données (data clustering), des classifieurs statistiques, des analyseurs de régression, des outils de visualisation, et des outils d'analyse discriminante. Toutes les techniques de Weka reposent sur la supposition que les données sont disponibles dans un unique fichier plat où une relation binaire, ou chaque type de données est décrit par un nombre fixe d'attributs (les attributs ordinaires, numériques ou symboliques, quelques autres types d'attributs sont aussi supportés) [38].

3.4 Approche proposée

3.4.1 Principe

Nous avons choisi la méthode ensembliste stacking la raison de ce choix est que seule cette méthode peut nous permettre de combiner plusieurs types des classifieurs en une seule méthode ensembliste, contrairement à d'autres méthodes comme le boosting ou bagging qui permettent de combiner plusieurs classifieurs mais de même type.

Pour comprendre comment construire un modèle ensembliste stacking il faut tout d'abord comprendre l'idée de la validation croisée, car cette idée est utilisée dans la première étape de préparation des données pour le méta apprentissage.

3.4.1.1 Validation croisée

On parle en général de validation croisée à K bloque pour désigner une technique d'évaluation d'un algorithme de machine Learning. Cela consiste à découper la base de données en K sous-ensemble (ou K folds) puis prendre un des K sous-ensemble comme base de données de validation et les K-1 restants comme base de données pour l'apprentissage. On répète l'opération sur toutes les combinaisons possibles. On obtient K mesures de performance dont la moyenne représente la performance de l'algorithme. [39].

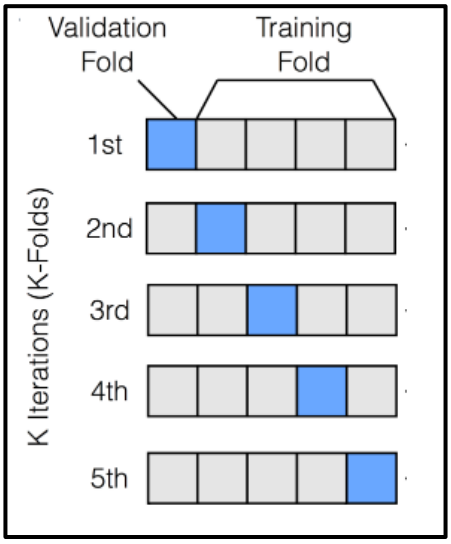

Figure 3. 1 : exemple de validation croisée.[39]

3.4.1.2 Etape 1

L'idée de la validation croisée est appliquée dans la première étape, elle consiste à diviser la base de données primaire en k sous-ensembles, puis chaque classifieur de base est entrainé uniquement sur k -1 sous-ensembles. Ensuite appliqué au sous-ensemble restant, et la sortie de tous les classifieurs de premier niveau constitue l'enregistrement d'entrée pour le classifieur de second niveau.

Un nouvel ensemble de données est construit les attributs de ce nouvel ensemble de données sont les prédictions des classifieurs de base et la classe d'attribut est la même que l'ensemble de données initial.

3.4.1.3 Etape 2

Un classifieur de base est entrainé sur la nouvelle base de données construite. C'est le classifieur de niveaux 2.

3.4.1.4 Etape 3

Reconstruire les classifieurs de premier niveau sur l'ensemble d'apprentissage initial afin que tous les exemples d'apprentissage soient utilisés.

On peut résumer les étapes dans l'algorithme ci-dessous :

Entrée : Base de données initialeD $= \{x_i, y_i\}_{i=1}^{m}\{x_i \in R_n, y_i \in \mathrm{Y}\}$.
Sortie : Un classifieur d'ensemble H.

1- Étape 1 : utilisation approche de validation croisée lors de la préparation d'un ensemble d'entraînement pour le classifieur de deuxième niveau.
2- Séparez aléatoirement D en K sous-ensembles de taille égale : D $= \{D_1, D_2, \ldots, D_K\}$
3- Pour k ← 1 à K faire
4- étape 1.1 : Construire les classifieurs du premier niveau
5- Pour t ← 1 à T faire
6- Construire un classifieurh_{kt} $pour$ $D \backslash D_k$.
7- Fin pour.
8- étape1.2 : Construire un nouvel ensemble d'enregistrement pour le classifieur de second niveau
9- Pour $x_i \in D_k$ faire
10- Obtenir un enregistrement$\{x_i', y_i\}$ ou $x_i' = \{h_{k1}(x_1), h_{k2}(x_2), \ldots, h_{kT}(x_i)\}$.
11- Fin pour.
12-Fin pour.
13- Etape 2 : construire un classifieur de second niveau.
14- Construire un classifieur h' à partir de l'ensemble $\{x_i', y_i\}$.
15- Étape 3 : Reconstruire les classifieurs du premier niveau sur l'ensemble initiaux complet
16- Pour t ← 1 à T faire
17- Construire un classifieur h_t basé sur D.
18- Fin pour
19- Retourner H(x) $= h'(h_1(\mathrm{x}), h_2(\mathrm{x}), \ldots, h_T(\mathrm{x}))$.

Algorithme 3.6 : algorithme stacking proposée. [71]

Pour classer un nouvel enregistrement, il suffit d'appliquer les classifieurs de base sur ce nouvel enregistrement, les prédictions obtenues forment l'enregistrement du jeu de données de second niveau. Puis faire appliquer le classifieur de second niveau sur l'enregistrement des prédictions obtenues pour obtenir la prédiction finale.

3.4.2 Les classifieurs de base

Il faut noter que chaque classifieur de base contient de nombreux paramètres, dans notre proposition, nous avons choisi les paramètres qui ont un impact direct sur la précision. Ces paramètres sont ajustés après plusieurs expérimentations.

3.4.2.1 K-plus proches voisins

Il y a beaucoup d'algorithmes de k plus proche voisin avec la même idée de base comme K-star ou IBL (Instance-Based Learning) [40] mais selon cette étude [41] IBK fonctionne mieux que les deux autres. L'algorithme IBK utilise la méthode de vote pour décider de la classification du nouvel exemple et le nombre de votes est indiqué par la valeur "k".

3.4.2.1.1 Paramètres

- **Nombre de plus proches voisins k**

Si k est trop petit, le résultat peut être sensible aux points de bruit. D'autre part, si k est trop grand, alors le voisinage peut inclure trop de points d'autres classes. Une estimation de la meilleure valeur pour k peut être obtenue par validation croisée. Dans notre implémentation, Le nombre de voisins les plus proches est spécifié à l'aide d'un focus de validation croisée à une limite supérieure donnée par la valeur spécifiée. [41]

- **Calcule de distance**

Le choix de la mesure de distance est une autre considération importante. Généralement, des mesures de distance euclidiennes ou de Manhattan sont utilisées. Dans notre implémentation nous avons choisi la mesure de distance euclidiennes [41].

3.4.2.2 Naïve bayes

Comme nous l'avons mentionné dans le deuxième chapitre l'algorithme naïve bayes est un simple algorithme probabiliste qui calcule un ensemble de probabilités par compter 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 de classe. L'algorithme a tendance à bien fonctionner et apprendre rapidement dans divers problèmes de classification**.** [42].

3.4.2.2.1 Paramètres

- **Discrétisation**

Nous avons utilisé le classifieur naïve bayes de la bibliothèque weka et le seul paramètre que nous avons ajusté est le paramètre de discrétisation, c'est le processus de conversion d'attributs

continu en valeurs nominales. Cette technique rend le modèle plus général et donne un meilleur résultat en général selon Ying Yang · Geoffrey I. [43].

3.4.2.3 Arbre de décision

L'algorithme C4.5 est un algorithme de classification qui produit des arbres de décision basés sur la théorie de l'information. Il s'agit d'une extension de l'ancien algorithme ID3 de Ross Quinlan, également connu dans Weka sous le nom de J48, J signifiant Java. Les arbres de décision générés par C4.5 sont utilisés pour la classification, l'implémentation J48 de l'algorithme C4.5 comportes de nombreuses fonctionnalités supplémentaires, notamment la prise en compte des valeurs manquantes, l'élagage des arbres de décision, etc. Dans l'outil de data mining WEKA, J48 est une implémentation java open source de l'algorithme C4.5. J48 permet la classification via des arbres de décision ou des règles générées à partir de ceux-ci, le sur ajustement est un risque majeur avec ces arbres de décision, il est donc essentiel de valider le modèle avant de le déployer sur l'ensemble de test. Au fur et à mesure que cet arbre de décision grandit, il a naturellement tendance à sur-ajuster les données. L'élagage est un processus par lequel le plus grand arbre qui est le plus généralisable est sélectionné, et toutes les branches en dessous de ce niveau sont élaguées. Cela améliore considérablement les performances sur les nouvelles données [46] [44], weka propose plusieurs options associées à l'élagage des arbres en cas de sur-ajustement et l'un des paramètres les plus importants, le facteur de confiance.

3.4.2.3.1 Paramètres

- **Facteur de confiance (confidence factor)**

Cela détermine l'agressivité du processus d'élagage. Plus cette valeur est élevée, plus vous êtes "confiant" que les données dont vous apprenez elles ont une bonne représentation de tous les événements possibles, et donc moins il y aura d'élagage. Des valeurs plus petites induisent plus d'élagage. Cela affecte considérablement les performances du classifieur. [45]

3.4.2.4 Machine a vecteurs de support

Libsvm est une bibliothèque d'apprentissage automatique open source utilisée pour la mise en œuvre de l'algorithme de machine à vecteurs de support. Il est développé à l'université nationale de Taiwan par Chih-Chung Chang et Chih-Jen Lin [47] c'est une bibliothèque pour les machines à vecteurs de support (SVM). Son objectif est d'aider les utilisateurs à utiliser facilement SVM comme un outil. Comme nous l'avons expliqué dans le chapitre précédent, le but du classifieur SVM est de trouver le meilleur point (en 1-D), ligne (en 2-D), plan (3D), hyperplan (en plus de 3-D) pour séparer les classes, SVM contient plusieurs paramètres, nous avons choisi d'ajuster les paramètres qui influencent la précision de la classification. [47].

3.4.2.4.1 Paramètres

- **Types de Kernel**

La fonction noyau peut être l'une des suivantes :

- Linéaire.
- Polynomial.
- RBF (radial basis function).
- Sigmoid. [47]

- **Paramètres du kernel RBF**

Lors de l'apprentissage d'un modèle SVM avec kernel Radial Basis Function (RBF), deux paramètres doivent être pris en compte : Cost et Gamma. Le paramètre C (cost), commun à tous les noyaux SVM, compense la mauvaise classification des exemples d'apprentissage par la simplicité de la surface de décision. Un C bas rend la surface de décision lisse, tandis qu'un C élevé vise à classer correctement tous les exemples d'apprentissage. Gamma définit l'influence d'un seul exemple d'apprentissage. Plus le gamma est grand, plus les autres exemples doivent être proches pour être affectés. Le bon choix de C et de gamma est essentiel aux performances du SVM. Et pour faire ça, il fallut beaucoup de temps pour ajuster ces paramètres manuellement en répétant beaucoup d'expériences, nous avons utilisé ce qu'il appelle GridSearchCV [libsvm 2] avec C et gamma espacés de manière exponentielle pour choisir de bonnes valeurs.

3.4.2.4.2 GridSearchCV

C'est une méthode d'optimisation (hyper parameter optimization) qui va nous permettre de tester une série de paramètres et de comparer les performances pour en déduire le meilleur paramétrage.

Il existe plusieurs manières de tester les paramètres d'un modèle et le GridSearch est une des méthodes les plus simples. Pour chaque paramètre, on détermine un ensemble de valeurs que l'on souhaite tester. Dans notre cas :

- Paramètre Cost : {0.01, 0.1, 0.5, 1}
- Paramètre Gamma : {0.01, 0.1, 0.5, 1}

Le GridSearch croise simplement chacune de ces hypothèses et va créer un modèle pour chaque combinaison de paramètres. Dans notre exemple nous aurons 16 modèles à construire. Vous comprenez donc rapidement qu'il ne faut pas abuser du GridSearch parce qu'il augmente considérablement les temps de calcul. [48]

3.4.2.5 Réseaux de neurones

Dans la bibliothèque weka, le classifieur de réseau de neurones est sous le nom multicouche perceptron ce classifieur contient beaucoup de paramètres, nous avons choisi d'ajuster 3 paramètres. [49].

3.4.2.5.1 Paramètres

- **Nombre de couches cachées**

Une couche cachée dans un réseau neuronal artificiel est une couche entre la couche d'entrée et la couche de sortie, où les neurones artificiels reçoivent un ensemble d'entrées pondérées et produisent une sortie via une fonction d'activation. C'est une partie typique de presque tous les réseaux neurones dans lesquels les ingénieurs simulent les types d'activité qui se déroulent dans le cerveau humain.

Un manque de connexions entre les couches peut rendre le réseau incapable de résoudre le problème tandis que beaucoup de connexions peuvent entraîner un sur-ajustement des données d'apprentissage. Surtout, lorsque nous utilisons un nombre élevé de couches et de neurones. Nous optimisons le nombre de couches cachées et le nombre de neurones dans chaque couche cachée pour augmenter la vitesse et l'efficacité du réseau de neurones. [49].

- **Taux d'apprentissage (Learning rate)**

Le taux d'apprentissage (α) détermine la vitesse d'itération de l'ajustement du poids dans le réseau de neurones en fonction de la fonction de perte de gradient, plus le taux d'apprentissage est petit, plus le déclin le long du gradient de perte est lent. Un taux d'apprentissage plus faible peut éviter que des valeurs potentiellement optimales ne soient négligées, obtenant ainsi une plus grande précision d'apprentissage, mais ce processus nécessite un temps de convergence plus long. Généralement, le réglage du taux d'apprentissage dépend de l'expérience, de la taille du modèle et de la complexité numérique. Par conséquent, le taux d'apprentissage doit être ajusté pour diverses conditions de données. Représente l'état d'optimisation pour différents taux d'apprentissage. Un faible taux d'apprentissage à une vitesse de convergence lente mais garantit que le minimum est identifié à chaque étape de l'apprentissage pour obtenir une précision d'apprentissage optimale. En revanche, un taux d'apprentissage élevé peut accélérer la vitesse de convergence mais peut se fixer sur une solution sous-optimale. [50]

- **Nombre d'époques**

Une époque correspond au moment où un ensemble de données entier est transmis en avant et en arrière via le réseau de neurones une seule fois. Faire passer l'ensemble de données via un réseau

de neurones une seule fois ne suffit pas. Il faut transmettre l'ensemble de données complet plusieurs fois au même réseau de neurones pour bien ajustez-les connexions entre les neurones.

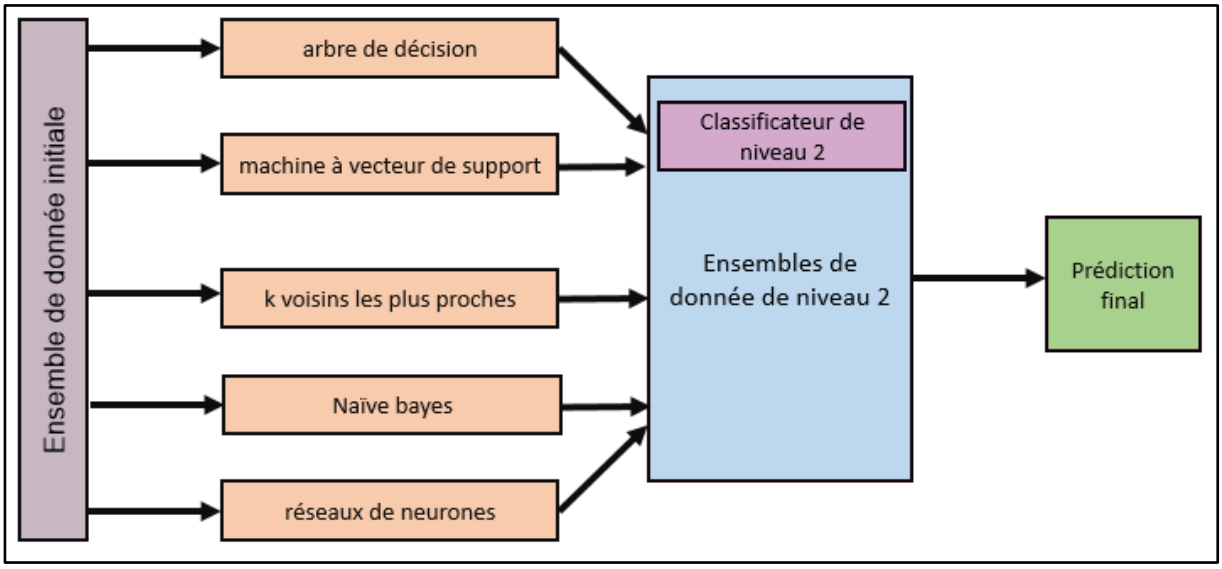

Figure 3. 2: L'architecture générale du modèle stacking utilisée dans notre projet

3.5 Expérimentations et analyse des résultats

3.5.1 Matériel utilisé

Ordinateur qui contient un processeur i5 de 4e génération et avec des spécifications résume dans table 3.1, nous avons fait 5 expérimentations différentes. Chaque expérience a une configuration de paramètre différente pour atteindre une performance maximale de chaque classifieur.

Processeur	Intel(R) Core (TM) i5-4210U 1.70GHZ – 2.40GHZ
Ram	8 GB
Disque dure	1 TB

Tableau 3.1 : spécification de l'ordinateur utilisé dans cette étude

3.5.2 Les bases de données utilisées

Divers domaines, notamment la recherche, les affaires, le marketing, les ventes, le développement de produits, l'éducation et la santé, utilisent des techniques de data mining. Dans le domaine médical le data mining permet des diagnostics plus précis. Disposer de toutes les informations sur le patient, telles que les dossiers médicaux, les examens physiques et les schémas de traitement, permets de prédire la présence d'une maladie spécifique et de prescrire des traitements plus efficaces. Il permet également de prévoir la durée d'hospitalisation. Cependant, pour utiliser data mining dans n'importe quel domaine, des bases de données doivent être

disponibles. C'est le principal facteur dans l'existence de l'idée de data mining. Nous avons choisi 2 bases de données synthétiques et 3 bases de données réelles, toutes les bases de données utilisées contiennent un seul attribut classe.

3.5.2.1 Base de données synthétique

Les données synthétiques comme leur nom indique, sont des données crées artificiellement plutôt que générées par des événements réels. Ils sont souvent créés à l'aide des algorithmes en utilisent ces données, nous avons besoin de ces données lorsque les exigences de confidentialité limitent la disponibilité des données ou lorsqu'on veut tester un produit à publier, mais ces données n'existent pas ou ne sont pas disponibles pour les testeurs. [51][52]

3.5.2.2 Base de données réelle

Les données réelles désignent les données d'un système de production, d'un fournisseur ou d'archives publiques, ou tout autre ensemble de données contenant autrement des données opérationnelles. Par exemple, un ensemble de données qui est une sauvegarde vieille de dix ans d'un système existant et qui contient des données sur de vrais individus, sujets ou cas, serait des données réelles. Un ensemble d'enregistrements publics acheté auprès d'un fournisseur pour être utilisé dans des tests constituerait également des données réelles. [53].

3.5.2.3 Description des bases de données utilisées

- **Obesity-level-indicators**

Cette base de données comprend des données servant à estimer les taux d'obésité chez les individus au Mexique, au Pérou et en Colombie. [54] [55].

Attribut	Nom d'attribut	Type	Nombre de valeur distinct
1	Sexe	Nominal	2
2	Age	Numérique	1402
3	Hauteur	Numérique	1574
4	Masse	Numérique	1525
5	histoire_familiale_avec_surpoids	Nominal	2
6	FAVC (aliments fréquents riches en calories)	Nominal	2
7	FCVC (quantité de légumes par repas)	Numérique	810
8	NCP (combien de repas principaux par jour)	Numérique	635
9	CAEC	Nominal	4
10	FUMEE (fumer)	Nominal	2
11	CH2O (combien d'eau quelqu'un boit)	Numérique	1268
12	SCC (surveillance quotidienne des calories)	Nominal	2
13	FAF (activité physique)	Numérique	1190
14	AUT (temps passé sur les appareils technologiques)	Numérique	1129
15	CALC (fréquence de consommation d'alcool)	Nominal	4
16	MTRANS (méthode de transport)	Nominal	5
17	NObeyesdad	Nominal	7

Tableau 3.2 : informations sur les attributs de la base de données « Obesity-level-indicators »

- **Asia data_set**

Base de données synthétiques pour les maladies pulmonaires [72].

Attribut	Nom d'attribut	Type	Nombre de valeur distinct
1	D (dyspnée), binaire 1/0 correspondant à "oui" et "non"	Nominal	2
2	T (tuberculose), binaire 1/0 correspondant à "oui" et "non"	Nominal	2

3	L (cancer du poumon), binaire 1/0 correspondant à "oui" et "non	Nominal	2
4	B (bronchite), binaire 1/0 correspondant à "oui" et "non"	Nominal	2
5	A (visite en Asie), binaire 1/0 correspondant à "oui" et "non	Nominal	2
6	S (fumer), binaire 1/0 correspondant à "oui" et "non"	Nominal	2
7	X (radiographie du thorax), binaire 1/0 correspondant à "oui" et "non"	Nominal	2
8	E (tuberculose versus cancer du poumon), binaire 1/0 correspondant à "oui" et "non	Nominal	2

Tableau 3.3 : informations sur les attributs de la base de données « Asia data set »

- **Diabètes Health Indicators Dataset**

Base de données sur les indicateurs du diabète.[59]

Attribut	Nom d'attribut	Type	Nombre de valeur distinct
1	HighBp	Numérique	2
2	Highchol	Numérique	2
3	cholcheck	Numérique	2
4	BMI	Numérique	66
5	smoker	Numérique	2
6	stroke	Numérique	2
7	HeartDiseaseorAttack	Numérique	2
8	physactivity	Numérique	2
9	fruits	Numérique	2
10	veggies	Numérique	2
11	hvyalcoholconsump	Numérique	2
12	Anyhealthcare	Numérique	2
13	Nodocbccost	Numérique	2
14	Genhlth	Numérique	5
15	Menhlth	Numérique	31
16	Physhlth	Numérique	30
17	diffwalk	Numérique	2
18	sex	Numérique	2
19	age	Numérique	13
20	education	Numérique	6
21	income	Numérique	8
22	Diabetes_binary	Nominal	2

Tableau 3.4 : informations sur les attributs de la base de données « Diabètes HealthIndicatorsDataset »

- **Heart Failure Prediction Data set**

Base de données des maladies cardiovasculaires.[60]

Attribut	Nom d'attribut	Type	Nombre de valeur distinct
1	Sexe	Nominal	2
2	Age	Numérique	50
3	chaistPainType	Nominal	4
4	RestingBp	Numérique	67
5	cholesterol	Numérique	222
6	FastingBs	Numérique	2
7	RestingECG	Nominal	3
8	MaxHR	Numérique	119
9	ExerciceAngina	Nominal	2
10	OldPeak	Numérique	53
11	ST_Slope	Nominal	3
12	HeartDisease	Nominal	2

Tableau 3.5 : informations sur les attributs de la base de données « Heart Failure PredictionDataset »

- **EEG Eye State Data Set**

Base de données sur l'état des yeux par EEG électroencéphalogramme. [56].

Attribut	Nom d'attribut	Type	Nombre de valeur distinct
1	AF3	Numérique	548
2	F7	Numérique	452
3	F3	Numérique	345
4	FC5	Numérique	312
5	T7	Numérique	285
6	P7	Numérique	330
7	O1	Numérique	290
8	O2	Numérique	294
9	P8	Numérique	304
10	T8	Numérique	346
11	FC6	Numérique	419
12	F4	Numérique	343
13	F8	Numérique	585
14	AF4	Numérique	592
15	eyeDetection	Nominal	2

Tableau 3.6 : informations sur les attributs de la base de données « EEG Eye State Data Set»

3.5.3 Les métriques

3.5.3.1 Matrice de confusion

Une matrice de confusion est un résumé des résultats de prédictions sur un problème de classification. Les prédictions correctes et incorrectes sont mises en lumière et réparties par classe. Les résultats sont ainsi comparés avec les valeurs réelles. Cette matrice permet de comprendre de quelle façon le modèle de classification est confus lorsqu'il effectue des prédictions. Ceci permet non seulement de savoir quelles sont les erreurs commises, mais surtout le type d'erreur commises. Les utilisateurs peuvent les analyser pour déterminer quels résultats indiquent comment les erreurs sont commises. Outre la machine learning, les matrices de confusion sont aussi utilisées dans le domaine des statistiques, du data mining et de l'intelligence artificielle. De manière générale, elles permettent d'analyser des données statistiques plus rapidement et de rendre les résultats plus simples à déchiffrer via la visualisation des données. Elles offrent l'opportunité d'analyser les erreurs dans les statistiques, le forage de données, ou même les examens médicaux.

Pour bien comprendre le fonctionnement d'une matrice de confusion, il convient de bien comprendre les quatre terminologies principales : TP, TN, FP et FN. Voici la définition précise de chacun de ces termes :

- **TP (True Positives)** : les cas où la prédiction est positive, et où la valeur réelle est effectivement positive.

- **TN (True Négative)** : les cas où la prédiction est négative, et où la valeur réelle est effectivement négative.

- **FP (False Positive) ou Type 1 Error** : les cas où la prédiction est positive, mais où la valeur réelle est négative.

- **FN (False Négative) ou Type 2 Error** : les cas où la prédiction est négative, mais où la valeur réelle est positive. [57].

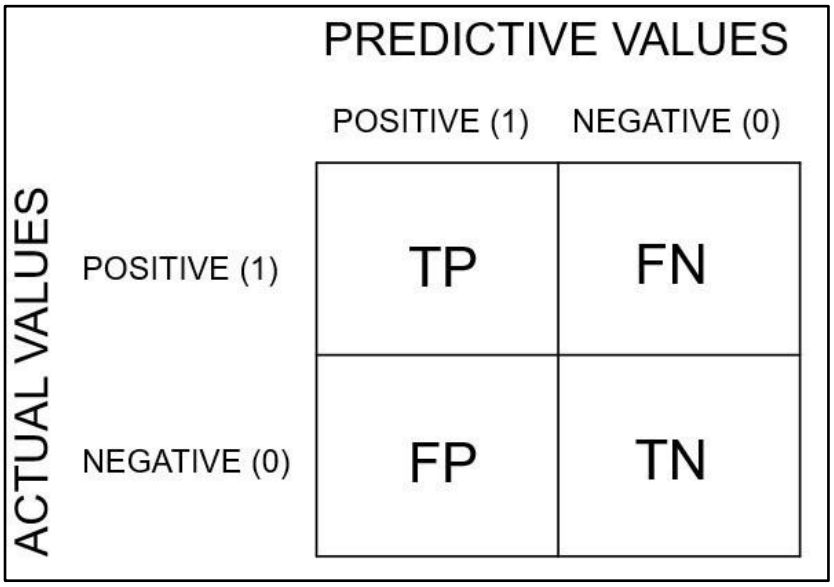

Figure 3. 3: matrice de confusion

3.5.3.2 Accuracy

La précision d'un algorithme de classification d'apprentissage automatique est un moyen de mesurer la fréquence à laquelle l'algorithme classe correctement un point de données. La précision est le nombre de points de données correctement prédits sur tous les points de données. Plus

formellement, il est défini comme le nombre de vrais positifs et de vrais négatifs divisé par le nombre de vrais positifs, de vrais négatifs, de faux positifs et de faux négatifs. Un vrai positif ou un vrai négatif est un point de données que l'algorithme a correctement classé comme vrai ou faux, respectivement. Un faux positif ou un faux négatif, en revanche, est un point de données que l'algorithme a classé de manière incorrecte.

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

3.5.3.3 Précision

La précision est définie comme le rapport entre le nombre d'enregistrements positifs correctement classés (vrais positifs) et le nombre total d'enregistrements positifs classés (correctement ou incorrectement).

En anglais on distingue "précision" et "accuracy" (la proportion des points correctement prédits) en français il n'y a pas de bonne traduction qui différencie les deux.

$$precision = \frac{TP}{TP + FP}$$

3.5.3.4 Rappel

Le rappel est calculé comme le rapport entre le nombre d'enregistrements positifs correctement classé comme positifs et le nombre total d'enregistrements positifs. Le rappel mesure la capacité du modèle à détecter les enregistrements positifs. Plus le rappel est élevé, plus les enregistrements positifs détectés sont nombreux. [58]

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

3.5.4 Les expérimentations

Dans toutes les expérimentations, nous avons validé les modèles obtenus avec une validation croisée avec 5 sous-ensembles, et les paramètres sont ajustés après plusieurs expérimentations pour chaque classifieurs de base.

3.5.4.1 Expérimentation 1

dans la première expérimentation, nous avons appliqué les classifieurs sur la première base de données " Obesity level indicators ", la configuration des paramètres est résumée dans le tableau 3.7 et les résultats résumés dans le tableau 3.8,nous voyons que les classifieurs de base donnent des bons résultats en général compte tenu de la métrique accuracy le réseau de neurones donne

95,49 % la meilleure performance parmi tous les classifieurs de base, mais notre modèle stacking a une meilleure accuracy avec 96,63 % en ce qui concerne les deux autres métriques (rappel et précision) il donne également le meilleur résultat parmi tous les classifieurs de base avec 0,966 dans les deux précision et rappel.

Et comme prévu, stacking prend beaucoup de temps pour construire son modèle environ 6 min juste pour construire le modèle une énorme différence par rapport aux classifieurs de base qui ne prennent que quelques secondes pour construire le modèle.

SVM		KNN		NB		AD	
TdK	Polynomial	K	1	Discrétisation	oui	Confidence	0.25
Cost	10.0	distance	euclidienne				
Gamma	--						
RN		Stacking					
Taux App	0.3	Cl Nv 2	SVM				
Epoque	500	NB Blocs	10				
Nb CC	default	Threads	4				

Tableau 3.7 : configuration des paramètres pour expérimentation 1

classifieurs	Accuracy %	Précision	Rappel	Temps (s)
SVM	95.07	0.951	0.951	27.24
KNN	81.05	0.805	0.811	0.015
NB	78.82	0.802	0.788	0.047
AD	93.22	0.933	0.932	0.062
RN	95.49	0.955	0.955	22.12
Stacking	96.63	0.966	0.966	330.42

Tableau 3.8 : résultats des classifications de base de données « Obesity-level-indicators »

3.5.4.2 Expérimentation 2

dans la deuxième expérimentation, nous avons appliqué les classifieurs avec une configuration des paramètres résument dans le tableau 3.9 sur la base de données « Asia data set », c'est une base de données des maladies pulmonaires, les résultats résument dans le tableau 3.10notez que tous les classifieurs de base donnent un résultat similaire en termes accuracy 84,04 % ,précision 0.848 et rappel 0.840, même avec notre technique stacking le résultat ne change pas si on regarde la matrice de confusion de tous les classifieurs on voit que le même nombre d'enregistrements 4202 sur 5000 a été correctement classifié et que pour tous les classifieurs les enregistrements restantes 798 étaient difficiles à classer même avec la classification ensembliste dans cette expérience, il est évident que les classifieurs de base sont meilleurs que le classifieur stacking car ils sont moins complexes et beaucoup plus rapides que stacking.

SVM		KNN		NB		AD	
TdK	Sigmoïde	K	3	Discrétisation	non	Confidence	0.25
Cost	10.0	distance	euclidienne				
Gamma	--						

RN		Stacking	
Taux App	0.3	Cl Nv 2	NB
époque	500	NB Blocs	10
Nb CC	2	Threads	4

Tableau 3.9 : configuration des paramètres pour expérimentation 2

classifieurs	Accuracy %	Précision	Rappel	temps (s)
SVM	83.96	0.847	0.840	0.812
KNN	83.88	0.846	0.839	0.001
NB	84.04	0.848	0.840	0.015
AD	84.04	0.848	0.840	0.031
RN	84.04	0.848	0.840	5.250
stacking	84.04	0.848	0.840	58.067

Tableau 3.10 : résultats des classifications de base de données « Asia data_set »

3.5.4.3 Expérimentation 3

Avec la configuration des paramètres résumés dans le tableau 3.11, nous avons appliqué les classifieurs sur la base de données « indicateurs de diabète » le résultat résumé dans le tableau 3.12, nous voyons que le résultat concernant accuracy est supérieur au 70% en général. Support vecteur machine 72,92% et le réseau de neurones 72,75%donnent les meilleurs résultats parmi tous les classifieurs de base, stacking les surpasses avec 73,32% même dans la précision et le rappel mais avec un coût énorme en temps.

SVM		KNN		NB		AD	
TdK	radial	K	7	Discrétisation	oui	Confidence	0.25
Cost	10.0	distance	euclidienne				
Gamma	0.01						

RN		Stacking	
Taux App	0.3	Cl Nv 2	SVM
Epoque	500	NB Blocs	10
Nb CC	2	Threads	4

Tableau 3.11 : configuration des paramètres pour expérimentation 3

classifieurs	Accuracy %	Précision	rappel	Temps(s)
SVM	72.92	0.730	0.72	18.13
KNN	70.04	0.701	0.700	0.016
NB	72.24	0.723	0.722	0.17
AD	69.90	0.699	0.699	1.36
RN	72.75	0.731	0.728	23.10
stacking	73.32	0.736	0.733	724.4

Tableau 3.12 : résultats des classifications de base de données « Diabètes Health Indicators Data set »

3.5.4.4 Expérimentation 4

La configuration des paramètres est résumée dans le tableau 3.13 et les résultats obtenus sont résumés dans le tableau 3.14.

Noter que l'arbre de décision donne un bon résultat par rapport aux classifieurs de base mais stacking donne les meilleurs résultats en termes accuracy 87.90 % rappel 0.80 et précision 0.879 mais prend plus de temps pour construire le modèle.

SVM		KNN		NB		AD	
TdK	radial	K	1	Discrétisation	false	Confidence	0.25
Cost	1.0	distance	euclidienne				
Gamma	0.01						

RN		Stacking	
Taux App	0.3	Cl Nv 2	SVM
Epoque	500	NB Blocs	10
Nb CC	default	Threads	4

Tableau 3.13 : configuration des paramètres pour expérimentation 4

classifieurs	Accuracy %	Précision	Rappel	Temps (s)
SVM	67.38	0.678	0.679	0.328
KNN	83.55	0.836	0.836	0.001
NB	84.96	0.850	0.850	0.031
AD	86.92	0.869	0.869	0.031
RN	81.26	0.813	0.813	3.92
stacking	87.90	0.880	0.879	39.67

Tableau 3.14 : résultats des classifications de base de données « Heart Failure PredictionDataset »

3.5.4.5 Expérimentation 5

Les paramètres utilisés sont résumés dans le tableau 3.15 et les résultats obtenus sont résumés dans le tableau 3.16, une autrefois l'arbre de décision donne un bon résultat par rapport aux classifieurs de base avec accuarcy de 84.52 % mais stacking le surpasse avec une accuarcy de 86.44 % le temps pris par ce classifieur est très énorme 36 min pour construire le modèle.

SVM		KNN		NB		AD	
TdK	linéaire	K	5	Discrétisation	oui	Confidence	0.25
Cost	1.0	distance	euclidienne				
Gamma	0.0						
RN		Stacking					
Taux App	0.3	Cl Nv 2	NN				
Epoque	500	NB Blocs	10				
Nb CC	default	Threads	4				

Tableau 3.15 : configuration des paramètres pour expérimentation 5

classifieurs	Accuracy %	Précision	Rappel	Temps(s)
SVM	63.86	0.637	0.639	45.6
KNN	83.86	0.838	0.839	0.016
NB	68.26	0.682	0.683	0.17
AD	84.52	0.845	0.845	1.234
RN	54.56	0.533	0.546	40.755
Stacking	86.44	0.865	0.864	2077.36

Tableau 3.16 : résultats des classifications de base de données « EEG Eye State Data Set »

3.6 Analyse des résultats

D’après les résultats on voit qu'il y a une relation entre les résultats obtenus et la structure des bases de données , dans la seconde expérience on a vu que stacking ne donne pas un bon résultat par rapport au coût du temps qu'il a fallu, la base de données utilisée dans la deuxième expérience "Asia dataset" est une base de données très simple, elle a 7 attributs chacun à deux valeurs étiquetées (oui ou non) et cela conduit les classifieurs à faire les mêmes erreurs dans cette base de données il n'y a pas des nouvelles informations apprises dans l'apprentissage donc si nous combinons les prédictions avec la méthode stacking, nous n'obtenons aucune amélioration des performances dans les meilleurs cas nous n'obtenons le même résultat du meilleur classifieur de base.

Notez que toutes les autres bases de données utilisées sont des ensembles de données qui ont beaucoup d'instances et certains ensembles de données contiennent plus de 20 attributs avec beaucoup de variables distinctes de type numérique, cette structure conduit les classifieurs de base à ne pas généraliser tous les cas et faire des erreurs différentes et apprendre différentes informations. Dans tous les autres expériences, stacking fonctionnent bien et donne les meilleurs résultats en d’autre termes il est stable par rapport aux classifieurs de base figure 3.4, la méthode d'ensemble c'est un moyen d'améliorer la précision et que ce que nous avons fait prouve, cette amélioration est due à la diversité de l'apprentissage, ces classifieurs apprennent de différentes manières et chaque classifieur de base apprend des nouvelles informations que l'autre manque, en donnant un exemple l’arbre de décision apprend de manière différente que le réseau de neurones le fait si le réseau de neurones a manqué des informations peut-être l'arbre de décision a apprenez ces informations manquantes et vice-versa donc si nous combinons tous les résultats dans une technique d'ensemble nous obtenons un meilleur résultat. C’est le principal avantage de la technique stacking.

Il est évident que stacking est plus complexe que tous les classifieurs de base utilisés, la complexité de SVM est o(n^3), réseau de neurones o(n^2), arbre de décision o(n log (n)) , KNN o(1), naïve bayes (n*d*c) où n = nombre de points de données , d = nombre d'attributs , c = nombre de classes , la complexité de stacking est égale au classifieur de base le plus complexe. Cette méthode d'ensemble prend un temps énorme par rapport aux classifieurs de base pour créer son modèle c'est le principal inconvénient, mais notez que dans les domaines médicaux le temps de création des modèles n'est pas important le plus important c’est la précision et la fiabilité des modèles.

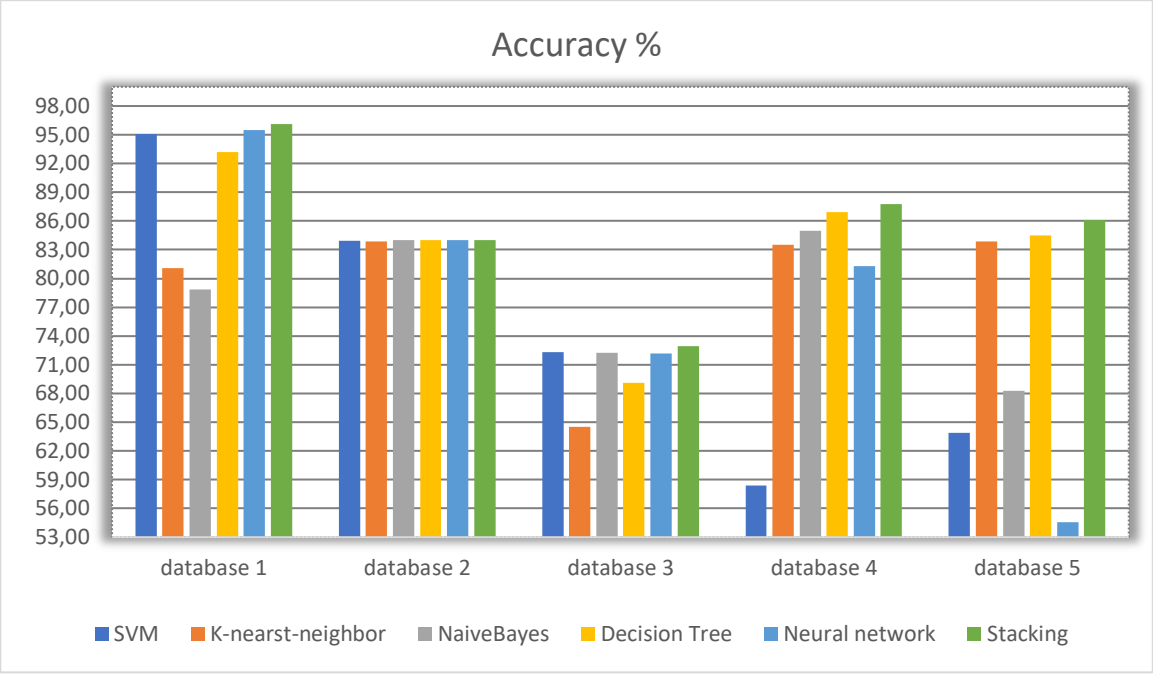

Figure 3. 4: Histogramme montrant les résultats obtenus dans les 5 expérimentations (accuracy %)

3.7 Conclusion

Dans ce chapitre, nous avons implémenté la méthode ensembliste stacking basée sur cinq algorithmes de classification de base à savoir : k plus proche voisin, arbre de décision, réseaux de neurones, machine a vecteurs de support, naïve bayes. Après plusieurs expérimentations, nous avons fixé les paramètres initiaux de chaque algorithme de base. Pour l'évaluation des résultats nous avons utilisé la technique de validation croisée et plusieurs métriques de performances sur des bases de données médicales. Les résultats montrent que notre méthode ensembliste améliore considérablement la précision de classification surtout dans les bases de données volumineuses, alors que la précision de la classification s'améliore légèrement pour les bases de données qui contiennent peu d'enregistrements et peu d'attributs, il est important de noter que les méthodes ensemblistes sont gourmandes en temps de calcul ce qui nous a poussé à utiliser le calcul parallèle (thread) pour réduire ce dernier.

Conclusion générale

La classification est une méthode très importante dans le data mining, et qui consomme beaucoup de recherches pour son optimisation. La majorité des algorithmes de classification de base souffrent des problèmes de précisions et de stabilités des résultats sur quelques types de données.

L'objectif de ce projet est de présenter une étude comparative entre les algorithmes de classification de base et d'implémenter une méthode ensembliste stacking pour résoudre le problème de précision et de stabilité des résultats.

Pour valider notre travail, nous avons comparé les algorithmes de classification classiques déjà implémentés dans la bibliothèque Weka et la méthode ensembliste stacking sur des bases de données médicales synthétiques et réelles.

A travers les différentes expérimentations réalisées, nous avons constaté une nette amélioration de la précision surtout dans les bases de données volumineuses.

Suite à réalisation de notre étude, il serait intéressant de proposer des solutions automatiques pour l'ajustement des paramètres initiaux de certains algorithmes. Réduire le temps de calcul des méthodes ensemblistes à travers le calcul parallèle et distribué.

Bibliograghie

[1] **David Hand, Heikki Mannila and Padhraic Smyth**, livre intitule: “Principles of Data Mining” 2001.

[8] **Michael J.A. Berry Gordon S. Lino** livre intitule : “Data Mining Techniques for Marketing, Sales, and Customer Relationship Management Second Edition”. 2004.

[10] **daniel t .larose**, livre intitule: "discovering knowledge in data an Introduction to Data Mining".2005

[21] **Charu C. Aggarwal** livre intitule : “Data Mining the Textbook” 2015.

[24] **J. Han and M. Kamber**, livre intitule : “Data Mining Concepts and Techniques”, Elevier, 2011.

[71] **Charu C.Aggrawal**, livre intitule : Data Classification Algorithms and Applications, 2015

[73] **Kevin P. Murphy** livre intitule : Machine Learning A Probabilistic Perspective : 2012

[4] **Usama. F, Gregory. P, and Padhraic** dans article "From Data Mining to Knowledge Discovery in Databases",1996.

[9] **Bharati M. RamageriIndian**, Indian Journal of Computer Science and Engineering dans article “data mining techniques and applications”,2010.

[11] **Kesavaraj G, Sukumaran S. A** dans article : “study on classification techniques in data mining. in Computing, Communications and Networking Technologies“ , 2013.

[12**] j.rquinlan** Centre for Advanced Computing Sciences, dans article :“Induction of Decision Trees “.1986.

[13] **Badr h, Abdelkarim M, Hanane Mohammed E**, dans article “A comparative study of decision tree ID3 and C4.5” .2014.

[14] **Marina Milanović** Faculty of Economics, University of Nis, Serbia, dans article : “chaid decision tree: methodological frame and application”.2016.

[15] **Kashvi.T, Sanjukta.DSrishti.V**, article “machine learning classification with k-nearest neighbors”.2016

[16] **Arun K. , Lawrence D. , Andreas B, and JunhuiCai** , dans article : ‘All of Linear Regression’. 2019.

[17] **Kristina P. Sinaga and Miin-Shen Yang**, dans article : "Unsupervised K-Means Clustering Algorithm».2020.

[18**] Mohammed Al-Maolegi , Bassam Arkok**, dans article : "an improved apriori algorithm for association rules" International Journal on Natural Language Computing (IJNLC) .2014.

[22] **Osmar R. Zaiane** ,dans article : Principles of Knowledge Discovery in Databases, CMPUT690, University of Alberta, 1999.

[25] **S. B. Kotsiantis**, dans article “Supervised Machine Learning: A Riview of Classification Techniques” 2007.

[27] **Thair N. Phyu**, dans article “Survey of Classification techniques in Data Mining”, in International Multiconference of Engineers and Computer Scientists, Hong Kong, 2009.

[28] **Bhavsar, Hetal , Ganatra, Amit**, dans article “A Comparative Study of Training Algorithms for Supervised Machine Learning”, International Journal of Soft Computing and Engineering (IJSCE) ,2012.

[29] **Duda R, Hart P**, dans article : "Pattern Classification and Scene Analysis," John Wiley and Sons, 1973.

[30] **Friedman, N., Geiger, D., Goldazmidt**, dans article : Bayesian Network Classifiers, 1997.

[31] **K. P. Soman**, dans article : “Insight into Data Mining Theory and Practice”, 2006.

[32] **Cover, T., Hart**, dans article : « Nearest Neighbor Pattern Classification », IEEE Transactions on Information Theory, 1967.

[35] Design Goals of the Java ” Programming Language". Oracle. 1999.

[38] **G. Holmes, A. Donkin and I.H. Witten**, dans article : « Weka: A machine learning workbench » Conference on Intelligent Information Systems, Brisbane, Australia, 1994 .

[40] **d. aha and d. kibler**, article : “ instance-based learning algorithms “,1991.

[41] **Ms S. Vijayarani1 , Ms M. Muthulakshmi**. Dans article “Comparative Analysis of Bayes and Lazy Classification Algorithms“2013.

[42] **Tina R. Patil, Mrs. S. S. Sherekar**, dans article “Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification” 2013.

[43] **Ying Yang · Geoffrey I.**, dans article “Discretization for Naive-Bayes learning: managing discretization bias and variance.”,2008.

[46] N. Saravanan V. Gayathri article : ”Performance and Classification Evaluation of J48 Algorithm and Kendall’s Based J48 Algorithm (KNJ48).”2018.

[47] **Chang, C. C., Lin, c. J**. Dans article : “LIBSVM: A library for support vector machines. ACM Trans, on Intelligent System and Technology, (2011).”

[48] **Petro Liashchynskyi Pavlo Liashchynskyi,** dans article :"Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS" ,2019.

[49] **Hassan Ramchoun, Mohammed Amine JanatiIdrissi, Youssef Ghanou, Mohamed Ettaouil**, dans article "Multilayer Perceptron: ArchitectureOptimization and Training".2016.

[50] **Kun-Cheng Ke Ming-Shyan Huang**, dans article : "Enhancement of Multilayer Perceptron Model Training Accuracy through the Optimization of Hyper parameters: A Case Study of the Quality Prediction of Injection Molded Parts " 2021.

[51] **Stefanie Koperniak**, dans article : “Artificial data give the same results as real data — without compromising privacy” 2017.

[54] **Palechor, F. M., & de la HozManotas, A**. Dans article « Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico ».2019.

[34] **zine-el-abidinesoudani** , « étude comparative des algorithmes dédiés a la classification »,mémoire : mathématiques et informatique appliquées , l'université du Québec à Trois-Rivières ,2005.

[2] attribute meaning Consultable à l’adressehttps://www.statistics.com/ ,consulté le [29-05-2022].

[3] What is KDD Consultable à l’adresse : https://www.tutorialspoint.com/what-is-kdd. Consulté le [11-04-2022].

[5] Difference Between Data Mining Supervised and Unsupervised Consultable à l’adresse : http://www.differencebetween.net/, consulté le [15-03-2022]

[6] what’s the difference between supervised and unsupervised Consultable à l’adresse : https://dataconomy.com/,consulté le [15-03-2022]

[7] wide skills data mining tutorial Consultable à l’adresse : https://www.wideskills.com. Consulté le [05-04-2022]

[19] **shivam .A** « Data Mining Vs. Machine Learning : The Key Difference" Consultable à l'adresse: https://www.simplilearn.com/data-mining-vs-machine-learning-article, Consulté le [09-04-2022].

[20] **kechit. G** Data Mining vs Machine Learning : Major Differences consultable sur : https://www.upgrad.com/blog/data-mining-vs-machine-learning ,consulté [5-02-2022].

[33] SupportVector Machine Algorithm , India , [consulter le 02/05/2022],Consultable à l'adresse :https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm, Consulté le [02-05-2022]

[36] python vs java Consultable à l'adresse :https://www.snaplogic.com/glossary/python-vs-java-performance. Consulté le [05-05-2022]

[37] Pour quoi les data scientistes preferment Python à Java Consultable à l'adresse : https://analyticsindiamag.com/why-do-data-scientists-prefer-python-over-java/ Consulté le [15-05-2022]

[39] « cross validation » Consultable à l'adresse : https://datascientest.com/glossary/validation-croisee-cross-validation. Consulté le [09-04-2022]

[44] J48 Classification (C4.5 Algorithm) in a Nutshell Consultable à l'adresse : https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e . Consulté le [25-04-2022]

[45] Paramètres de classifieur J48, Consultable à l'adresse : https://www.schankacademy.com/demos/data-analytics/xt/lib/docs/0/j48_parameters.pdf. Consulté le [29-03-2022].

[52] Cemdilmegani "The Ultimate Guide to Synthetic Data : Uses, Benefits & Tools" Consultable à l'adresse : https://research.aimultiple.com/synthetic-data/ Consulté le [17-03-2022]

[53] "real data definition" Consultable à l'adresse: https://www.lawinsider.com/dictionary/real-data. Consulté le [06-04-2022]

[55] « Estimation of obesity levels based on eating habits and physical condition Data Set » Consultable à l'adresse : https://archive.ics.uci.edu/ml/datasets Consulté le [03-05-2022]

[56] "EEG eye state data set " Consultable à l'adresse : https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State# Consulté le [03-05-2022]

[57] « Confusion Matrix : l'outil de mesure de performances du Machine Learning" » Consultable à l'adresse : https://www.lebigdata.fr/confusion-matrix-definition Consulté le [04-05-2022]

[58] « Confusion Matrix » Consultable sur le site : www2.cs.uregina.ca Consulté le [03-05-2022]

[59] « Diabetes Health Indicators Dataset » Consultable à l'adresse : https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset. Consulté le [03-05-2022]

[60] « Heart Failure Prediction Dataset » Consultable à l'adresse https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction. Consulté le [03-05-2022]

[61] Extraction des connaissances consultable à l'adresse : https://extractiondesconnaissances.wordpress.com/tag/ecd/. Consulté le [06-05-2022]

[62] KNN Classification Tutorial using Scikit-learn consultable à l'adresse :https://www.datacamp.com/tutorial/k-nearest-neighbor-classification.Consulté le [13-04-2022]

[63] how to implement linear regression for machine learning consultable à l'adresse : https://www.edureka.co/blog/linear-regression-for-machine-learning/#linear. Consulté le [03-05-2022]

[64] Means Clustering using Python consultable à l'adresse : https://medium.com/@luigi.fiori.lf0303/k-means-clustering-using-python-db57415d26e6 Consulté le [23-05-2022]

[65] Bagging algorithms in Python consultable à l'adresse : https://www.section.io/engineering-education/implementing-bagging-algorithms-in-python/. Consulté le [03-03-2022]

[66] Optimization of stacking ensemble configurations through Artificial Bee Colony algorithm consultable à l'adresse :

https://www.semanticscholar.org/paper/Optimization-of-stacking-ensemble-configurations-Shunmugapriya-Kanmani/c20ce426b2dee40b792dc7107459783e2d9a7870 Consulté le [03-03-2022]

[67] Les arbres de décisions consultable à l'adresse :

https://zestedesavoir.com/tutoriels/962/les-arbres-de-decisions/comprendre-le-concept/ Consulté le [23-02-2022]

[68] les réseaux de neurones consultables à l'adresse :

https://www.researchgate.net/figure/Le-perceptron-multicouches_fig6_30517821. Consulté le [23-02-2022]

[69] Démystifier le Machine Learning, Partie 2 : les Réseaux de Neurones artificiels, consultable à l'adresse :

https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-2-les-reseaux-de-neurones-artificiels Consulté le [23-02-2022]

[70] Web-based Classification Application for Forest Fire Data Using the Shiny Framework and the C5.0 Algorithm, consultable à l'adresse :

https://www.researchgate.net/publication/301317735_Web-based_Classification_Application_for_Forest_Fire_Data_Using_the_Shiny_Framework_and_the_C50_Algorithm. Consulté le [15-04-2022]

[72] Base de données "Asia data-set" consultable à l'adresse : https://www.openml.org/search?type=data&status=active&id=43151 . Consulté le [25-03-2022]