الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : IA-AP

Thème

Etude comparative et mise en œuvre des algorithmes du machine learning supervisé: pour diagnostic de l'hypothyroïdie

▪ **Présenté par :**

DJOUDI Houssem

▪ **Dirigé par :**

Dr. DERKAOUI Orkia

Année universitaire 2024-2025

REMERCIEMENTS

Je tiens à exprimer mes sincères remerciements à toutes les personnes qui ont contribué à la réalisation de ce travail.

En premier lieu, je remercie Allah le Tout-Puissant de m'avoir donné la force et la persévérance pour mener à bien ce projet.

Je voudrais adresser mes vifs remerciements à ***Me Derkaoui ORKIA*** *mon directeur de mémoire, pour ses conseils précieux, sa disponibilité et son encadrement tout au long de cette recherche.*

Mes remerciements s'adressent également aux membres du jury et à tous les enseignants qui ont contribué à ma formation.

Une pensée particulière va à ma famille qui m'a soutenu et encouragé tout au long de mon parcours universitaire.

Je n'oublie pas mes amis et collègues qui ont toujours été présents pour m'aider.

Merci à tous.

DEDICACES

Je dédie ce travail à la mémoire de mes grands parents paternelles ainsi que ma tante Asma, mon oncle Abdelkader et mon grand père Ahmed.

À ma mère, aucun hommage ne pourrait être à la hauteur de tous les sacrifices dont elle a fait preuve.

À ma très chère sœur Aya, je tiens à te remercier pour ton aide, ta patience et ton écoute.

À mes chers tantes et oncles, leurs époux et épouses, sans oublier ma cher grand-mère à qui je souhaite une longue vie.

À mon frère de cœur Kamel, à mes très chers camarades de promotion Majid, Aissa, Ahmed KOUIDRI.

À tous ceux qui on contribué de près ou de loin pour que ce projet soit possible, je vous remercie.

الملخص:

السياق والهدف

تهدف الدراسة إلى تحسين تشخيص **قصور الغدة الدرقية** (نقص هرمون الغدة الدرقية) باستخدام نماذج التعلّم الآلي. مجموعة البيانات المستخدمة) من UCI) تحتوي على **٣،٧٧٢عينة** مع **٣٠سمة سريرية**)اختبارات هرمونية، تاريخ طبي، إلخ.)

أفضل الخوارزميات

- **J48** و **Random Forest** تفوقتا على الخوارزميات الأخرى بسبب **دقتهما> ٩٩٪** و **متانتهما**.

الإستنتاجات الرئيسية:

- **الخوارزميات** J48 و Random Forest هما الأمثل لتصنيف قصور الغدة الدرقية.

المعالجة المسبقة للبيانات:

- ❖ استبدال القيم المفقودة وإزالة القيم الشاذة (IQR 3.0) أمران حاسمان.
- ❖ التقسيم الموجه (Supervised Discretization) يُحسّن استقرار النماذج.

القيود:

- طرق "الغلاف (Wrapper) "و"الترشيح (Filter) "لا تحسّن الأداء (السمات ذات صلة بالفعل).

Résume

Contexte et Objectif

- L'étude vise à optimiser le diagnostic de l'hypothyroïdie (déficit hormonal thyroïdien) via des modèles d'apprentissage automatique. Le dataset utilisé (provenant de l'UCI) contient **3 772 instances** avec **30 attributs** cliniques (tests hormonaux, antécédents médicaux, etc.).

Meilleurs algorithmes :

- **J48** et **Random Forest** surpassent les autres grâce à leur précision (>99%) et leur robustesse.

Conclusions Clés

1. **Algorithmes** : **J48** et **Random Forest** sont optimaux pour la classification de l'hypothyroïdie.
2. **Prétraitement** :
   - Remplacer les valeurs manquantes et supprimer les outliers (IQR 3.0) sont cruciaux.
   - La discrétisation supervisée améliore la stabilité des modèles.
3. **Limites** :
   - Les méthodes *Wrapper* et *Filter* n'améliorent pas les performances (attributs déjà pertinents).

Summary

Context and Objective

- The study aims to optimize the diagnosis of hypothyroidism (thyroid hormone deficiency) using machine learning models. The dataset used (from UCI) contains **3,772 instances** with **30 clinical attributes** (hormonal tests, medical history, etc.).

Best Algorithms

- **J48** and **Random Forest** outperform others due to their **accuracy (>99%)** and **robustness**.

Key Conclusions

1. **Algorithms**: J48 and Random Forest are optimal for hypothyroidism classification.
2. **Preprocessing**:
   - Replacing missing values and removing outliers (IQR 3.0) are critical.
   - Supervised discretization improves model stability.
3. **Limitations**:
   - Wrapper and Filter methods do not improve performance (attributes are already relevant).

Liste des figures

Les figures de chapitre I

Les figures de chapitre II

Les figures de chapitre III

Les figures de chapitre IV

Liste des tableaux

Les tableaux de chapitre I

Les tableaux de chapitre I

Liste d'abréviation

LR : Linéaire Regression

IA : Intelligence Artificielle

ML : Machine Learning

K-NN : K Nearest Neighbors

RF : Random Forest

SVM : Support Vector Machine

ML : Machine Learning

HTs : Hormones thyroïdiennes

TSH : Thydroïd Stimulating Hormone

TRH : Thyrotropin Releasing Hormone

T3 : 3, 5,3'-triiodothyronine,

T4 : 3, 5,3',5'-tétraïodothyronine

WEKA : Waikato environment for knowledge analysis

Table des matières

CHAPITRE I: Généralité Sur La Machine Learning

CHAPITRE II: L'Hypothyroïdie:Présentation De La maladie

CHAPITRE III: Initiation WEKA

CHAPITRE IV: Prétraitement Et Classification D'Hypothyroïdie

Introduction Générale

Introduction générale

Avec l'évolution des modes de vie et le vieillissement démographique, la prévalence des maladies chroniques connaît une augmentation significative. Parmi ces pathologies, l'hypothyroïdie représente un enjeu majeur de santé publique, affectant des millions de personnes mondialement. La prédiction précoce de ces affections par des méthodes scientifiques fondées sur l'analyse de facteurs cliniques, biologiques et environnementaux joue un rôle crucial dans l'amélioration de la prise en charge médicale, la réduction des coûts sanitaires et l'optimisation des stratégies de prévention.

Problématique

Ce mémoire aborde la problématique de la prédiction des risques de maladies chroniques, avec un focus sur l'hypothyroïdie. Pour résoudre cette tâche de classification médicale, nous développons trois modèles distincts :

1. Un modèle basé sur la technique d'apprentissage ensembliste *Bagging* (algorithme *Random Forest*)
2. Un second modèle utilisant la technique de *Boosting*
3. Un modèle fondé sur les réseaux de neurones, compte tenu de leur efficacité croissante dans le domaine médical.
   L'objectif est de concevoir des architectures robustes capables d'identifier les marqueurs prédictifs pour comparer in fine leurs performances.

Contribution

Notre recherche apporte trois contributions principales.

- Une synthèse des méthodes d'apprentissage automatique et des réseaux de neurones appliqués aux problèmes de classification médicale, notamment la prédiction de pathologies chroniques.
- La conception de trois pipelines de prédiction, incluant les stratégies de prétraitement des données cliniques spécifiques aux défis sanitaires.
- L'implémentation, l'entraînement et l'évaluation comparative des modèles sur des jeux de données biomédicaux, avec analyse critique de leur précision diagnostique.

Organisation du mémoire L'étude est structurée en 4 chapitres :

- CHAPITRE I : Généralité Sur La Machine Learning
- CHAPITRE II : L'Hypothyroïdie: Présentation De La maladie
- CHAPITRE III : Initiation WEKA
- CHAPITRE IV : Prétraitement Et Classification D'Hypothyroïdie

CHAPITRE I

Généralité Sur La Machine Learning

1. Introduction générale

Intelligence Artificielle (IA) est l'un des domaines les plus récents de la science et de l'ingénierie.

Les travaux ont sérieusement débute après la seconde guerre mondial, et le nom lui mémé a été inventé en 1956. Régulièrement cité comme "domaine ou j'aimerai bien y être" par les scientifique dans d'autres disciplines. Un étudiant en physique peut raisonnablement se dire que toutes les bonnes idées ont été trouvées par Gali lée, Newton, Einstein et le reste. De l'autre coté, tout reste ouvert en IA.

Historiquement, quatre approches de l'IA ont été suivies, chacune par des gens différents avec des méthodes différentes. Une approche axée sur l'homme doit être en partie une science empirique, impliquant observation et hypothèse sur le comportement humaine.une approche rationnelle implique une combinaison de mathématique et d'ingénierie [1].

Les différents groupes se sont décriés et se sont aides mutuellement, voici les quatre approches :

1) **Acting humanly:** le test de Turing "l" art de créer des machines qui exécutent des fonctions requérant une intelligence lorsqu'elles sont exécutées par des êtres humains".
2) **Thinking humanly:** la modélisation cognitive" L'excident nouveau défi de construire des ordinateurs qui pensent, des machines avec des consciences, au sens figurré".
3) **Thinking rationally:** les lois de la pensée "L'étude des calculs qui rendent possibles la perception,le rationne ment et les actes".
4) **Acting rationnaly:** les agents rationnels "L'IA es l'étude de la conception des agents intelligents" [2]

L'IA englobe plusieurs sous domaines allant du plus générale (apprentissage et perception) au plus spécifique, comme jouer aux échecs, démontrer des théorèmes mathématique, conduire une voiture ou diagnostique des maladies. L'IA se révèle être utile dans toutes les taches intellectuelles. C'est vraiment un domaine universel et pluri-disciplinaire.

Le but de la recherche en IA est de créer une technologie qui permette aux ordinateurs et aux machines de fonctionner d'une manière intelligente.

Le problème générale de la création d'une intelligence a été divisé en plusieurs sous problèmes. Celles-ci consistent en des capacités que les chercheurs espèrent qu'un système intelligent pourra exécuter [2].

Il existe plusieurs stratégies utilisées en IA ; entre autres l'Apprentissage automatique.

On peut citer trois types d'algorithme d'apprentissage automatique :

- Apprentissage Supervisé.
- Apprentissage non Supervisé.
- Apprentissage par Renforcement.

1.1. Problématique

La classification est un problème central de l'apprentissage automatique (ML) et de l'IA. Une règle de classification est une procédure permettant d'affecter à un objet l'étiquette du groupe auquel il appartient, autrement dit de le reconnaitre.

Nous allons nous intéresser à la problématique de la classification d'image qui est la tache d'attribuer à une image d'entrée X un label Y à partir d'un ensemble fixe de catégories. C'est l'un des problèmes fondamentaux de la vision par ordinateur. Nous allons choisir une basses de recherche (DataSet).Cette base se composé de 1222 petites images Gray scale, avec apprentissage supervisé et deux algorithmes les plus populaires en classification on veut utiliser et comparaison entre SVM et KNN.

2. Etat de l'art

2.1. Qu'est le machine learning ?

Le ML est une discipline de l'IA qui offre aux ordinateurs la possibilité d'apprendre à partir d'un ensemble d'observations que l'on appelle ensemble d'apprentissage.

Le Machine Learning (ML) à connu un essor de son utilisation et de son application à des problèmes d'automatisation dans divers domaines [3].

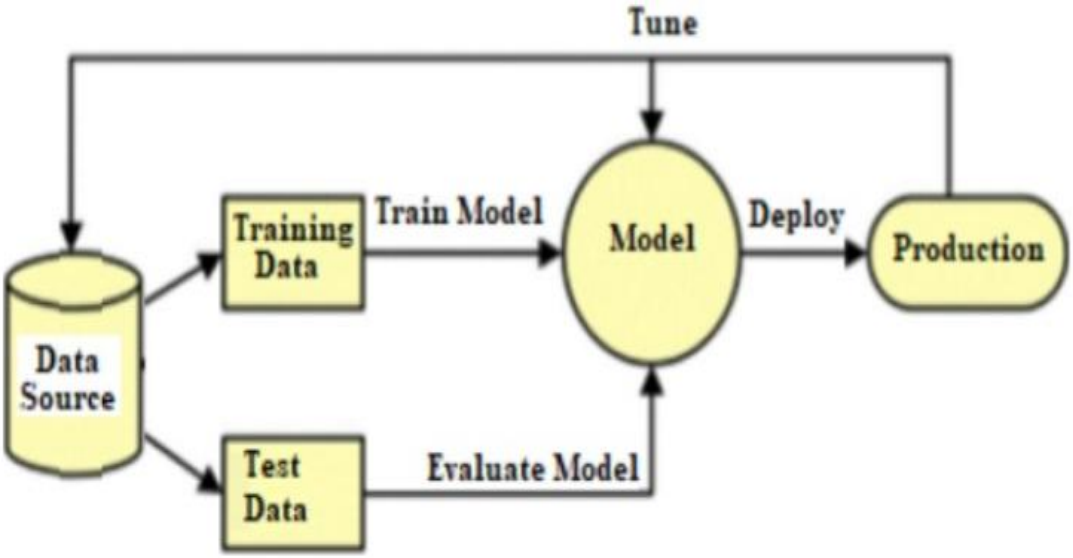


Figure (I.01): *Le processus typique du ML.*

2.2. Les différents types de Machine Learning

2.3. Apprentissage Supervisé et Non supervisé

- L'apprentissage **supervisé** est la tache d'apprentissage automatique la plus simple et la plus connue.
- L'apprentissage **Non supervisé** ou Clustering ne demande aucun étiquetage préalable des données. Le but est que le modèle réussisse à regrouper les observations disponibles en catégories par lui-même [4].

2.4. Régression et Classification

- Un modèle de Classification est un modèle de ML dont les sorties y appartiennent à un ensemble fini de valeurs (exemple :bon,myen,mauvais)[5]
- Un modèle de Régression est un modèle de ML dont les sorties y sont des nombres (exemple :la température de demain) [6]

2.5. Apprentissage par Renforcement

Apprentissage par Renforcement

Est un domaine de l'apprentissage automatique qui s'intéresse à la façon dont les agents logiciels doivent agir dans un environnement afin de maximiser une certaine notion de récompense cumulative [7].

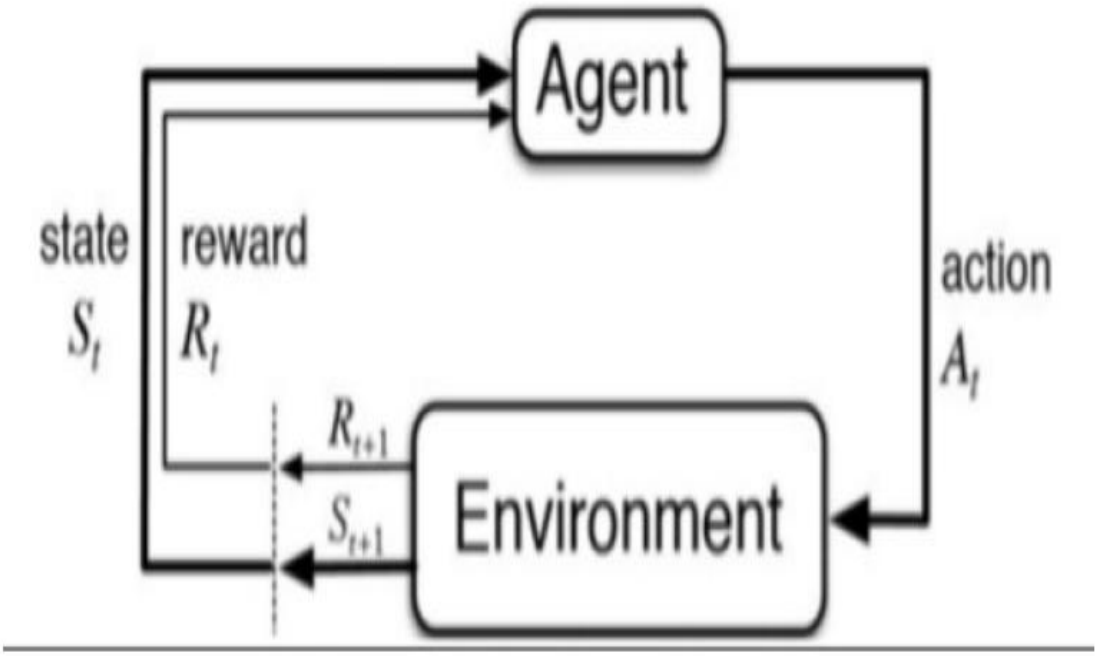


Figure (I.02): *Apprentissage par Renforcement*

2.6. Apprentissage Semi-Supervisé

L'apprentissage Semi-Supervisé est une combinaison de méthodes d'apprentissage automatique supervisé et non supervisé,il peut etre fructueux dans les domaines de l'apprentissage automatique et de l'exploration de données ou les données Non étiquetées son déjà présentes et ou l'obtention des données étiquetées est un processus fastidieux avec des méthodes d'apprentissage automatique plus courantes. vous formez un algorithme d'apprentissage automatique sur un ensemble de données étiqueté dans lequel ,chaque enregistrement comprend les infirmations sur les résultats[8].

3. Les différents types d'algorithme

3.1. La Régression Linéaire

- Une régression linéaire est un modèle de ML supervisé,avec x en entrée et y en sortie elle est de la forme [9].

$$y = ax + b \quad \textbf{(1)}$$

b et a sont des valeurs réales à apprendre. On définit par

$$f(x) = ax + b \quad \textbf{(2)}$$

1) Régression Linéaire Simple

Est l'une des techniques les plus utilisées dans l'apprentissage automatique et cela revient principalement à sa simplicité et la facilité d'interprétation de ses résultats. Comme on a pris l'habitude d'applique le modèle d'apprentissage sur un exemple pour le voir en action, on va procéder ainsi pour l'algorithme de régression linéaire simple. Tout d'abord, on importe le classe **LinearRegression** et définit les données *x* et *y* sur lesquelles le modèle va performer [9].

2) Régression Linéaire Multiple

Est une méthode de régression mathématique étendant la régression linéaire simple pour décrire les variations d'une variable endogène associée aux variations de plusieurs variables exogènes [10].

Avantages:

- Le modèle est facile à interpréter.

Inconvénients:

- Sensible aux bruits.

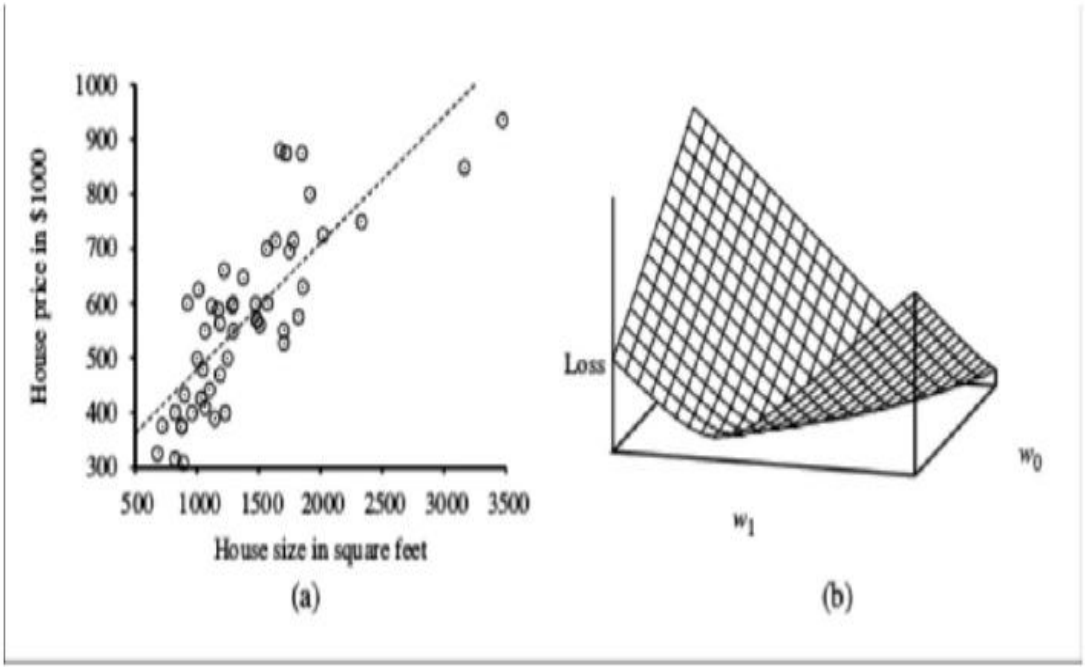


Figure (I.03): *(a) représente le prix des maisons en fonction de leur surface qui sont en vent à Berkeley. (b) montre un exemple d'ensemble d'apprentissage de n points sur le plan x,y*

- Négligence des interactions entre les variables prédictives.

3.2. Les k plus proches voisins

L'algorithme des K-Nearest Neighbors (KNN) (K plus proches voisins) est un algorithme de classification supervisé. Chaque observation de l'ensemble d'apprentissage est représentée par un point dans un espace à n dimensions ou n est le nombre de variables prédictives. Pour prédire la classe d'une observation, on cherche les k points les plus proches de cet exemple [11].

Avantages:

- Apprentissage rapide.
- Méthode facile à comprendre.
- Adapté aux domaines ou chaque classe est représentée par plusieurs prototypes et ou les frontières sont irrégulières.

Inconvénients:

- Prédiction lente car il faut revoir tous les exemples à chaque fois.
- Méthode gourmande en place mémoire.
- Sensible aux attributs non pertinents et corrélés.
- Particulièrement vulnérable au fléau de la dimensionnalité.

Pour *k*=3 la classe majoritaire du point central est la classe B, mais si change la valeur du voisinage *k*=6 la

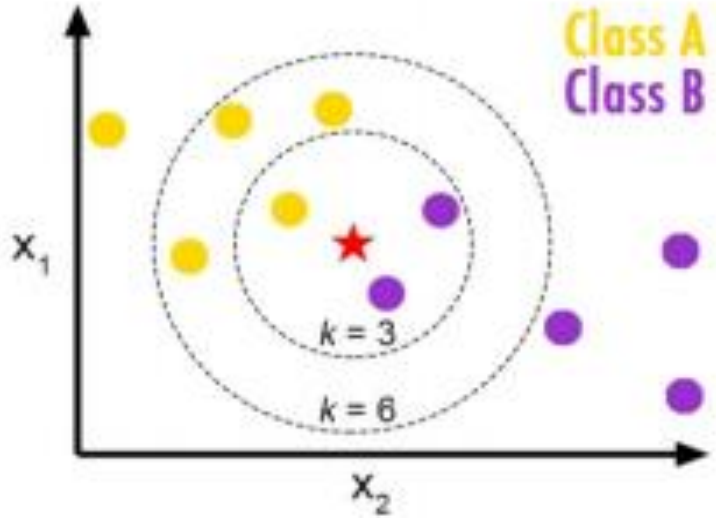


Figure (I.04): *KNN classe majoritaire*

Class majoritaire devient la classe A.

3.3. Le Classifieur Naïf de Bayes

Le **classifieur naïf de bayes** est un algorithme supervisé probabiliste que la présence d'une caractéristique particulière dans une classe n'est pas liée à la présence de toute autre caractéristique.

Il est principalement utilisé à des fins de regroupement et de classification dépend de la probabilité conditionnelle de se produire [12].

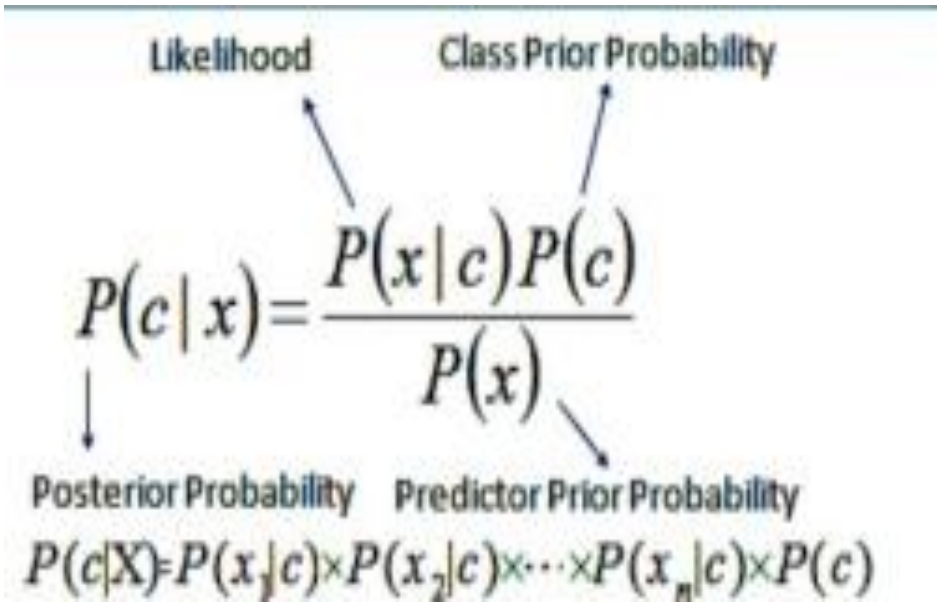


Figure (I.05): *Naive Bayes.*

Avantages:

- L'algorithme offre de performance.

Inconvénients:

- La prédiction devient erronée si l'hypothèse indépendance conditionnelle est invalide.

3.4. K-means

L'algorithme des K-moyennes (k-means) est un algorithme non Supervisé le plus simples qui résolvent le problème de clustering bien connu. La procédure suit un moyen simple et facile de classer un ensemble de données donné(Dataset) à travers un certain nombre de clusters. Les étapes d'algorithme sont [13]:

- Choisir k points qui représentent la position moyenne des clusters.
- répéter jusqu'à stabilisation des points centraux :
-affecter chacun des M points au plus proche des k points centraux.
- mettre à jour les points centraux en calculant les centres de gravité des k cluster.

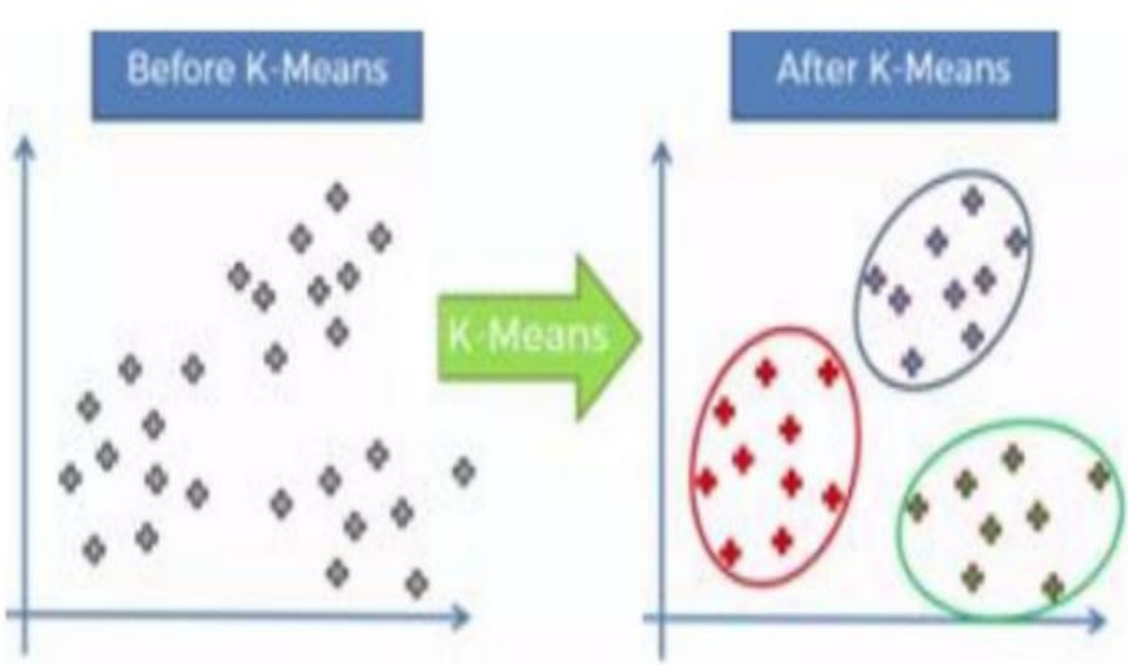


Figure (I.06): *K-means Clustering*

Avantages:

- Implementable pour des grands volumes de données.

Inconvénients:

- Le choix du paramètre K n'est pas découvert mais choisi par l'utilisateur.
- La solution dépend des K centre de gravité choisie lors de l'initialisation.

3.5. Les arbres de décision

Un arbre de décision (décision tree) est un algorithme d'apprentissage supervisé qui va permettre la prise de décision en prenant en entrée une population, un échantillon pour ensuite procéder à une catégorisation basée sur des facteurs discriminants. Cet outil va donc répartir les individus en groupes homogènes et va émettre des prédictions à partir de données connu [14].Un arbre de décision se présente comme sur la figure.

Avantages:

- Peu de préparation des données.

- Performance sur de grands jeux de données.

Inconvénients :

- L'existence d'un risque de sur-apprentissage si l'arbre devient très complexe.

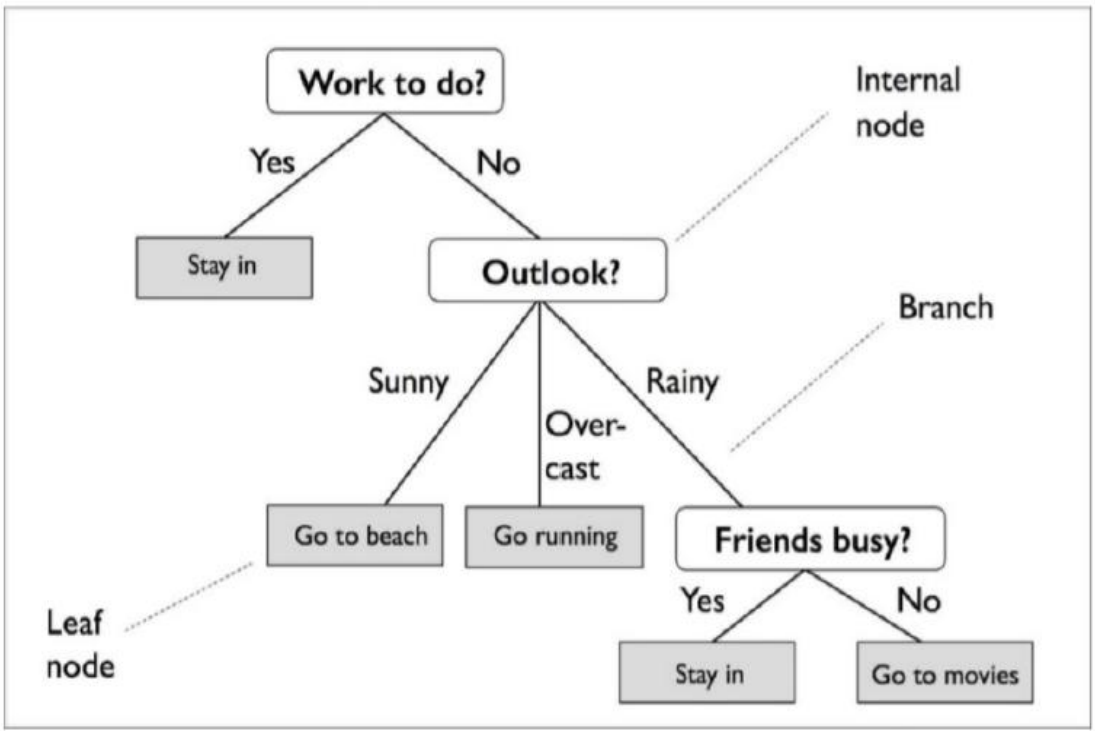


Figure (I.07): *Exemple d'arbre de décision*

3.6. Les forêts aléatoires

Les algorithmes de forêts aléatoires (Random Forest ou RF) sont connus pour être des outils très efficaces de classification dans de nombreux domaines, notamment en finance .Il s'agit d'une méthode de classification d'ensemble, qui établit un ensemble de classificateurs, contrairement aux arbres de décision CART et C5.0 qui ne construisent qu'un classificateur [15].

Avantages:

- C'est un des meilleurs algorithmes pour ce qui est de la précision.

- Incorporation de la validation croisée.

Inconvénients:

- Une implémentation difficile.

3.7. Les machines à vecteur de support

Les machines à vecteurs de support (Support Vector Machine ou SVM) sont des algorithmes d'apprentissage Supervisé, utiles tant pour les problèmes de classification que régression, et dont l'objectif est de séparer les données en classe à l'aide d'un séparateur que l'on appellera "frontière" et qui va maximiser la distance entre ces classes, appelée "Marge" [16].

Cas [17].

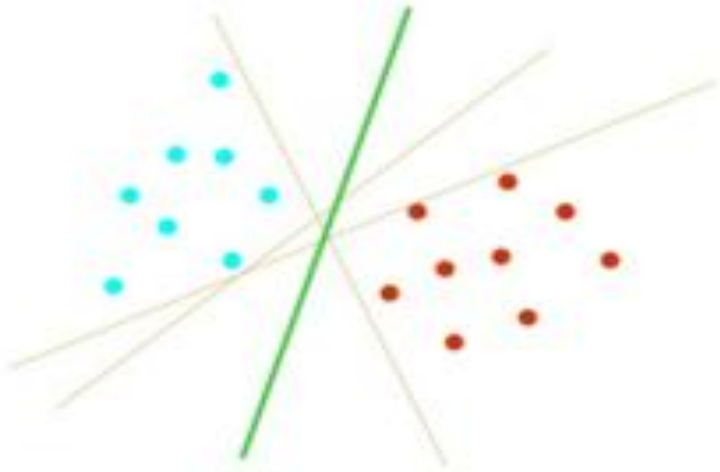


Figure (I.08): *Illustration d'un hyperplan de séparation optimale*

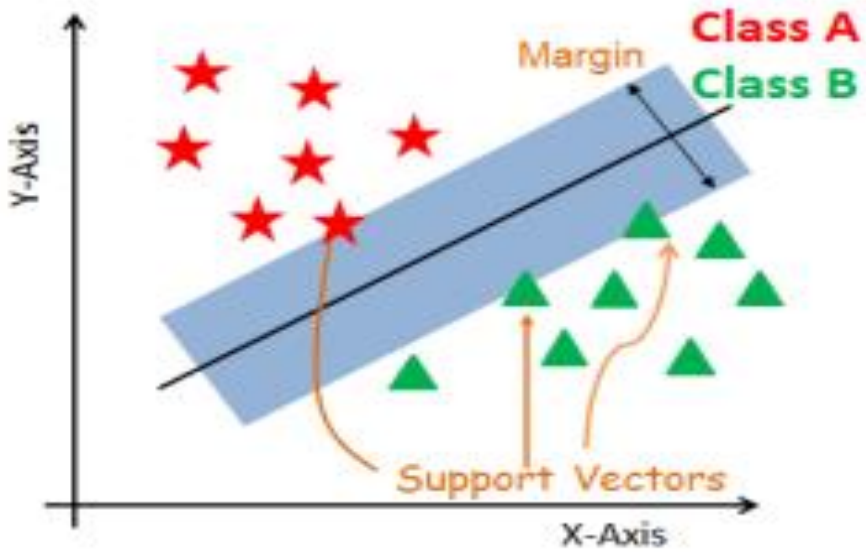


Figure (I.09): *Support Vector Machine*

Avantages:

- Sa grande précision de prédiction.

- Fonction bien sur de plus petits data sets.

- Ils peuvent être plus efficace car ils utilisent un sous-ensemble de points d'entrainement.

Inconvénients:

- Ne convient pas à des jeux de données plus volumineux, car le temps d'entrainement avec les SVM peut être long.

- Moins efficace sur les jeux de données contenant des bruits et beaucoup d'outiliers [18].

3.8. Les Réseaux de Neurones

Les **Réseaux de Neurones** sont une série d'algorithmes qui s'efforcent de reconnaitre les relations sous jacentes dans un ensemble de données grâce à un processus qui imite le fonctionnement du cerveau humain. En ce sens, les réseaux de neurones font référence à des systèmes de neurones, qu'ils soient de nature organique ou artificielle [19].

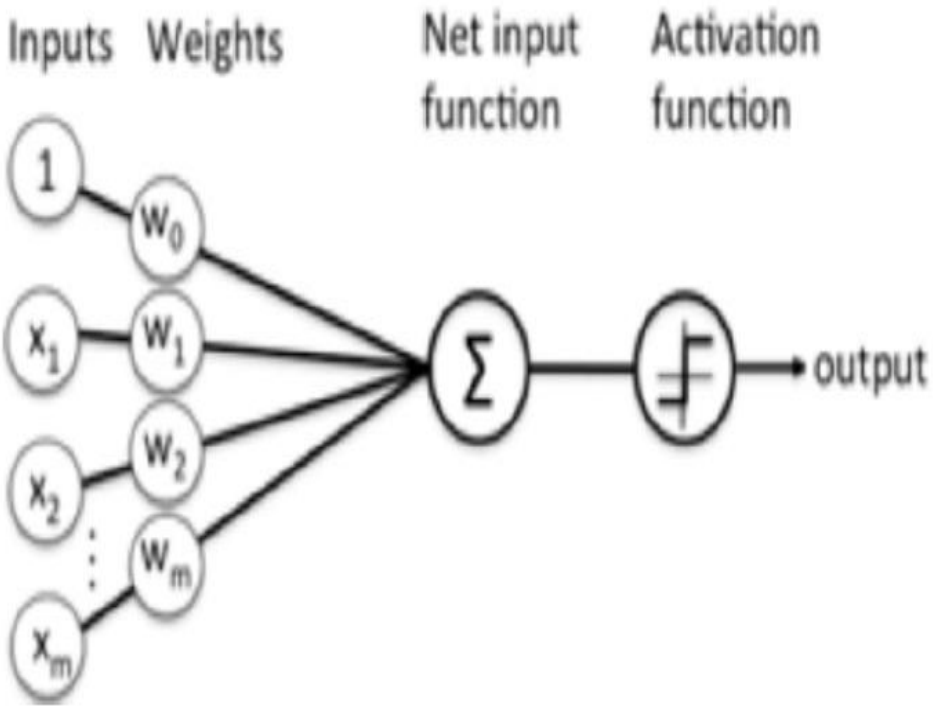


Figure (I.10): *Les Réseaux de Neurones*

4. Performance et sur-apprentissage

Le sur-apprentissage ou (Overfiting) désigne le processus dans lequel un modèle s'adapte tellement aux données historiques qu'il en devient inefficace pour des prédictions futures. L'algorithme trouvera donc des relations dans les données d'entrainement qui au final ne s'appliquent pas dans le cas des données étudiés [20].

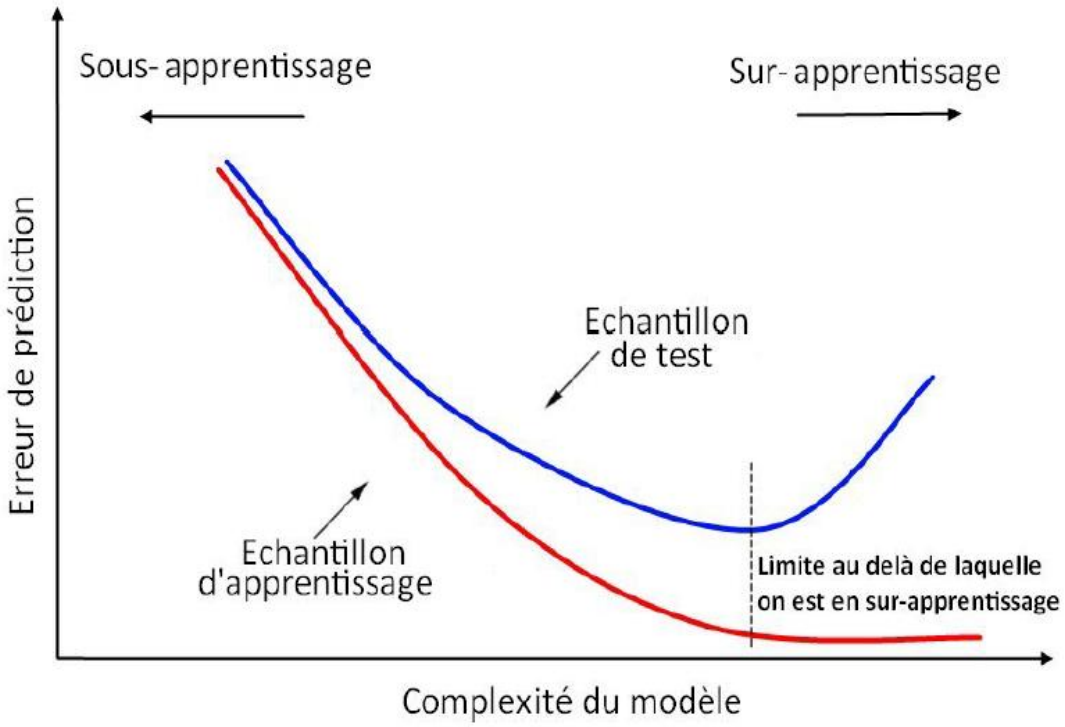


Figure (I.11): *Sur-Apprentissage*

Le **sous-apprentissage (Underfitting)** est l'inverse et désigne le cas ou l'algorithme n'apprend pas assez de relations que pour faire des prédictions précisent, et se caractérise par une faible variance, mais un biais élevé.

Le meilleur modèle est celui du juste milieu, il ne doit souffrir ni d'Underfitting ni d'Overfitting.

Pour résoudre ce, on divise les données en deux Groupes distincts. Le premier sera L'ensemble d'apprentissage. Le deuxième sera **l'ensemble de test**.

Pour avoir une bonne séparation des données en données d'apprentissage et données de test, on utilise La validation Croisée.

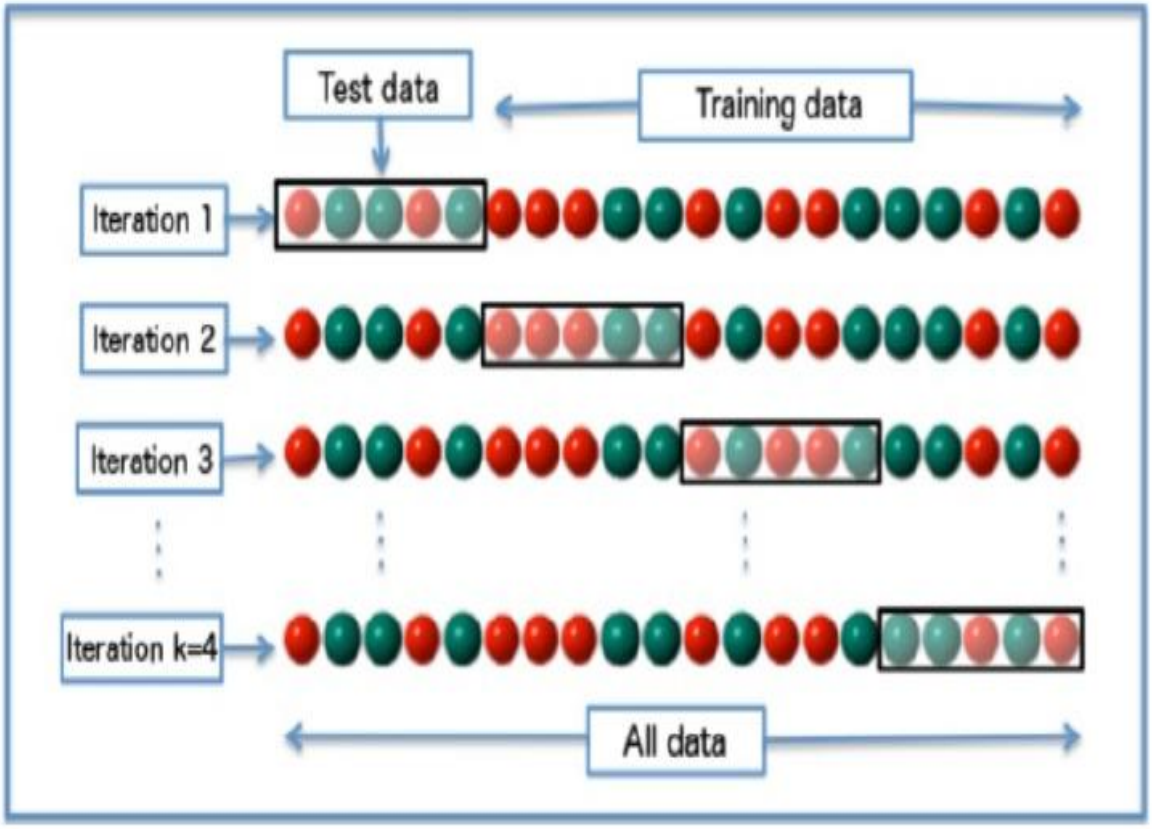


Figure (I.12): La Validation Croisée

L'idée c'est séparer aléatoirement les données dont on dispose en *k* parties, une fera office d'ensemble de test et les autres constitueront l'ensemble d'apprentissage.

- Après que chaque échantillon ait été utilisé une fois comme ensemble de test[21].

5. Conclusion

L'apprentissage automatique (ML) peut être Supervisé ou Non Supervisé, si nous avons moins de données et des données Clairement étiquètes pour la formation (Training), optez pour l'apprentissage Supervisé.

L'apprentissage Non supervisé donnerait généralement de meilleures performances et résultats pour les grands ensembles de données.

CHAPITRE II

L'Hypothyroïdie: Présentation De La maladie

1. Introduction

L'hypothyroïdie est une maladie se caractérisant par l'incapacité de la glande thyroïde à produire suffisamment certaines hormones essentielles au bon fonctionnement du métabolisme**.** Aux premiers stades, les symptômes de l'hypothyroïdie ne sont pas visibles. Au fil du temps, les conséquences d'une hypothyroïdie non soignée se font ressentir sur la santé générale. Elle peut provoquer entre autres une prise de poids, des douleurs au niveau des articulations, des troubles de la fertilité ou des troubles digestifs.

Des examens faciles à mettre en œuvre permettent de diagnostiquer l'hypothyroïdie. Le traitement à base d'hormone thyroïdienne synthétique est simple, sans risque, et efficace une fois que le médecin a déterminé la posologie adéquate pour son patient [22].

2. Présentation de la thyroïde

La thyroïde est une petite glande d'environ 5 cm de diamètre qui est située sous la peau du cou. Les deux moitiés (lobes) de la glande sont connectées par une partie centrale (appelée isthme) qui confère à la thyroïde la forme d'un papillon. Normalement, la thyroïde ne se voit pas et peut être à peine palpée. Si elle grossit, les médecins peuvent la sentir facilement et une masse proéminente (goitre) peut apparaître dans le cou (parfois en dessous ou sur les côtés de la pomme d'Adam).

La thyroïde sécrète les hormones thyroïdiennes qui contrôlent la vitesse des fonctions chimiques de l'organisme (métabolisme de base). Les hormones thyroïdiennes influencent le métabolisme de base de 2 façons :

- En stimulant presque tous les tissus de l'organisme pour produire des protéines
- En augmentant la quantité d'oxygène utilisée par les cellules

Les hormones thyroïdiennes affectent de nombreuses fonctions vitales de l'organisme, comme la fréquence cardiaque, la vitesse à laquelle les calories sont brûlées, l'intégrité de la peau, la croissance, la production de chaleur, la fertilité et la digestion [23].

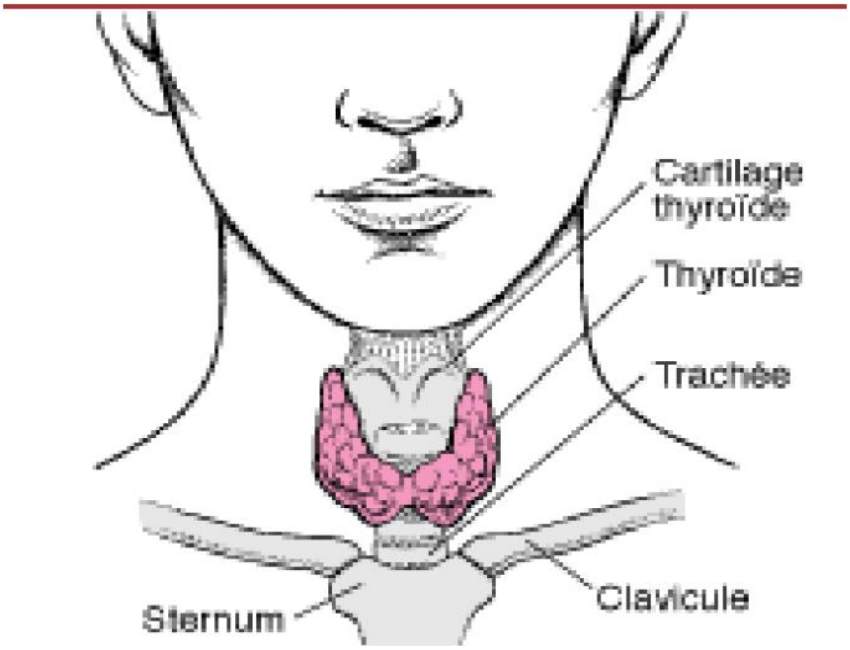


Figure (II.01): *La thyroïde.*

3. Organogénèse de la thyroïde

Il est admis actuellement que la thyroïde dérive de deux ébauches embryonnaires bien distinctes:

- **Une ébauche médiane:** l'ébauche thyroïdienne principale: qui se développe à partir de l'intestin pharyngien sur la ligne médiane, c'est donc une dérivée entoblastique.
- **Deux ébauches latérales:** les ébauches thyroïdiennes accessoires : chacune provient du corpsultimo branchial CUP (dérive de la 5éme poche entoblastique) qui sera colonisé ultérieurement par les cellules de la crête neurale.

Donc la glande thyroïde est d'origine entoblastique et neurectoblastique [24].

4. Régulation de la synthèse des hormones thyroïdiennes

Le principal système de régulation est représenté par l'axe thyréotrope, la sécrétion des HTs (Hormones thyroïdiennes) est principalement sous le contrôle de la TSH (Thydroïd Stimulating Hormone) hypophysaire qui stimule spécifiquement la prolifération des cellules folliculaires. L'hormonosynthèse se déroule en plusieurs étapes, capture de l'iode, iodation de la thyroglobuline, pinocytose, hydrolyse de la thyroglobuline et sécrétion hormonale. La TSH est sous le contrôle de l'hypothalamus, puisque sa sécrétion est stimulée par la TRH (Thyrotropin Releasing Hormone). Ce système est complété par un système de

rétrocontrôle négatif exercé par les THs, car leur augmentation entraine une diminution de la sécrétion de la TRH. Par ailleurs, le statut nutritionnel influence également la fonction thyroïdienne et en particulier le catabolisme des hormones, en cas de jeûne, de dénutrition ou d'hypercatabolisme, la 5' désiodase est inhibée avec diminution des taux sanguins de T3 et augmentation de ceux de T3 reverse [25].

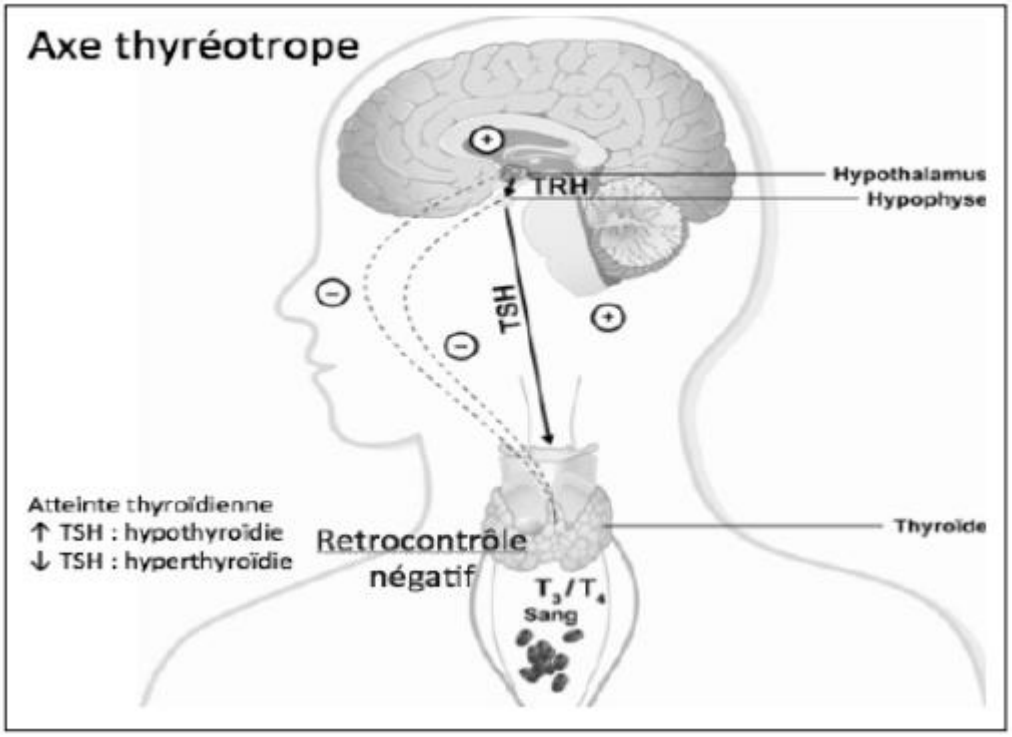


Figure (II.02): *Régulation de la production des hormones thyroïdiennes.*

5. Effets des hormones thyroïdiennes sur l'organisme

Les hormones thyroïdiennes agissent sur de nombreux organes, leur sécrétion est indispensable au développement et au maintien de l'homéostasie[26].

5.1. Au cours de la vie embryonnaire et fœtale

La thyroïde maternelle assure les besoins de l'embryon jusqu'à la dixième semaine de vie intra-utérine et elle passe librement la barrière placentaire. La thyroïde fœtale va ensuite devenir fonctionnelle. Le rôle des hormones thyroïdiennes est important au niveau de la croissance osseuse et surtout de la maturation nerveuse. Tout déficit dû à une carence maternelle ou embryonnaire peut se manifester par un retard de développement psychomoteur. Le dépistage de l'hypothyroïdie néonatale est essentiel afin de corriger très précocement le déficit [26].

5.2. Effets métaboliques

Les hormones thyroïdiennes augmentent tous les métabolismes. Elles sont de ce fait:

- thermogéniques (base de l'ancien test diagnostique étudiant le métabolisme de base);
- hyperglycémiantes;
- hypolipidémiantes;
- protéolytiques;
- ostéolytiques [26].

5.3. Effet sur le système nerveux central

Les HT favorisent la myélinisation des fibres nerveuses et stimulent le développement et la croissance des axones, des corps cellulaires et des dendrites[26].

5.4. Effet sur les muscles squelettiques

La carence en HT entraîne une augmentation du volume et de la consistance des muscles squelettiques donc la contraction est ralentie alors que dans l'hyperthyroïdie cette contraction se fait à une vitesse presque normale mais est relativement inefficace. L'administration à un hypothyroïdien de faible dose d'HT augmente l'efficacité du travail musculaire, alors que de fortes doses la diminue [26].

5.5. Effet cardio-vasculaire

Les HT augmentent le débit vasculaire et surtout le rythme cardiaque. Elles imitent un état hyperadrénergique en stimulant les récepteurs β-adrénergiques du myocarde. L'effet cardiaque est couplé à une vasodilatation périphérique due à l'augmentation du métabolisme de tous les tissus et à la calorigenèse [26].

5.6. Effet sur le système digestif

Les HT augmentent la motricité intestinale, le débit sanguin intestinal, la consommation d'oxygène et l'absorption intestinale [26].

5.7. Effet sur la fonction rénale

Les HT augmentent le taux de filtration glomérulaire et le débit sanguin rénal. Cependant, en excès, elles diminuent la capacité de concentration hydrique du rein. Elles maintiennent donc une diurèse hydrique [26].

5.8. Effet sur le comportement

Enfin, les hormones thyroïdiennes agissent sur le comportement psychique d'une personne. En cas d'excès en hormones thyroïdiennes, on remarque souvent un état d'agitation anxieuse accompagnée d'irritabilités et d'insomnies; l'humeur est souvent triste; un épisode aigu peut amener à une psychose maniaco-dépressive. Au contraire, en cas de manque d'hormones thyroïdiennes, les troubles psychiques sont caractérisés par un ralentissement intellectuel, une indifférence affective et une tristesse [26].

6. L'hypothyroïdie

L'hypothyroïdie est un dysfonctionnement endocrinien qui se définie comme l'incapacité de la glande thyroïde à produire suffisamment d'hormones thyroïdiennes associée à une augmentation de l'hormone stimulant la thyroïde (TSH) et qui peut survenir suite à une défaillance primaire des glandes ou suite à une stimulation insuffisante de la thyroïde par l'hypothalamus ou l'hypophyse engendrant un état d'hypométabolisme [27].

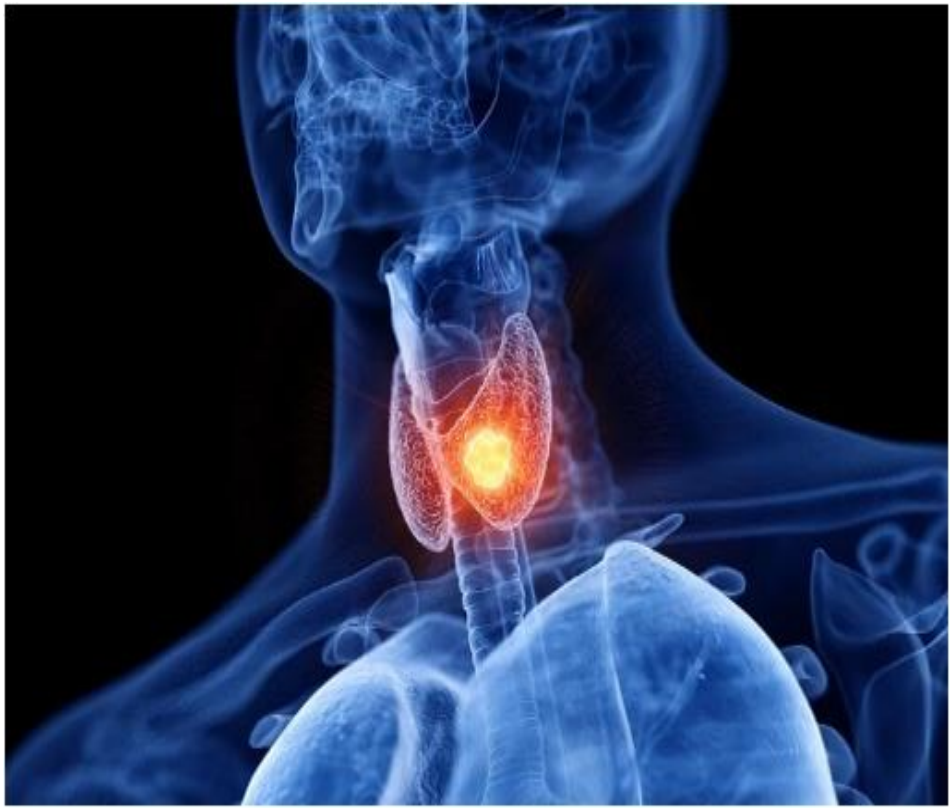


Figure (II.03) : L'hypothyroïdie.

7. Qui est touché par l'hyperthyroïdie?

On estime qu'entre 1 et 4 % de la population française souffre d'hyperthyroïdie, le plus souvent dans sa forme légère, sans symptôme.

L'hyperthyroïdie est plus fréquente :

- chez les femmes,
- chez les personnes âgées de plus de 60 ans,
- dans les familles au sein desquelles un cas de maladie de la thyroïde a été diagnostiqué par le passé,
- chez les femmes qui ont récemment accouché (jusqu'à 7 % d'entre elles seraient touchées de manière transitoire par ce problème de santé pendant l'année suivant la naissance) [28].

8. Les symptômes et les complications de l'hyperthyroïdie

8.1. Quels sont les symptômes de l'hyperthyroïdie ?

Il est assez fréquent que l'hyperthyroïdie ne produise pas de symptômes, particulièrement chez les personnes âgées de plus de 60 ans. Lorsque des symptômes apparaissent, on observe une accélération du rythme cardiaque et des palpitations, des bouffées de chaleur avec transpiration et soif excessive, des tremblements, des troubles du sommeil, de l'irritabilité, de l'anxiété, des selles molles et fréquentes, de la fatigue, une perte de force musculaire, une baisse de libido, un amaigrissement, des chutes de cheveux, etc.

- **L'accélération du rythme cardiaque :** le pouls est fréquemment supérieur à cent battements par minute au repos, la personne se plaint de palpitations, d'essoufflement ou de « battements » dans la poitrine.
- **Des troubles de la régulation de la température du corps :** la personne transpire facilement, elle a les mains moites et parfois des bouffées de chaleur ; elle craint la chaleur et se plaint de soif excessive.
- **Les troubles du système nerveux :** la personne souffre de tremblements, notamment au niveau des mains, de difficultés à trouver le sommeil, de nervosité, d'irritabilité et de sautes d'humeur, d'anxiété voire de dépression. Fréquemment, anxiété et nervosité sont les premiers signes notables de l'hyperthyroïdie.
- **Les troubles du système digestif :** le transit intestinal est accéléré et les selles sont plus fréquentes (mais les diarrhées sont rares).

- **Les troubles de l'état général :** fatigue permanente, faiblesse musculaire (en particulier des bras et des cuisses), perte de poids importante (jusqu'à plusieurs kilos par semaine malgré un appétit augmenté), fonte des muscles, règles moins fréquentes et moins abondantes, baisse du désir sexuel, peau fine et cheveux cassants ou qui tombent.

Dans certaines formes de la maladie (maladie de Basedow), ces symptômes sont parfois associés à une augmentation de volume de la thyroïde (goitre à la base du cou), un gonflement de la peau des jambes au niveau des tibias, et des troubles oculaires : les yeux semblent anormalement écarquillés ou « sortir de la tête » (exophtalmie) et la personne se plaint d'avoir les yeux secs et qui piquent[29].

8.2. Quelles sont les complications de l'hyperthyroïdie?

Les complications de l'hyperthyroïdie non traitée sont essentiellement cardiaques (insuffisance cardiaque, fibrillation auriculaire et autres troubles du rythme), psychiatriques (confusion, agitation, délire, par exemple) ou liées à l'état général de la personne (fatigue intense, amaigrissement important).

Parce qu'un excès de T3/T4 diminue l'absorption du calcium au niveau des os, les personnes qui souffrent d'une hyperthyroïdie non traitée risquent également de développer de l'ostéoporose [29].

9. Le diagnostic des problèmes de la thyroïde

La première étape de la consultation implique généralement une palpation au niveau du cou du malade. Lors de cet examen préliminaire, le médecin cherche à détecter la présence de nodules ou d'un goitre en l'occurrence. La deuxième phase de l'examen a pour but de déceler d'éventuelles irrégularités en termes de sécrétions hormonales par le biais d'une analyse de sang. Ensuite, une échographie thyroïdienne est réalisée pour examiner plus en profondeur les nodules préalablement détectés. Ainsi, le nombre d'excroissances et autres spécificités connexes, de même que les détails entourant les chaînes ganglionnaires du cou sont révélées. La ponction cytologique (permettant une analyse cellulaire au microscope) et la scintigraphie (pour distinguer la typologie des nodules, froids ou chauds) sont les examens complémentaires indispensables pour déterminer le type de pathologie thyroïdienne et prescrire ainsi le traitement adéquat [30].

Figure (II.04): Le diagnostic des problèmes de la thyroïde.

10. Les médicaments contre l'hyperthyroïdie

Lors de traitement contre l'hyperthyroïdie, les médicaments prescrits visent soit à réduire les taux sanguins d'hormones thyroïdiennes (en bloquant leur production par la thyroïde), soit à soulager les symptômes et, en particulier, à soutenir le cœur si l'accélération du rythme cardiaque est trop élevée. De plus, chez les patients dont la thyroïde a été neutralisée ou enlevée, des hormones thyroïdiennes de synthèse sont administrées pour rétablir leur taux sanguin normal [28].

10.1. Les antithyroïdiens de synthèse

Ces médicaments bloquent la production des hormones thyroïdiennes par la thyroïde. Ils permettent de contrôler efficacement l'hyperthyroïdie pendant une longue durée (par exemple lors de maladie de Basedow) ou en attendant un traitement chirurgical ou par iode radioactif. La dose prescrite est individuelle. Elle est fixée par le médecin en fonction du résultat des dosages sanguins de T3 et de T4. Le retour à un taux normal d'hormones thyroïdiennes n'est jamais immédiat : deux à quatre mois de traitement antithyroïdien peuvent être nécessaires.

Les antithyroïdiens de synthèse ont des effets indésirables qui touchent divers organes : démangeaisons, rougeurs cutanées, douleurs articulaires, fièvre ou baisse anormale des globules blancs (également appelée **agranulocytose**). Lors d'agranulocytose, le patient est plus exposé aux maladies infectieuses.

Cet effet indésirable est rare (moins de 1 % des personnes en traitement) mais potentiellement dangereux.

Pour surveiller le traitement, le médecin prescrit des analyses de sang avant de débuter le traitement, toutes les semaines pendant les six premières semaines de traitement, puis de façon plus espacée mais régulière. De plus, le patient est informé qu'il doit cesser immédiatement son traitement en cas de fièvre, d'angine ou de tout autre signe d'infection. Dans ce cas, il doit rapidement consulter son médecin pour avis [28].

10.2. Les autres médicaments utilisés en cas d'hyperthyroïdie

10.2.1. Les médicaments destinés à ralentir le cœur

Lorsque l'hyperthyroïdie provoque une accélération ou des troubles du rythme cardiaque sévères, il est nécessaire d'associer aux antithyroïdiens des médicaments pour ralentir et régulariser les battements du cœur. Ces médicaments font partie de la famille des bêtabloquants [28].

10.2.2. Les hormones thyroïdiennes

Des hormones thyroïdiennes de synthèse doivent parfois être prescrites quand les médicaments antithyroïdiens provoquent des taux sanguins d'hormones thyroïdiennes inférieurs à la normale.

Elles sont également utilisées en remplacement des hormones thyroïdiennes naturelles chez les personnes qui ont subi une ablation de la thyroïde ou chez qui la thyroïde a été neutralisée par l'iode radioactif. Ce traitement est identique à celui suivi par les personnes qui souffrent d'insuffisance thyroïdienne (hypothyroïdie).

La survenue d'effets indésirables, similaires aux symptômes de l'hyperthyroïdie, doit faire penser à un surdosage [28].

11. Conclusion

L'hypothyroïdie est une pathologie thyroïdienne ou la glande thyroïde est incapable de maintenir une sécrétion adéquate d'hormones, ce qui amène l'hypophyse à libérer plus de quantités de TSH pour compenser le manque, en raison du dysfonctionnement de la glande thyroïde, la production endogène d'hormones thyroïdiennes reste insuffisante. Les manifestations cliniques de cette insuffisance peuvent varier considérablement car les récepteurs des hormones thyroïdiennes sont présents dans la plupart des organes et des tissus du corps. Les systèmes cardiovasculaires, pulmonaires, gastro-intestinaux, muscle squelettique et neurologiques peuvent tous être affectés, de même que les reins, la peau et les tissus conjonctifs, comme elle produit d'importants troubles métaboliques [25].

CHAPITRE III

Initiation

WEKA

1. Introduction

WEKA est un logiciel open-source puissant dédié à l'exploration de données et à l'apprentissage automatique. Connu pour sa facilité d'utilisation et sa richesse fonctionnelle, il est couramment utilisé par les chercheurs, les enseignants et les professionnels du domaine de la data science. Son accessibilité en fait une solution de choix pour ceux qui souhaitent expérimenter avec des algorithmes avancés sans nécessiter de compétences en programmation.

2. Historique de WEKA

WEKA (Waikato Environment for Knowledge Analysis) a été développé par l'Université de Waikato en Nouvelle-Zélande dans les années 1990. Son objectif initial était de fournir un outil pratique pour l'analyse des données et l'exploration des modèles d'apprentissage automatique.

Au fil des années, Weka a évolué en intégrant des algorithmes avancés et en proposant une interface graphique conviviale. Il a gagné en popularité dans la communauté scientifique grâce à sa simplicité et à son efficacité. Son statut open-source a permis une adoption large et une amélioration continue grâce aux contributions des chercheurs et des développeurs du monde entier.

3. Points forts de Weka

- **Interface intuitive et conviviale :** Weka propose une interface graphique qui simplifie l'utilisation des algorithmes d'apprentissage automatique, ce qui le rend accessible aux débutants comme aux experts.
- **Large bibliothèque d'algorithmes :** Il intègre une vaste gamme d'algorithmes de classification, de clustering, de régression et de sélection de caractéristiques, offrant ainsi une flexibilité importante.
- **Visualisation avancée des données :** Grâce à ses outils graphiques, Weka permet une compréhension approfondie des tendances et des relations dans les ensembles de données.
- **Compatibilité étendue :** Il prend en charge divers formats de données, y compris ARFF, CSV et autres formats couramment utilisés dans le domaine.

- **Extensibilité et personnalisation :** Son architecture open-source permet aux utilisateurs avancés de développer et d'ajouter leurs propres algorithmes et outils d'analyse.
- **Automatisation et scripting :** Weka peut être utilisé en mode ligne de commande ou via des scripts, facilitant ainsi l'automatisation des processus d'analyse.

4. Faiblesses de Weka

- **Limitations en big data :** Weka est principalement conçu pour fonctionner sur des ensembles de données de taille modérée. Il peut rencontrer des problèmes de performances lorsqu'il est utilisé avec de très grands volumes de données.
- **Interface graphique datée :** Bien que fonctionnelle, l'interface graphique de Weka n'a pas connu de mise à jour majeure et peut sembler vieillissante par rapport aux standards modernes.
- **Manque de flexibilité par rapport aux langages de programmation:** Comparé à des outils comme Python (avec Scikit-learn) ou R, Weka offre moins de contrôle et d'optimisation sur les modèles, ce qui peut limiter les possibilités avancées.
- **Moins adapté aux applications en production :** Weka est principalement utilisé pour l'expérimentation et la recherche, mais il n'est pas le choix privilégié pour les systèmes de production nécessitant des performances élevées et une intégration avancée.

5. Conclusion

Weka est un outil puissant, accessible et polyvalent pour l'exploration de données et l'apprentissage automatique. Il constitue une excellente option pour les chercheurs, les enseignants et les étudiants souhaitant expérimenter avec divers algorithmes sans une courbe d'apprentissage trop abrupte. Toutefois, ses limitations en matière de big data et son interface vieillissante peuvent restreindre son utilisation pour certaines applications avancées. Mal gré cela, il reste un choix pertinent pour des analyses de données de taille moyenne et constitue un excellent point de départ pour ceux qui souhaitent se familiariser avec l'apprentissage automatique.

6. Évaluation des modèles en apprentissage automatique

L'évaluation d'un modèle est une étape essentielle pour mesurer ses per formances et sa capacité à généraliser sur de nouvelles données. Différentes méthodes existent pour évaluer un modèle, chacune ayant ses avantages et ses limites. Voici les cinq principales méthodes d'évaluation utilisées dans Weka et leur mise en œuvre.

6.1. Utilisation de l'ensemble d'entraînement :

Explication : Cette méthode consiste à tester le modèle sur les mêmes données que celles utilisées pour l'entraînement. Comme le modèle a déjà vu ces données, il risque d'afficher une précision très élevée, ce qui peut être trompeur. Cette méthode ne permet donc pas de mesurer la capacité de généralisation du modèle.

Utilisation dans Weka

- Charger un dataset dans Weka.
- Sélectionner un algorithme de classification.
- Cocher l'option "**Use training set**" dans l'onglet "Test options".
- Lancer l'évaluation.

6.2. Utilisation d'un ensemble de test fourni

Explication : Dans cette méthode, l'entraînement est effectué sur un ensemble de données donné, et le test est réalisé sur un fichier de test distinct. Cela permet d'obtenir une meilleure estimation de la performance du modèle sur de nouvelles données, mais la qualité du test dépend fortement de la répartition des données.

Utilisation dans Weka :

- Charger un dataset pour l'entraînement.
- Sélectionner un algorithme de classification.
- Cocher l'option "**Supplied test set**".
- Cliquer sur "**Set**" pour charger un fichier de test externe.
- Lancer l'évaluation.

6.3. Validation croisée (k-fold cross-validation)

Explication : La validation croisée divise le jeu de données en k parties (ou "folds"). Le modèle est entraîné sur $k - 1$ parties et testé sur la partie restante. Ce processus est répété k fois, en changeant la partie utilisée pour le test à chaque itération. Cela permet d'avoir une estimation plus fiable des performances du modèle.

Utilisation dans Weka :

- Charger un dataset dans Weka.
- Sélectionner un algorithme de classification.
- Cocher l'option "**Cross-validation**".
- Choisir une valeur pour **k** (exemple : 10 pour une validation croisée à 10 folds).
- Lancer l'évaluation.

6.4. Leave-One-Out Cross-Validation (LOOCV)

Explication : Cette méthode est une version extrême de la validation croisée où chaque instance du dataset est utilisée comme un test, et toutes les autres servent pour l'entraînement. Cela signifie que si le dataset contient N instances, alors N modèles seront entraînés et testés. Cette méthode four nit une estimation précise de la performance du modèle, mais elle est très coûteuse en temps de calcul, surtout pour des grands datasets.

Utilisation dans Weka :

- Suivre les mêmes étapes que pour la validation croisée.
- Régler le nombre de folds à N (le nombre total d'instances dans le dataset).
- Lancer l'évaluation.

6.5. Découpage en pourcentage (Percentage Split)

Explication : Cette méthode divise aléatoirement le dataset en deux par ties : une partie pour l'entraînement et une autre pour le test. Par exemple, un découpage à 70% signifie que 70% des données sont utilisées pour l'apprentissage et 30% pour l'évaluation. Cette approche est simple et rapide mais dépend fortement de la répartition des données.

Utilisation dans Weka :

- Charger un dataset dans Weka.
- Sélectionner un algorithme de classification.
- Cocher l'option "Percentage split".
- Spécifier le pourcentage d'entraînement (exemple : 70%).
- Lancer l'évaluation.

6.6. Conclusion

Chaque méthode d'évaluation a ses avantages et ses inconvénients :

- **L'utilisation de l'ensemble d'entraînement surestime** les performances et ne reflète pas la généralisation du modèle.
- **L'utilisation d'un ensemble de test fourni** permet une bonne évaluation mais dépend fortement de la qualité des données de test.
- **La validation croisée** est l'une des meilleures méthodes, offrant une bonne estimation de la performance du modèle.
- La **LOOCV** est très précise mais coûteuse en temps de calcul.
- Le **découpage en pourcentage** est rapide mais peut introduire une variabilité dans les résultats selon la répartition des données.

La validation croisée est généralement recommandée pour évaluer correctement un modèle tout en évitant les biais liés à un mauvais découpage des données.

Cette image montre l'interface du Weka GUI Chooser, qui permet de choisir différentes applications pour l'analyse de données et l'apprentissage automatique avec Weka. Voici une explication de chaque bouton sur le côté droit :

7. Présentation de la première l'interface de weka

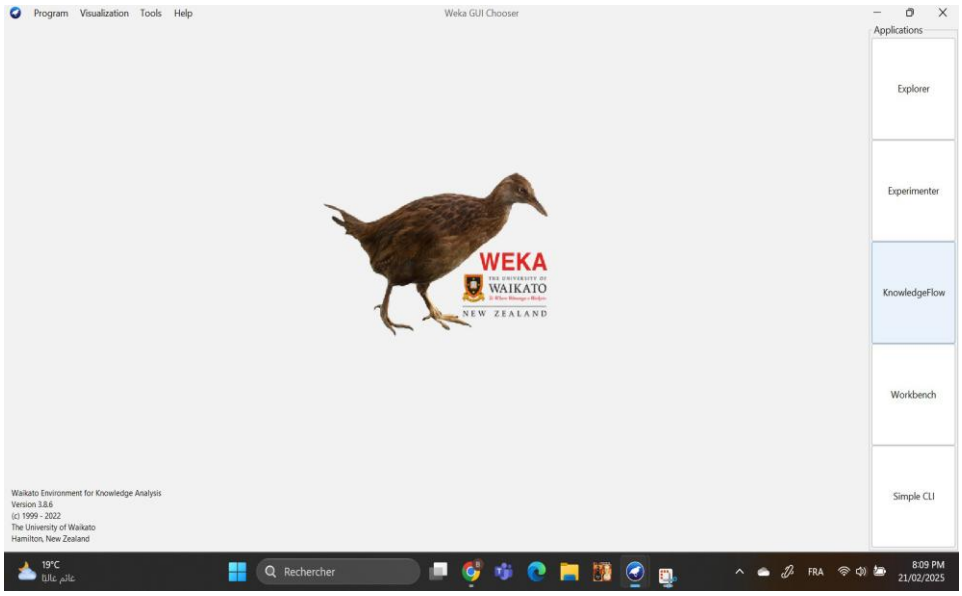


Figure (III.01): *Interface principale de Weka GUI Chooser*

8. Description des options

8.1. Explorer

- Interface principale pour manipuler les jeux de données, entraîner des modèles de classification, clustering et appliquer des filtres.

- Offre des outils pour visualiser les données et évaluer les performances des algorithmes.

8.2. Présentation de EXPLORER l'interface de weka

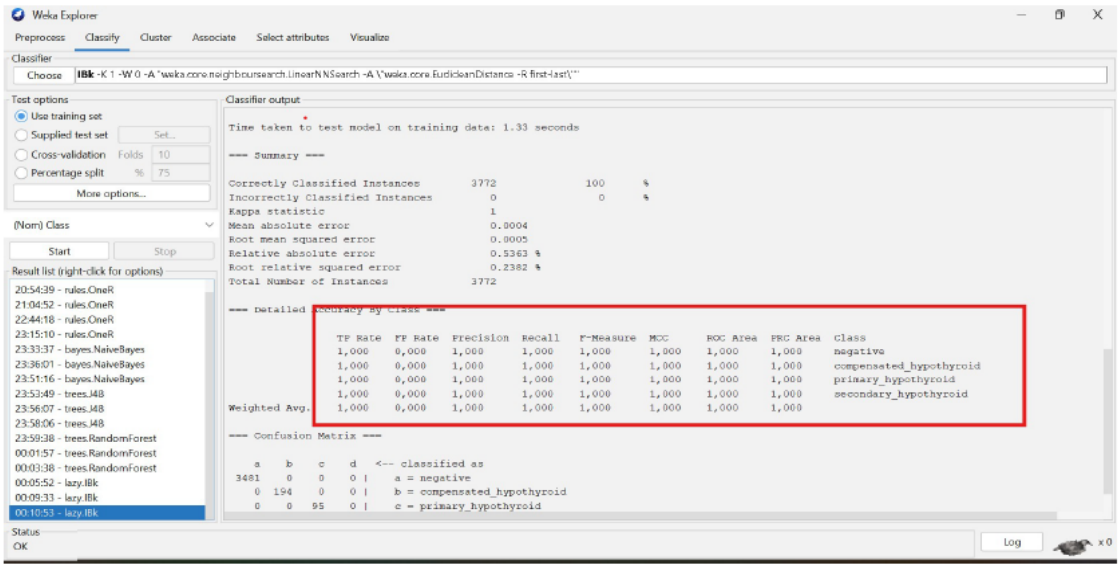


Figure (III.02): *Interface EXPLORER.*

9. Explication de l'interface Weka Explorer

L'interface **Weka Explorer** est une interface graphique permettant le prétraitement des données, la classification, le clustering et la visualisation. L'image montre l'onglet Classify, où les modèles d'apprentissage automatique peuvent être appliqués à un ensemble de données. Voici une explication des différentes sections de l'interface :

9.1. Sélection du Classifieur (Haut de l'interface)

- Le bouton "**Choose**" permet de sélectionner un algorithme de classification.

- Dans l'image, l'algorithme sélectionné est **IBk** (K-Nearest Neighbors, KNN)

Avec les paramètres suivants :

* -K 1 : Utilisation d'un seul voisin.

* -W 0 : Aucun poids attribué aux voisins.

* -A "weka.core.neighboursearch.LinearNNSearch -R first-last" : Utilisation de la distance Euclidienne pour la recherche des plus proches voisins.

9.2. Options de Test (Panneau de Gauche)

- Différentes méthodes pour évaluer le modèle :

- **Use training set :** Le modèle est testé sur les mêmes données d'entraînement (sélectionné dans ce cas).

- **Supplied test set :** Permet de tester sur un ensemble de données externe.

- **Cross-validation :** Divise les données en plusieurs sous-ensembles (folds) pour améliorer la généralisation.

- **Percentage split :** Utilise une fraction des données pour l'entraînement et le reste pour le test (exemple : 75%).

9.3. Sortie du Classifieur (Panneau Principal)

- Affiche les résultats du modèle, comprenant :

- **Résumé** des instances correctement et incorrectement classifiées.

- **Statistiques d'erreur**, telles que l'erreur absolue moyenne (MAE) et l'erreur quadratique moyenne (RMSE).

- **Statistique Kappa**, mesurant l'accord entre les valeurs réelles et prédites.
- **Précision détaillée par classe**, incluant la précision, le rappel, la F-mesure et l'aire sous la courbe ROC pour chaque classe.

9.4. Matrice de Confusion

- Affiche le nombre d'instances de chaque classe qui ont été bien ou mal classées.
- Dans l'exemple de l'image, toutes les instances sont parfaitement classées (100% de précision).

9.5. Liste des Résultats (Panneau Inférieur Gauche)

- Enregistre les résultats des classifications précédentes.
- Différents algorithmes (*OneR, Naïve Bayes, J48, Random Forest, IBk*) ont été testés dans cette session.
- Un clic droit permet d'afficher des détails supplémentaires, comme des graphes ou des arbres de décision.

9.6. Description de la base de données utiliser "Hypothyroid"

9.6.1. Généralités

- Nombre total d'instances : 3 772
- Nombre total d'attributs : 30 (y compris la classe cible)
- Objectif : Prédire l'hypothyroïdie
- Type de données : Catégoriques et numériques
- Mode de test : Validation croisée en 10 partitions

9.6.2. Attributs

Informations démographiques et antécédents médicaux :

- ❖ Age
- ❖ Sex
- ❖ On thyroxine
- ❖ Query on thyroxine
- ❖ On antithyroid medication
- ❖ Sick
- ❖ Pregnant

- ❖ Thyroid surgery
- ❖ I131 treatment
- ❖ Query hypothyroid
- ❖ Query hyperthyroid
- ❖ Lithium
- ❖ Goitre
- ❖ Tumor
- ❖ Hypopituitary
- ❖ Psych

Mesures biologiques et tests médicaux :

- TSH measured
- TSH
- T3 measured
- T3
- TT4 measured
- TT4
- T4U measured
- T4U
- FTI measured
- FTI
- TBG measured
- TBG

Source des données :

- Referral source

Classe cible :

- Class (negative , compensated_hypothyroid , secondary_hypothyroid , primary_hypothyroid)

9.7. Les algorithms utiliser

9.7.1. OneR

Définition : L'algorithme One Rule (OneR) est une méthode simple de classification qui crée des règles basées sur une seule caractéristique du dataset. Il sélectionne l'attribut qui donne la meilleure prédiction en fonction d'un seuil optimal.

Fonctionnement :

- Il analyse chaque attribut indépendamment et génère une règle basée sur celui-ci.
- Il choisit la règle qui a la plus faible erreur de classification.
- Ensuite, cette règle est appliquée pour classer les nouvelles instances.

Avantages :

- Très simple à comprendre et à implémenter.
- Rapide en entraînement et en prédiction.
- Peut servir comme base pour comparer d'autres algorithmes plus com plexes.

Inconvénients :

* Moins précis que des modèles plus avancés.
* Fonctionne mal si les données nécessitent plusieurs attributs pour une classification correcte.

9.7.2. Naïve Bayes

Définition : Naïve Bayes est un classificateur probabiliste basé sur le théorème de Bayes, qui suppose que les attributs sont indépendants les uns des autres (hypothèse du caractère naïf).

Fonctionnement :

- Il calcule la probabilité qu'une instance appartienne à une classe en utilisant la formule de Bayes.
- Chaque attribut est évalué indépendamment, et la classe avec la plus grande probabilité est choisie.

Avantages :

- Très rapide, même sur de grands ensembles de données.
- Fonctionne bien avec des données catégoriques et textuelles.
- Robuste face aux données bruitées ou manquantes.

Inconvénients :

- Hypothèse d'indépendance souvent irréaliste.

- Moins performant si les variables sont fortement corrélées.

9.7.3. J48 (C4.5)

Définition : J48 est une implémentation de l'algorithme C4.5, qui est une amélioration de l'ID3. Il génère un arbre de décision en analysant les attributs du dataset.

Fonctionnement :

- Il sélectionne l'attribut qui divise le mieux les données en fonction du gain d'information.

- Il crée des branches pour chaque valeur de l'attribut sélectionné. — L'arbre est construit récursivement jusqu'à ce que toutes les instances soient bien classées.

Avantages :

- Facile à interpréter sous forme d'arbre de décision.

- Fonctionne bien avec des données mixtes (numériques et catégoriques).

- Prend en charge les valeurs manquantes et la réduction de l'overfitting grâce à l'élagage.

Inconvénients :

- Peut être sensible aux données bruitées.

- L'arbre peut devenir trop complexe si le dataset est très grand.

9.7.4. Random Forest

Définition : Random Forest est un ensemble d'arbres de décision qui vote pour déterminer la classe d'une instance.

Fonctionnement :

- Il génère plusieurs arbres de décision à partir de sous-échantillons des données (bagging).

- Chaque arbre prend une décision et la classe finale est choisie par un vote majoritaire.

- Il introduit de l'aléatoire pour éviter l'overfitting.

Avantages :

- Très robuste et précis.
- Moins sensible aux données bruitées et aux valeurs aberrantes.
- Peut traiter des datasets de grande taille.

Inconvénients :

- Plus lent que les arbres de décision individuels.
- Moins interprétable en raison du grand nombre d'arbres générés.

9.7.5. IBK (K-Nearest Neighbors, KNN)

Définition : K-Nearest Neighbors (KNN), ou IBK dans Weka, est un algorithme basé sur la proximité des points de données.

Fonctionnement :

- Il ne construit pas de modèle explicite, mais stocke les données d'en traînement.
- Lorsqu'une nouvelle instance est testée, il recherche les *K* instances les plus proches en fonction d'une distance (souvent Euclidienne).
- La classe majoritaire parmi les voisins est attribuée à l'instance testée.

Avantages :

- Très simple et efficace pour les petites bases de données.
- Fonctionne bien pour les problèmes où les classes sont bien séparées.

Inconvénients :

- Très lent pour de grands datasets, car il doit comparer chaque instance avec toutes les autres.
- Sensible au choix de *K* et à la définition de la distance.
- Sensible aux dimensions élevées (effet de la malédiction de la dimension).

Conclusion : Chaque algorithme a ses propres forces et faiblesses. Le choix du meilleur dépend du type de données et du problème à résoudre. Random Forest et J48 offrent une grande précision et une bonne généralisa tion, tandis que Naïve Bayes est rapide et efficace pour les données textuelles. OneR est utile pour une première analyse, et KNN est puissant pour des da tasets de petite taille mais devient inefficace sur des données volumineuses.

9.8. Les tests de comparaison

9.8.1. OneR

Méthode d'évaluation	**C C I**	**Précision**	**Rappel**	**F-Mesure**
Use Training Set	97.0042 %	0,999	0,980	0,989
Cross-Validation(10)	96.2354%	0.696	0.702	0.698
Percentage Split(75%)	96.0764%	0,993	0,975	0,984

Tableau (III.01): *Résultats pour OneR*

9.8.2. Naïve Bayes

Méthode d'évaluation	**C C I**	**Précision**	**Rappel**	**F-Mesure**
Use Training Set	95.4401%	0,962	0,992	0,977
Cross-Validation(10)	95.281%	0,961	0,993	0.977
Percentage Split(75%)	95.9703%	0,968	0,992	0,980

Tableau (III.02): *Résultats pour Naïve Bayes*

9.8.3. J48 (C4.5)

Méthode d'évaluation	**C C I**	**Précision**	**Rappel**	**F-Mesure**
Use Training Set	98.2%	0,999	1,000	0,999
Cross-Validation(10)	99.5758%	0,998	0,999	0.998
Percentage Split(75%)	99.4698%	0,999	0,999	0,999

Tableau (III.03): *Résultats pour J48 (C4.5)*

9.8.4. Random Forest

Méthode d'évaluation	**C C I**	**Précision**	**Rappel**	**F-Mesure**
Use Training Set	100%	1,000	1,000	1,000
Cross-Validation(10)	99.3107%	0,997	0,997	0.997
Percentage Split(75%)	99.3637%	0,994	0,999	0,997

Tableau (III.04): *Résultats pour Random Forest*

9.9. Analyse des Résultats

9.9.1. OneR

OneR affiche une forte précision sur l'ensemble d'entraînement avec une précision de 97,00%, un rappel de 0,980 et une F-mesure de 0,989, indiquant un excellent ajustement sur les données d'apprentissage. Cependant, en vali dation croisée, la précision chute à 96,23%, et les scores de précision (0,696), rappel (0,702) et F-mesure (0,698) suggèrent un fort surapprentissage. En découpage en pourcentage (75%-25%), les performances restent élevées avec une précision de 96,07%, une précision de 0,993 et un rappel de 0,975, mon trant une meilleure généralisation.

Conclusion: OneR fonctionne bien sur l'ensemble d'entraînement mais a des difficultés à généraliser correctement, bien qu'il se comporte mieux avec un découpage de 75%.

9.9.2. Naïve Bayes

Naïve Bayes obtient des résultats très stables avec une précision de 95,44% sur l'ensemble d'apprentissage et des scores de précision (0,962), rappel (0,992) et F-mesure (0,977) indiquant une bonne capacité de classification. En validation croisée, les performances sont presque identiques (95,28%) avec des valeurs similaires, montrant une excellente robustesse du modèle. Avec le dé coupage à 75%, les résultats restent constants avec une précision de 95,97%, confirmant sa bonne capacité de généralisation.

Conclusion : Naïve Bayes est un algorithme stable et fiable, présentant des performances homogènes quel que soit le mode d'évaluation.

9.9.3. J48 (C4.5)

J48 affiche des performances exceptionnelles, atteignant une précision de 98,2% sur l'entraînement, avec une F-mesure de 0,999, ce qui montre une parfaite classification sur cet ensemble. En validation croisée, la précision augmente encore (99,57%), prouvant une excellente généralisation, et le dé coupage à 75% confirme ces résultats avec une précision de 99,46%. Les scores élevés et homogènes montrent que J48 est bien adapté aux données utilisées.

Conclusion : J48 est un excellent classificateur avec une généralisation fiable et une précision élevée sur tous les tests.

9.9.4. Random Forest

Random Forest affiche une précision parfaite (100%) sur l'ensemble d'en traînement, indiquant un ajustement complet aux données. Cependant, en validation croisée, la précision descend légèrement (99,31%) et reste très éle vée en découpage à 75% (99,36%), prouvant sa robustesse. Les valeurs de précision, rappel et F-mesure étant quasiment parfaites, ce modèle montre une capacité de généralisation impressionnante.

Conclusion: Random Forest est le modèle le plus performant, offrant une classification presque parfaite avec une excellente généralisation.

9.9.5. IBK (KNN)

IBK (K-Nearest Neighbors) montre une précision parfaite (100%) sur l'en semble d'entraînement, suggérant un fort surapprentissage. En validation croisée, la précision baisse à 91,51% avec une F-mesure de 0,956, ce qui indique une sensibilité aux variations des données. Le découpage à 75% montre une légère amélioration (91,94%), mais reste inférieur aux autres modèles comme J48 ou Random Forest.

Conclusion : KNN est puissant mais sensible aux variations des données, ce qui le rend moins robuste en validation croisée.

9.10. Conclusion: Quelle est la meilleure méthode et le meilleur algorithme?

Parmi les différentes méthodes d'évaluation, la validation croisée est la plus fiable. Elle permet de mieux tester la capacité des modèles à bien fonc tionner sur de nouvelles données, évitant ainsi les pièges du surapprentissage.

En ce qui concerne les algorithmes, Random Forest et J48 (C4.5) sont les grands gagnants. J48 atteint une précision impressionnante de 99,57% en validation croisée, tandis que Random Forest reste extrêmement perfor mant avec 99,31%. Cela montre qu'ils sont capables de bien généraliser sur différents ensembles de données.

Si l'on devait choisir un seul modèle, Random Forest serait le meilleur choix, car il combine précision, robustesse et excellente généralisa tion, tout en étant plus flexible face aux variations des données.

9.11. Experimenter

9.11.1. Introduction

L'outil Experimenter de WEKA permet de comparer plusieurs algo rithmes d'apprentissage automatique en réalisant des expériences reproduc tibles. Cet outil est essentiel pour obtenir des évaluations statistiques précises et comparer différents classificateurs.

9.11.2. Pourquoi utiliser Experimenter?

- Comparer plusieurs algorithmes sur divers ensembles de données.

- Obtenir des tests statistiques (ex. test de Student) pour évaluer les différences de performance.

- Exécuter plusieurs tests en une seule fois pour automatiser l'évaluation.

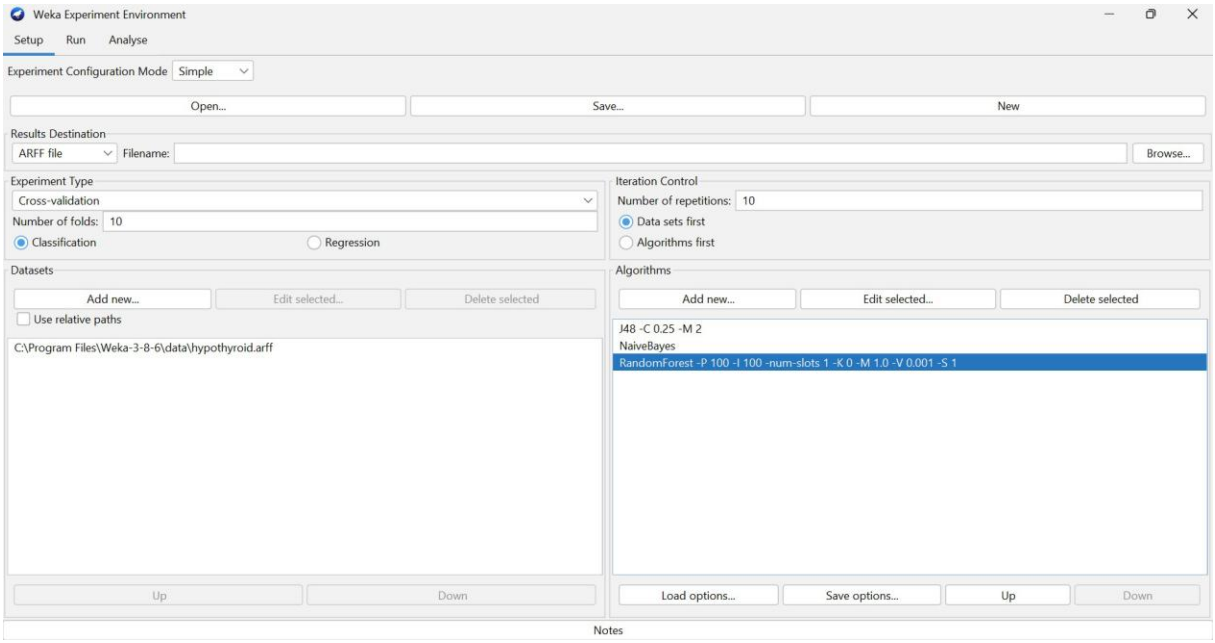


Figure (III.03) : *Interface Expérimenter de Weka GUI Chooser*

9.11.3. Étapes d'utilisation de Expérimenter

9.11.4. Ouvrir WEKA

Lancez WEKA et sélectionnez l'onglet **Expérimenter**.

9.11.5. Créer une nouvelle expérience

Cliquez sur **New** pour démarrer une nouvelle expérience.

9.11.6. Configurer l'expérience

- **Mode :** Sélectionnez *Cross-validation* pour une évaluation robuste.
- **Num Folds :** Utilisez *10* (par défaut, 10-fold cross-validation est une bonne pratique).

9.11.7. Ajouter les algorithmes à tester

- Allez dans l'onglet **Algorithms**.
- Cliquez sur **Add New** et sélectionnez les classificateurs à comparer (ex. J48, Naïve Bayes, Random Forest, IBK).
- Modifiez les paramètres si nécessaire en cliquant sur Edit.

9.11.8. Sélectionner le dataset

- Allez dans l'onglet **Datasets**.
- Cliquez sur **Add New** et chargez votre fichier *.arff* ou un autre format compatible.
- j'ai utilser la même base de donnée *hypothyroid.arff* dans explorer

9.11.9. Démarrer l'expérience

- Allez dans l'onglet Run.
- Cliquez sur Start et attendez l'exécution des tests.

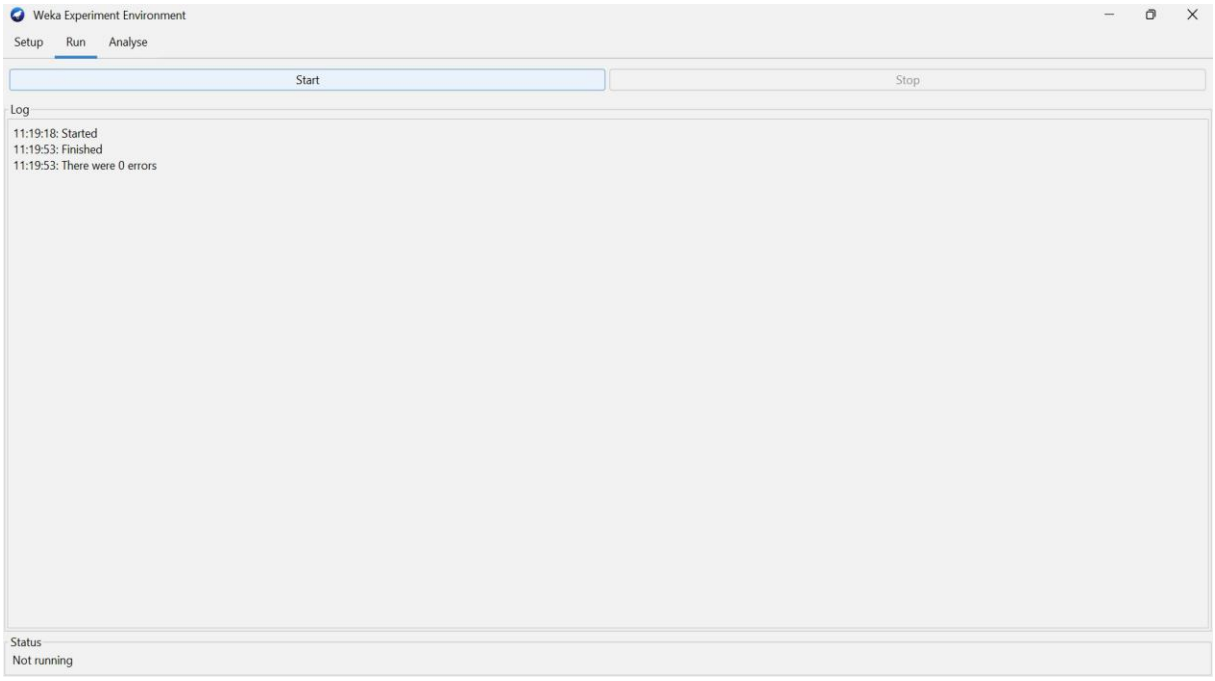


Figure (III.04): *Interface run de exemple*

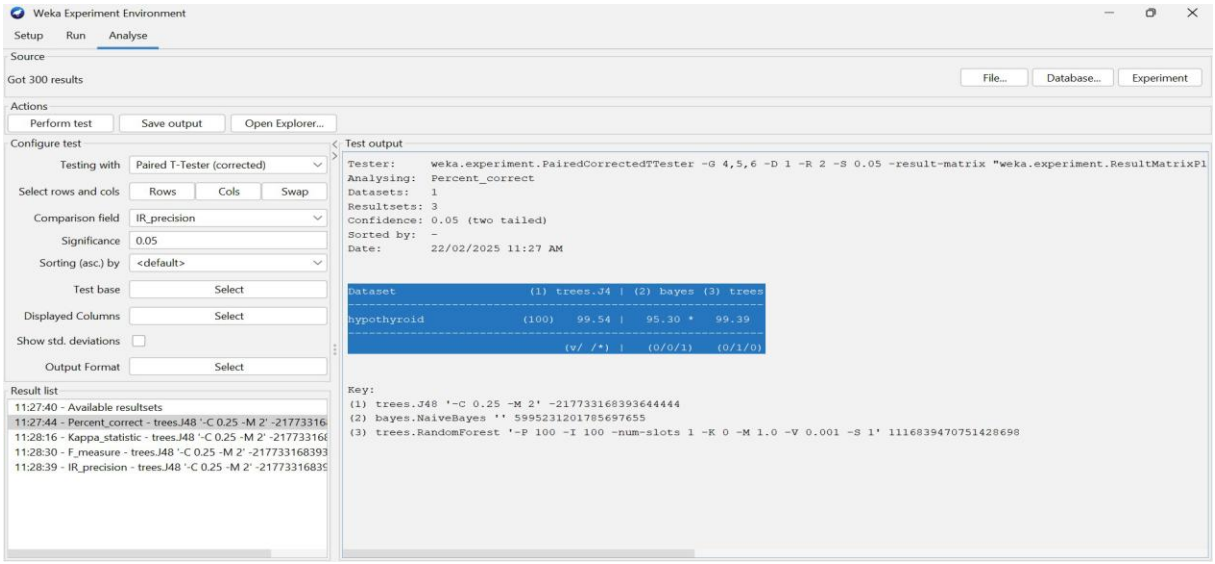


Figure (III.05): *Interface analyse de exemple*

9.11.10. Analyser les résultats

- Passez à l'onglet Analyse.

- Cliquez sur Perform test pour comparer les performances des algo rithmes.

- Les résultats afficheront des mesures comme la précision, le rappel et la F-mesure.

9.11.11. Exemple de comparaison

Si vous testez **J48, Naïve Bayes et Random Forest,** les résultats pourraient être les suivants :

Algorithme	Pourcentage Correct (%)	Kappa	F-mesure
J48	99.54	0.97	1.00
Naïve Bayes	95.30	0.60	0.98
Random Forest	99.39	0.96	1.00

Tableau (III.05): *Les résultats des testes*

9.11.12. Conclusion

L'outil **Experimenter** de WEKA permet de comparer objectivement plu sieurs algorithmes de classification en appliquant des méthodes d'évaluation rigoureuses comme la validation croisée.

Les résultats obtenus montrent que J48 et Random Forest offrent les meilleures performances avec un taux de précision élevé (99.54% et 99.39%) et une F-mesure parfaite (1.00), indiquant une excellente capa cité de généralisation et de classification. Naïve Bayes, bien que performant (95.30% de précision), présente une F-mesure plus faible (0.98) et un kappa de 0.60, suggérant qu'il est plus sensible aux caractéristiques des données.

En conclusion, Random Forest et J48 apparaissent comme les choix les plus robustes pour ce jeu de données, tandis que Naïve Bayes reste une alternative rapide et efficace malgré une légère perte de précision.

9.12. KnowledgeFlow

KnowledgeFlow est une interface alternative à l'Explorateur de WEKA, permettant une approche visuelle et modulaire pour le prétraitement des données, l'apprentissage des modèles et leur évaluation.

9.12.1. Pourquoi utiliser KnowledgeFlow ?

Contrairement à l'Explorateur, KnowledgeFlow permet de :

- Construire un workflow graphique pour les étapes de traitement des données.
- Manipuler facilement plusieurs fichiers de données et classificateurs dans une interface visuelle.
- Exécuter plusieurs expérimentations en parallèle, ce qui peut être plus efficace pour comparer des modèles.

9.12.2. Utilisation de KnowledgeFlow

1. **Ouvrir WEKA** et sélectionner l'onglet **KnowledgeFlow**.
2. **Ajouter des composants** :
   - Chargement des données (*ArffLoader* pour fichiers ARFF).
   - Prétraitement (*Filter* pour la normalisation, transformation des don nées, etc.).
   - Sélection d'un algorithme (*Classifier* comme J48, Naïve Bayes ou Random Forest).
   - Validation (*Evaluation* pour tester le modèle).
   - Visualisation des résultats (*TextViewer*, *GraphViewer*).

3. **Lier les composants** en connectant les différentes étapes du workflow.

4. **Exécuter l'expérience** et analyser les résultats j'ai utiliser la meme base de donne, algo : c4.5, méthode d'évaluation :cross validation .

a. voici le résultat :

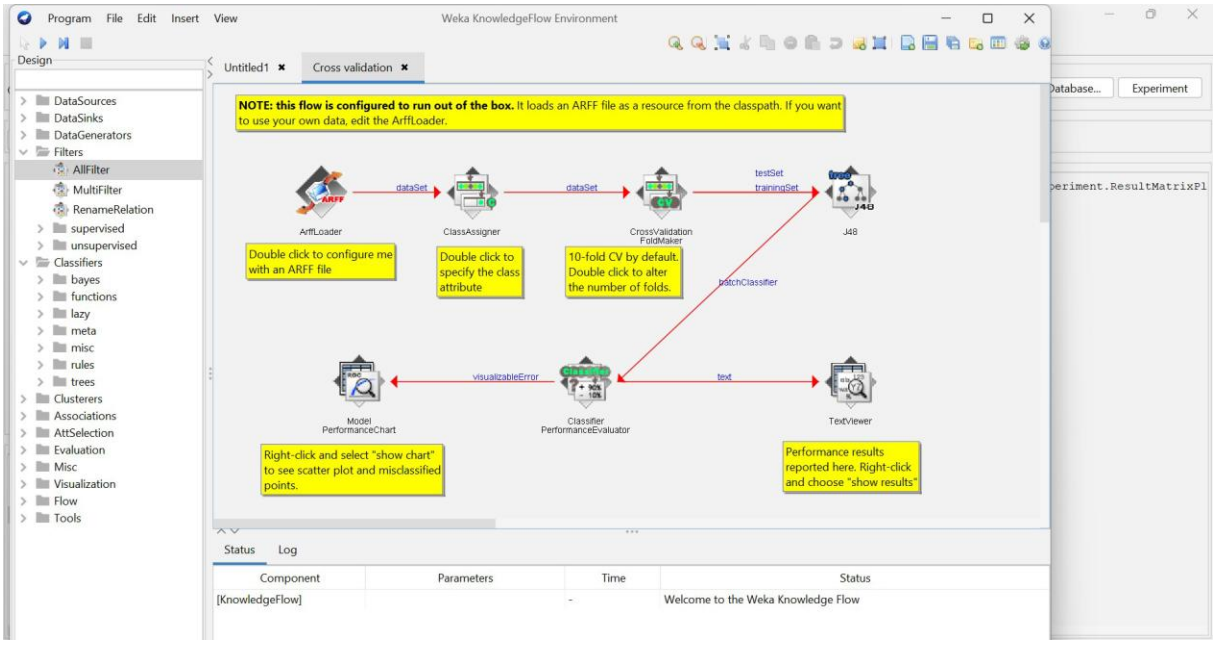


Figure (III.06): *Interface KnowledgeFlow de Weka GUI Chooser*

9.12.3. Avantages et inconvénients

Avantages :

- Interface visuelle intuitive sans avoir besoin de programmation. — Possibilité d'exécuter plusieurs tâches simultanément.
- Meilleure compréhension du flux de traitement des données.

Inconvénients :

- Moins flexible que l'interface en ligne de commande.
- Peut être plus complexe pour les utilisateurs habitués à l'Explorateur de WEKA.

9.12.4. Conclusion

KnowledgeFlow est un outil puissant de WEKA qui facilite la gestion et l'expérimentation des modèles de classification grâce à une approche modu laire et interactive. Il est particulièrement utile pour les utilisateurs préférant une interface graphique au lieu de la ligne de commande ou de l'Explorateur

classique. Cependant, il peut être moins efficace pour des tâches nécessitant une configuration fine des algorithmes.

En résumé, KnowledgeFlow est un excellent choix pour visuali ser et expérimenter facilement différents flux de traitement, bien qu'il puisse nécessiter une période d'adaptation pour les utilisateurs avancés habitués aux autres interfaces de WEKA.

9.13. Workbench

WEKA Workbench est une interface complète qui regroupe tous les ou tils de WEKA en un seul environnement. Il permet aux utilisateurs d'accéder aux fonctionnalités de l'Explorer, de KnowledgeFlow, de l'Experimenter et à la ligne de commande via une interface unifiée.

9.13.1. Pourquoi utiliser Workbench ?

Workbench est conçu pour offrir une expérience centralisée et facilite le passage entre différentes méthodes d'analyse de données. Il permet de:

- Accéder à toutes les fonctionnalités de WEKA depuis un seul environ nement.
- Basculer facilement entre Explorer, Experimenter, Knowledge Flow et la ligne de commande.
- Bénéficier d'une interface plus moderne et mieux organisée.

9.13.2. Utilisation de Workbench

1. **Ouvrir WEKA** et sélectionner **Workbench** depuis l'écran d'accueil.
2. **Choisir l'outil désiré**
   - **Explorer** pour le prétraitement, la classification et la visualisation des résultats.
   - **KnowledgeFlow** pour concevoir un flux de travail graphique.
   - **Expérimenter** pour comparer plusieurs modèles en effectuant des tests rigoureux.
   - **Ligne de commande** pour une exécution plus flexible et automatisée des algorithmes.
3. **Charger un ensemble de données** et sélectionner un algorithme d'apprentissage.

4. Exécuter le modèle et analyser les résultats.

9.13.3. Avantages et inconvénients

Avantages :

- Interface tout-en-un, évitant de passer d'un outil à un autre.
- Plus intuitive et moderne que les anciennes interfaces séparées.
- Idéale pour les débutants comme pour les utilisateurs avancés.

Inconvénients :

- Peut sembler plus complexe au premier abord en raison du grand nombre d'options disponibles.
- L'utilisation intensive peut nécessiter plus de ressources système.

9.13.4. Exemples de 3 alogorithm par la cross_validation (10)

Métriques	1R	J48	IBK
TP Rate	0.980	0.999	0.964
FP Rate	0.093	0.021	0.619
Precision	0.992	0.998	0.949
Recall	0.980	0.999	0.964
F-Measure	0.986	0.998	0.956
MCC	0.833	0.979	0.379
ROC Area	0.944	0.993	0.682
PRC Area	0.991	0.999	0.950

Tableau (III.06): *Exemple d'execution*

9.13.5. Conclusion

WEKA Workbench est une évolution naturelle de WEKA, offrant un accès centralisé à tous ses outils en une seule interface. Son principal avantage est de faciliter le passage entre différentes approches d'apprentis sage automatique sans devoir ouvrir plusieurs fenêtres.

En résumé, Workbench est l'option idéale pour les utilisateurs cherchant une expérience complète et fluide, bien qu'il puisse nécessiter un temps d'adaptation pour ceux qui préfèrent les anciennes interfaces séparées.

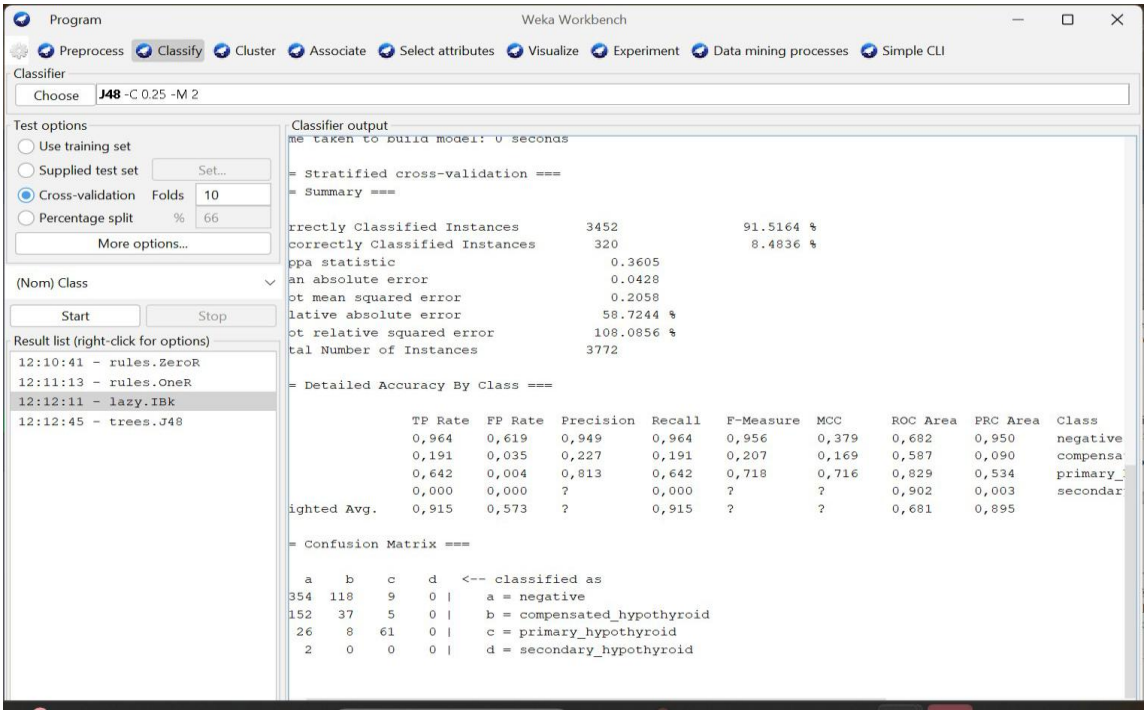


Figure (III.07): *Interface de Workbench : exemple IBK (knn)*

9.14. Simple CLI

La fenêtre Simple CLI (Command Line Interface) de Weka est un outil puissant permettant d'interagir avec Weka via des commandes textuelles au lieu de l'interface graphique. Cette interface est idéale pour les utilisateurs avancés ou ceux qui souhaitent automatiser des tâches de machine learning avec des scripts.

9.14.1. Comment fonctionne la fenêtre Simple CLI ?

1. **Lancer la fenêtre Simple CLI** : Vous pouvez ouvrir la fenêtre **Simple CLI** en lançant Weka et en sélectionnant "Simple CLI" dans le menu principal de l'interface graphique.

2. **Entrée des commandes** : Une fois que la fenêtre est ouverte, vous tapez des commandes pour effectuer des tâches spécifiques comme l'en traînement de modèles, l'évaluation, ou l'exécution de traitements sur les données. Les commandes sont tapées dans la zone de texte de la fenêtre CLI, et les résultats sont affichés dans la même fenêtre.

3. **Structure des commandes :** La syntaxe des commandes suit une structure de base, par exemple :

java weka.classifiers.trees.J48 -t data.arff -x 10

Cette commande indique à Weka d'utiliser le classificateur J48 (un arbre de décision), de charger le fichier data.arff pour l'entraînement, et de réaliser une validation croisée à 10 plis.

4. **Exécution des tâches :** Après avoir tapé une commande, vous ap puyez sur Entrée, et Weka exécute la tâche demandée. Les résultats s'affichent dans la même fenêtre. Par exemple, pour une validation croisée, Weka affichera les résultats d'évaluation du modèle, les métriques comme la précision, le rappel, la F-mesure, etc.

9.14.2. Avantages de la fenêtre Simple CLI

1. **Interface légère et rapide :** Simple CLI est moins gourmande en ressources système comparée à l'interface graphique de Weka, ce qui permet de traiter de grands ensembles de données plus rapidement.
2. **Automatisation via des scripts :** Vous pouvez automatiser des tâches en utilisant des scripts. Par exemple, en exécutant plusieurs commandes dans un fichier script .txt, ce qui permet d'enchaîner différentes étapes d'analyse sans intervention manuelle.
3. **Plus de contrôle sur les processus :** La fenêtre CLI offre un contrôle plus direct sur les commandes et les options disponibles, ce qui permet une personnalisation fine des analyses.
4. **Utilisation dans des environnements sans interface graphique :** La CLI est très utile dans des environnements serveurs ou lorsqu'il n'y a pas d'interface graphique disponible, comme sur des serveurs Linux.
5. **Plus de flexibilité :** Vous pouvez utiliser des commandes avancées et des options supplémentaires pour modifier le comportement des algorithmes, ajouter des matrices de coûts, ajuster les critères d'évaluation, etc.

9.14.3. Inconvénients de la fenêtre Simple CLI

1. **Courbe d'apprentissage :** Pour les débutants, la CLI peut être difficile à prendre en main. Vous devez connaître les commandes et leur syntaxe. Contrairement à l'interface graphique, il n'y a pas d'options disponibles à cliquer.

2. **Pas de visualisation directe :** Contrairement à l'interface graphique de Weka, la CLI ne propose pas de visualisation directe des données, des modèles ou des résultats, ce qui peut rendre l'analyse plus difficile, surtout pour les utilisateurs visuels.
3. **Erreurs de syntaxe :** L'absence de validation automatique des com mandes peut entraîner des erreurs de syntaxe difficiles à diagnostiquer, surtout pour les utilisateurs novices. L'interface graphique est plus to lérante dans ce domaine.
4. **Pas d'outils interactifs :** La CLI ne propose pas de mécanismes interactifs comme les outils de visualisation ou les graphiques qui peuvent être très utiles pour comprendre et interpréter les modèles et les résultats.

9.14.4. Informations importantes dans la fenêtre Simple CLI :

1. **Commandes disponibles :** Vous pouvez utiliser la commande help pour obtenir une liste complète des commandes disponibles et des options spécifiques pour les classificateurs, les filtres, les évaluations, etc. Exemple de commande pour obtenir de l'aide :

   help

2. **Retour de l'exécution des commandes :** Les résultats s'affichent dans la fenêtre CLI après chaque exécution de commande. Ils incluent des informations comme les métriques de performance, les statistiques détaillées, les matrices de confusion, etc.
3. **Variables d'environnement :** Vous pouvez définir et utiliser des variables pour faciliter l'exécution des commandes répétitives. Par exemple :

   set inputFile="C:\data\train.arff"

   java weka.classifiers.trees.J48 -t ${inputFile} -x 10

4. **Contrôle d'exécution :** La CLI permet de gérer l'exécution des tâches, avec des commandes comme exit pour quitter l'interface ou kill pour stopper une exécution en cours.

9.14.5. Résumé

La fenêtre **Simple CLI** de Weka est idéale pour les utilisateurs avan cés qui préfèrent une approche basée sur des commandes pour effectuer des analyses et expérimentations en machine learning. Elle est rapide, flexible et

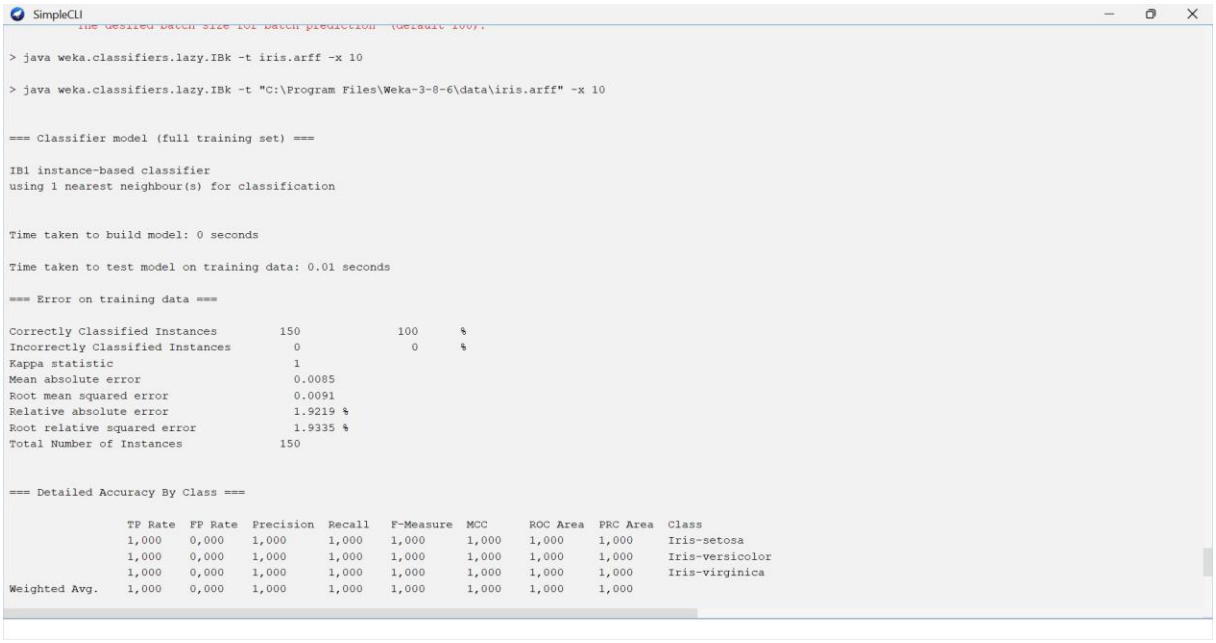


Figure (III.08): *Exemple d'excution IBK (knn) de la base de donnee aris.arff avec cross_validation (10)*

Permet une automatisation des processus grâce à des scripts. Cependant, elle nécessite une connaissance des commandes et n'offre pas les fonctionnalités visuelles et interactives que l'on trouve dans l'interface graphique de Weka.

Si vous êtes à l'aise avec les lignes de commande et souhaitez une méthode plus directe et rapide pour exécuter des tâches sur Weka, la CLI est un excellent choix !

10. Conclusion générale

Chaque mode d'interaction de Weka possède des avantages et des inconvénients, et le choix du mode dépend des besoins et des préférences de l'utilisateur :

- **Explorer** est idéal pour les utilisateurs débutants ou intermédiaires qui souhaitent explorer, visualiser et appliquer des algorithmes de machine learning de manière intuitive.

- **Experimenter** est préférable pour les utilisateurs qui cherchent à ef fectuer des évaluations de modèles comparatives et à tester plusieurs configurations de classificateurs sur des ensembles de données variés.
- **KnowledgeFlow** est recommandé pour ceux qui préfèrent une approche visuelle pour concevoir des workflows et traiter des données.
- **Workbench** est l'interface principale pour gérer des tâches basiques de machine learning, mais peut être moins intuitive que le mode Explorer.
- **CLI** est destiné aux utilisateurs avancés qui souhaitent automatiser et personnaliser entièrement leurs processus d'analyse.

En résumé, le mode Explorer est parfait pour les utilisateurs débutants grâce à sa simplicité d'utilisation. Le mode CLI est le meilleur pour ceux qui recherchent flexibilité et automatisation, tandis que Expérimenter et KnowledgeFlow offrent des solutions plus spécialisées adaptées aux comparaisons de modèles et à l'élaboration de workflows visuels.

11. Informations sur la version

Dans le coin inférieur gauche, on voit aussi des informations sur la version de Weka utilisée :

- Version 3.8.6
- Développé par l'Université de Waikato, Nouvelle-Zélande

12. Introduction sur La méthode Bootstrap 0.632

La méthode Bootstrap 0.632 est une technique d'évaluation de modèles statistiques utilisée pour estimer la performance d'un modèle d'apprentissage automatique. Elle appartient à la famille des méthodes de rééchantillonnage (resampling) et est particulièrement utile lorsqu'il y a peu de données disponibles ou lorsque l'on veut obtenir une estimation robuste de la performance d'un modèle.

12.1. Principe de la méthode

Le principe de la méthode *Bootstrap 0.632* repose sur un échantillon nage avec remplacement à partir des données d'entraînement pour créer plu sieurs sous-ensembles de données. Ensuite, le modèle est entraîné sur ces sous-ensembles et évalué sur les points de données non inclus dans ces sous ensembles (appelés *points de validation*).

Les étapes de la méthode sont les suivantes :

1. **Echantillonnage avec remplacement :** À partir de l'ensemble de données d'origine, des échantillons de la même taille que l'ensemble d'origine sont tirés avec remplacement.
2. **Création du sous-ensemble d'entraînement et de test :** Les échantillons sélectionnés dans le processus d'échantillonnage sont utilisés pour former un *sous-ensemble d'entraînement*. Les données non sélectionnées forment un *sous-ensemble de test* (également appelé validation).
3. **Calcul des scores :** Le modèle est entraîné sur le sous-ensemble d'en traînement, puis testé sur le sous-ensemble de test. Cette opération est répétée plusieurs fois (par exemple, 1000 fois).
4. **Estimation des performances :** Le score final de performance (par exemple, précision, erreur) est calculé en combinant les performances des différentes itérations en utilisant une pondération spéciale de 0.632 (63.2 %). Cette pondération combine les résultats du modèle évalué sur les points de données utilisés pour l'entraînement et ceux évalués sur les points de validation non sélectionnés.

12.2. Formule de la méthode 0.632

Le score final (ou erreur) de la méthode 0.632 est calculé comme suit :

$\text{Erreur}_{0.632} = 0.632 \times \text{Erreur d'entraînement} + 0.368 \times \text{Erreur de validation}$

Où :

* *Erreur d'entraînement* est l'erreur sur les échantillons utilisés pour l'en traînement.

* *Erreur de validation* est l'erreur sur les échantillons non utilisés pour l'entraînement (c'est-à-dire ceux qui sont dans le sous-ensemble de test).

12.3. Avantages de la méthode

- **Robustesse :** La méthode offre une estimation plus stable des performances du modèle par rapport à une simple validation croisée ou une séparation unique en ensembles d'entraînement et de test.
- **Utilisation de toutes les données :** En utilisant un sous-ensemble des données pour l'entraînement et un autre pour la validation, la mé thode permet d'utiliser efficacement toutes les données disponibles.

- **Peu sensible aux petites variations de données :** Elle est plus robuste aux variations dans les échantillons, ce qui permet d'éviter le surapprentissage (overfitting) et d'obtenir des évaluations plus fiables.

12.4. Inconvénients de la méthode

- **Calcul coûteux :** En raison de l'échantillonnage répété et de l'évaluation multiple du modèle, cette méthode peut être computationnellement coûteuse, notamment pour de grands ensembles de données ou des modèles complexes.

- **Biais:** Bien que la méthode soit robuste, elle peut encore présenter un biais si les données sont mal représentées ou si les échantillons sont fortement déséquilibrés.

12.5. Code d'implémentation

```python
import pandas as pd
import numpy as np
from scipy.io import arff
from sklearn.preprocessing import LabelEncoder

def load_arff_data(file_path):
    data, meta = arff.loadarff(file_path)
    df = pd.DataFrame(data)
    string_columns = df.select_dtypes([object]).columns
    for col in string_columns:
        df[col] = df[col].str.decode('utf-8')
    return df, meta

def bootstrap_632_sampling(data):
    n_samples = len(data)
    selected_indices = np.random.choice(n_samples, size=n_samples, replace=
        True)
    training_set = data.iloc[selected_indices].reset_index(drop=True)
    unique_selected = np.unique(selected_indices)
    not_selected = np.setdiff1d(np.arange(n_samples), unique_selected)
    test_set = data.iloc[not_selected].reset_index(drop=True)
    return training_set, test_set

def save_to_arff(data, meta, filename, relation_name="bootstrap_data"):
    with open(filename, 'w', encoding='utf-8') as f:
        f.write(f'@relation {relation_name}\n\n')
        for column in meta.names():
            attr_type = meta[column][0]
            if attr_type == 'nominal':
                values_str = '{' + ','.join(map(str, meta[column][1])) + '}
                    '
                f.write(f'@attribute {column} {values_str}\n')
            else:
                f.write(f'@attribute {column} {attr_type}\n')
        f.write('\n@data\n')
        for _, row in data.iterrows():
            line = []
            for column in meta.names():
                value = row[column]
                if meta[column][0] == 'nominal' and ',' in str(value):
                    value = f'"{value}"'
                line.append(str(value))
            f.write(','.join(line) + '\n')

def main():
    file_path = "C:\Program Files\Weka-3-8-6\data\hypothyroid.arff"

    np.random.seed(42)
    data, meta = load_arff_data(file_path)
    train_data, test_data = bootstrap_632_sampling(data)
    print("Taille des donnees originales:", data.shape)
    print("Taille des donnees d'entrainement:", train_data.shape)
    print("Taille des donnees de test:", test_data.shape)
    save_to_arff(train_data, meta, "bootstrap_train.arff", "
        bootstrap_training_data")
    save_to_arff(test_data, meta, "bootstrap_test.arff", "
        bootstrap_test_data")
    return train_data, test_data

if __name__ == "__main__":
    train_data, test_data = main()
```

Figure (III.09): *Code Python pour le Bootstrap 0.632*

12.6. Explication du code

- **load_arff_data(file_path) :** Cette fonction charge le fichier ARFF en utilisant le module scipy.io.arff, convertit les données en un DataFrame pandas, et décode les colonnes de type chaîne pour garantir un traitement correct des données textuelles.

- **bootstrap_632_sampling(data):** Cette fonction applique la mé thode Bootstrap 0.632. Elle sélectionne au hasard des échantillons avec remplacement du jeu de données pour créer un ensemble d'entraî nement. Les données non sélectionnées dans cet échantillonnage de viennent l'ensemble de test. Cette méthode permet d'évaluer la perfor mance du modèle sur différentes divisions des données.

- **save_to_arff(data, meta, filename, relation_name) :** Cette fonction enregistre les ensembles de données résultants (entraînement et test) dans de nouveaux fichiers ARFF. Les attributs (colonnes) sont écrits avec leurs types respectifs (par exemple, nominal, numérique) comme spécifié dans les métadonnées.

- **main() :** La fonction principale exécute l'ensemble du processus. Elle charge les données, applique la méthode d'échantillonnage Bootstrap, affiche les dimensions des ensembles résultants et les enregistre dans de nouveaux fichiers ARFF.

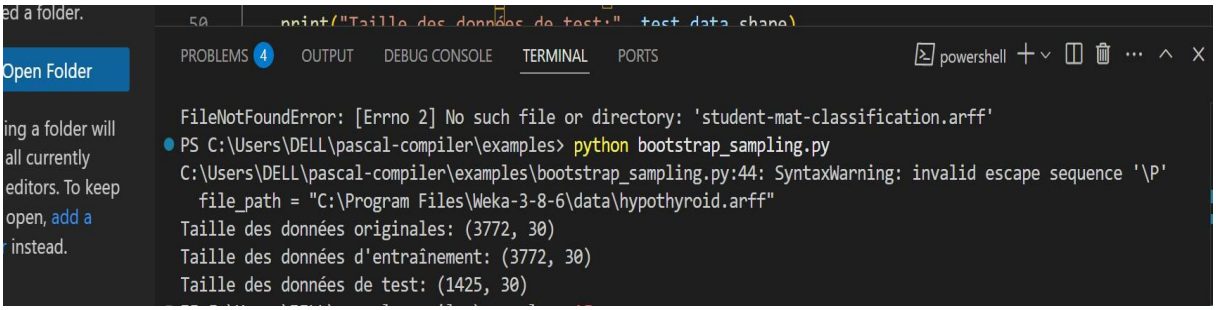


Figure (III.10): *Resultat de code*

Figure (III.11): *Resultat de code*

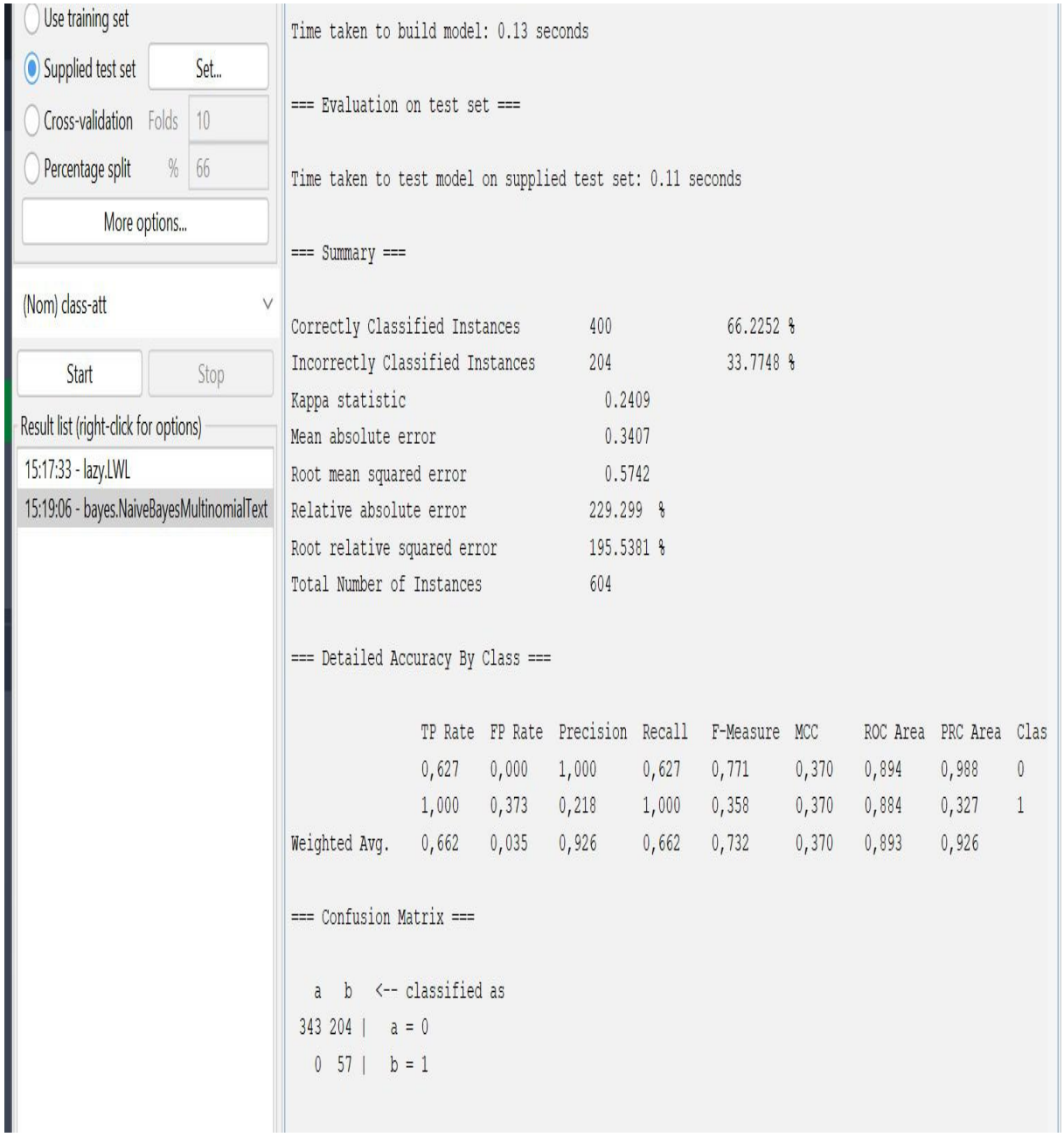


Figure (III.12): *Exemple d'execution mais avec une autre base de donnee avec la methode bootstab 0.632.*

13. Conclusion: Méthode Bootstrap 0.632 vs. autres approches

La méthode **Bootstrap 0.632** se démarque des méthodes classiques (va lidation croisée, hold-out, ou bootstrap standard) par sa pondération unique (*0.632*), qui intègre à la fois l'erreur d'entraînement et l'erreur de test pour corriger le biais d'optimisme. Alors que le bootstrap standard se concentre sur l'estimation de la variance et que la validation croisée évalue les performances via des sous-ensembles fixes, le Bootstrap 0.632 utilise une probabilité théo rique ($1-\frac{1}{e} \approx 63.2\%$) pour équilibrer surapprentissage et sous-apprentissage.

Contrairement au *hold-out*, souvent instable sur petits échantillons, ou au *k-fold*, dépendant de la segmentation des données, le Bootstrap 0.632 rééchantillonne avec remise, optimisant ainsi l'usage de données limitées. En revanche, pour les grands jeux de données, la validation croisée reste plus efficiente en calcul. Le choix final dépendra donc du volume de données, de la complexité du modèle, et du compromis entre précision et ressources disponibles.

CHAPITRE IV

Prétraitement Et Classification D'Hypothyroïdie

1. Introduction

Le prétraitement des données est une étape clé en data mining pour améliorer la qualité des datasets avant l'analyse. Ce chapitre explore différentes techniques avec WEKA, notamment la discrétisation, la gestion des valeurs manquantes et des outliers via l'IQR, ainsi que la sélection d'attributs. En appliquant ces méthodes sur un dataset réel, vous mesurerez leur impact sur les performances des algorithmes. L'objectif est de comprendre comment ces transformations influencent l'analyse des données. Une bonne maîtrise des concepts vus en cours est essentielle pour réussir ce chapitre.

2. Étape 01

2.1 Choix et Présentation du Dataset

2.1.1 Origine du Dataset

- **Nom du dataset :** Hypothyroid
- **Source:** UCI Machine Learning Repository
- **Créateur(s):** Garavan Institute
- **Année:** 1987

2.1.2 Contexte et Objectif

Ce dataset a été conçu pour aider au diagnostic de l'hypothyroïdie en analysant des données médicales telles que les niveaux d'hormones et les antécédents des patients. Il est souvent utilisé pour tester des algorithmes de classification et pour explorer la gestion des valeurs manquantes.

2.1.3 Structure du Dataset

- **Nombre d'instances:** 3 163
- **Nombre d'attributs:** 30 (y compris la classe cible)
- **Types d'attributs:** Numériques et catégoriels
- **Classes:** 4 (negative, compensated_hypothyroid, primary_hypothyroid, secondary_hypothyroid)

2.1.4 Description des Attributs

Le dataset contient des informations médicales sur la thyroïde. Voici la liste des attributs:

Nom	**Type**	**Rôle**	**Description**
Age	Numérique	Entrée	Âge du patient
Sex	Catégoriel	Entrée	Sexe (M/F)
on_thyroxine	Booléen	Entrée	Prend de la thyroxine
query_on_thyroxine	Booléen	Entrée	Suspicion de prise
on_antithyroid_medication	Booléen	Entrée	Prend antithyroïdien
Sick	Booléen	Entrée	Patient malade
Pregnant	Booléen	Entrée	Enceinte
thyroid_surgery	Booléen	Entrée	Chirurgie thyroïdienne
I131_treatment	Booléen	Entrée	Traitement iode-131
query_hypothyroid	Booléen	Entrée	Suspicion hypothyroïdie
query_hyperthyroid	Booléen	Entrée	Suspicion hyperthyroïdie
Lithium	Booléen	Entrée	Prend lithium
Goiter	Booléen	Entrée	Présence d'ungoitre
Tumor	Booléen	Entrée	Présence d'unetumeur
Hypopituitary	Booléen	Entrée	Hypopituitarisme
Psych	Booléen	Entrée	Problèmes psychiatriques
TSH	Numérique	Entrée	Niveau TSH
T3	Numérique	Entrée	Niveau T3
TT4	Numérique	Entrée	Niveau thyroxine total
T4U	Numérique	Entrée	Index thyroxine libre
FTI	Numérique	Entrée	Indice thyroxine libre
TBG	Numérique	Entrée	Globuline liant T4
referral_source	Catégoriel	Entrée	Source référence

TBG_measured	Booléen	Entrée	Mesure TBG faite
TSH_measured	Booléen	Entrée	Mesure TSH faite
T3_measured	Booléen	Entrée	Mesure T3 faite
TT4_measured	Booléen	Entrée	Mesure TT4 faite
T4U_measured	Booléen	Entrée	Mesure T4U faite
FTI_measured	Booléen	Entrée	Mesure FTI faite
Class	Catégoriel	Sortie	Hypothyroid / Négatif

Tableau (IV.01): *Les informations médicales sur la thyroïde*

2.2. Le pourcentage des valeurs manquantes

Voici un script Python permettant de charger un fichier .arff et de calculer le pourcentage des valeurs manquantes par colonne ainsi que le pourcentage global.

```
import pandas as pd
from scipy.io import arff

def calculate_missing_percentage(arff_file):
    # Charger le fichier ARFF
    data, meta = arff.loadarff(arff_file)

    # Convertir en DataFrame
    df = pd.DataFrame(data)

    # D coder les cha nes de caract res binaires
    for column in df.select_dtypes([object]):
        df[column] = df[column].str.decode('utf-8')

    # Calcul du pourcentage des valeurs manquantes par colonne
    missing_per_column = df.isnull().mean() * 100

    # Calcul du pourcentage global des valeurs manquantes
    total_missing_percentage = df.isnull().sum().sum() / (df.shape[0] *
   df.shape[1]) * 100

    print("Pourcentage des valeurs manquantes par colonne:")
    print(missing_per_column)
    print(f"\nPourcentage global des valeurs manquantes: {
   total_missing_percentage:.2f}%")

# Sp cifier le chemin du fichier ARFF
arff_file = r"C:\Program Files\Weka-3-8-6\data\hypothyroid.arff"
calculate_missing_percentage(arff_file)
```

Figure (IV.01): *Calcul des valeurs manquantes dans un fichier ARFF*

2.3. Résultats de code

Le tableau suivant présente le pourcentage des valeurs manquantes pour chaque colonne du dataset hypothyroid.arff :

Attribut	**Pourcentage de valeurs manquantes**
Age	0.03%
Sex	0.00%
On thyroxine	0.00%
Query on thyroxine	0.00%
On antithyroid medication	0.00%
Sick	0.00%
Pregnant	0.00%
Thyroid surgery	0.00%
I131 treatment	0.00%
Query hypothyroid	0.00%
Query hyperthyroid	0.00%
Lithium	0.00%
Goitre	0.00%
Tumor	0.00%
Hypopituitary	0.00%
Psych	0.00%
TSH measured	0.00%
TSH	9.78%
T3 measured	0.00%
T3	20.39%
TT4 measured	0.00%
TT4	6.12%

T4 Umeasured	0.00%
T4U	10.26%
FTI measured	0.00%
FTI	10.21%
TBG measured	0.00%
TGB	100.00%
referral source	0.00%
Class	0.00%

Tableau *(IV.*02): *Le pourcentage des valeurs manquantes pour chaque colonne*

Pourcentage global des valeurs manquantes : 5.23%.

3. Etape 02 : Évaluation Initiale des Performances

3.1 Remarque 01

La précision et le rappel et le F_mesure c'est les résultat de la class négative et j'ai oublié de la mentionnée dans le chapitre 2 (donc la comparaison elle est entre class négative seulement pour chaque algo)

3.2 Remarque 02

J'ai utiliser la même base de données de chapitre 2 par ce qu'elle Respecte les conditions et j'ai déjà fais cette comparaison donne le chapitre 2

3.3 OneR

Méthode d'Évaluation	CCI	Précision	Rappel	F-Mesure
Cross-Validation(10)	96.2354%	0.696	0.702	0.698
Percentage Split(80%)	95.756%	0,987	0,981	0,984

Tableau *(IV.03):* *Résultats pour OneR*

3.4 NaïveBayes

Méthode d'Évaluation	CCI	Précision	Rappel	F-Mesure
Cross-Validation(10)	95.281 %	0,961	0,993	0,977
Percentage Split(80%)	96.4191 %	0,972	0,993	0,982

Tableau *(IV.04):* *Résultats pour Naïve Bayes*

3.5. J48 (C4.5)

Méthode d'Évaluation	CCI	Précision	Rappel	F-Mesure
Cross-Validation(10)	99.5758 %	0,998	0,999	0,998
Percentage Split(80%)	99.3369 %	0,999	0,999	0,999

Tableau (IV.05): *Résultats pour J48 (C4.5)*

3.6. Random Forest

Méthode d'Évaluation	CCI	Précision	Rappel	F-Mesure
Cross-Validation(10)	99.3107%	0,949	0,997	0,997
Percentage Split(80%)	99.4695%	0,996	1,000	0,998

Tableau (III.06): *Résultats pour Random Forest*

3.7. IBK (KNN): k=1

Méthode d'Évaluation	CCI	Précision	Rappel	F-Mesure
Cross-Validation(10)	91.5164%	0,949	0,964	0,956
Percentage Split(80%)	92.1715%	0,955	0,967	0,961

Tableau (IV.07): *Résultats pour K-Nearest Neighbors (KNN)*

3.8. Justification du Choix des Algorithmes

Les algorithmes J48 (C4.5) et Random Forest sont les meilleurs choix pour la classification du dataset Hypothyroid, basés sur les résultats obtenus avec la méthode de validation Percentage Split (80%).

Performances supérieures :

J48 (C4.5) atteint un CCI de 99.34% et une **F-Mesure de 0.999**, indiquant une classification presque parfaite.

Random Forest obtient un **CCI de 99.47%** et une **F-Mesure de 0.998**, confirmant sa robustesse et sa précision.

Adaptabilité aux données médicales :

Le dataset Hypothyroid contient des attributs catégoriques et numériques.

J48 (C4.5) génère des règles sous forme d'arbre de décision, ce qui permet une interprétation facile, essentielle dans le domaine médical.

Random Forest, en combinant plusieurs arbres, améliore la robustesse et réduit les erreurs dues aux variations des données.

Capacité à gérer le bruit et les valeurs manquantes :

Les tests médicaux peuvent contenir des valeurs manquantes ou du bruit.

Les arbres de décision et les forêts aléatoires sont reconnus pour bien gérer ces problèmes, contrairement à des méthodes comme Naïve Bayes ou KNN, plus sensibles aux erreurs.

Comparaison avec les autres algorithmes :

Algorithme	CCI (%)	F-Mesure	Remarque
J48 (C4.5)	99.34	0.999	Meilleur équilibre précision/rappel
Random Forest	99.47	0.998	Très robuste, gère bien le bruit
Naïve Bayes	96.42	0.982	Moins performant pour ce dataset
OneR	95.75	0.984	Limité en classification
KNN (k=1)	92.17	0.961	Sensible au bruit

Tableau (IV.08): *Comparaison des performances des algorithmes.*

Conclusion: Grâce à leur forte précision, leur robustesse et leur capacité à gérer des données médicales complexes, J48 (C4.5) et Random Forest sont les meilleurs choix pour classifier le dataset Hypothyroid.

4. Étape 3: Expérimentations de Prétraitement avec WEKA

4.1. Normalisation et Standardisation

4.1.1. Définitions

Normalisation: La normalisation consiste à transformer les valeurs des attributs numériques afin de les ramener dans un intervalle spécifique, généralement [0, 1] ou [−1, 1].

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Standardisation: La standardisation transforme les valeurs des attributs pour qu'elles aient une moyenne de 0 et un écart-type de 1.

$$X_{stand} = \frac{X - \mu}{\sigma} \quad (2)$$

où μ est la moyenne et σ l'écart-type.

4.1.2. Importance de ces étapes

Équilibrer les échelles des variables : Évite qu'une caractéristique ayant une grande plage de valeurs domine les autres.

Améliorer la convergence des algorithmes : Facilite l'apprentissage pour des modèles comme la régression logistique ou les réseaux de neurones.

Optimiser les calculs de distance : Utile pour les algorithmes basés sur les distances (ex : KNN, K-Means, SVM).

Stabiliser les modèles : Réduit les biais liés aux unités de mesure différentes.

4.1.3 Application dans WEKA

Pour appliquer la normalisation ou la standardisation dans WEKA, suivez les étapes suivantes:

1. Ouvrir WEKA et charger un dataset :

- Lancer WEKA et aller dans Explorer.

- Charger le dataset en cliquant sur "Open file" et sélectionner un fichier au format .arff, .csv, etc.

2. Aller dans l'onglet "Preprocess" :

- Cet onglet permet d'appliquer les filtres de transformation des données.

3. Appliquer la discrétisation :

- Cliquer sur "Choose" dans la section "Filter".
- Sélectionner le filtre : weka -> filters -> unsupervised -> attribute -> Discretize

4. Configurer le filtre pour utiliser "Equal Width" ou "Equal Frequency" :

- Cliquer sur le filtre **Discretize** une fois sélectionné.
- Une fenêtre avec les options s'ouvre.
- **Equal Width** (Intervalles égaux) :
- Définir le nombre d'intervalles dans l'option **Bins** (ex : 10).
- Vérifier que Use **equal frequency** est désactivé.
- **Equal Frequency** (Fréquence égale) :
- Définir le nombre d'intervalles dans l'option **Bins**.
- **Activer l'option** Use equal frequency.

5. Appliquer et vérifier les résultats :

- Cliquer sur **Apply** pour appliquer la transformation.
- Vérifier la nouvelle distribution des valeurs dans l'aperçu des données.

4.2. Différence entre "Equal Width" et "Equal Frequency"

Equal Width (Intervalles égaux) : Divise les données en intervalles de même largeur.

Equal Frequency (Fréquence égale) : Crée des intervalles contenant un nombre similaire d'instances.

4.3. Utilisation de SupervisedDiscretize dans WEKA

4.3.1. Présentation

Le filtre SupervisedDiscretize est une méthode de discrétisation supervisée qui prend en compte la classe cible afin d'optimiser la séparation des valeurs continues.

4.3.2. Étapes d'utilisation dans WEKA

1. **Charger un dataset** dans WEKA via l'onglet **Explorer**.

2. **Accéder à l'onglet "Preprocess"** et sélectionner : Choose → weka

-> filters -> supervised -> attribute -> SupervisedDiscretize.

3. **Configurer les options** (ex : nombre d'intervalles, discrétisation basée sur l'entropie).

4. **Appliquer le filtre et vérifier** la transformation des attributs continus en intervalles discrets optimisés.

4.3.3 Avantages

- Optimisation des intervalles en fonction de la classe cible.
- Amélioration des performances des modèles comme Naïve Bayes.
- Utilisation de l'entropie pour une discrétisation plus efficace.

4.3.4 Etape 02 : Évaluation des Performances avec la Discrétisation (Validation croisée 10 folds) :

Discrétisation	**Méthode**	**Intervalles (bins)**	**Algorithme**	**CCI (%)**	**Précision**	**F-Mesure**
Aucune (brut) Aucune (brut)	- -	- -	J48 Random Forest	99.5758 99.3107	0,998 0.997	0.998 0.997
Non supervisée	Equal Width	4 6 10	J48 J48 J48	92.789 93.0541 93.2662	0,928 0,941 0,934	0,963 0,964 0,965
	Equal Frequency	10	J48	98.2768	0,995	0,997
Supervisée	SupervisedDiscretize	-	J48	99.5758	0,998	0,998
Non supervisée	Equal Width	4 6 10	Random Forest Random Forest Random Forest	92.1262 93.3192 92.5769	0,928 0,943 0,943	0,959 0,966 0,966
	Equal Frequency	10	Random Forest	94.9099	0,956	0,977
Supervisée	SupervisedDiscretize	-	Random Forest	99.0986	0.994	0.995

Tableau (IV.09): *Résultats obtenus après discrétisation (Validation croisée 10 folds)*

4.3.5. Analyse des Résultats de la Discrétisation

Résultats Sans Discrétisation (Données Brutes) :

- J48 : CCI = 99.576%, Random Forest : CCI = 99.311%.

- F-Mesure élevée (≈ 0.998).

- Les performances sont déjà très élevées sans transformation.

Discrétisation Non Supervisée :

- Equal Width (J48) : La performance est nettement inférieure aux données brutes (CCI entre 92.8% et 93.3%).

- Equal Width (Random Forest) : Similaire à J48, avec une perte de performance (92.1% - 93.3%).

- Equal Frequency (J48 et Random Forest) : Une nette amélioration avec J48 (CCI = 98.277%) et Random Forest (CCI = 94.910%).

Discrétisation Supervisée :

- J48 (SupervisedDiscretize) : Meilleure précision (CCI = 99.576%), identique aux données brutes.

- Random Forest (SupervisedDiscretize) : Très bon score (CCI = 99.099%), légèrement inférieur aux données brutes.

4.3.6 Quelle méthode booste le plus les algorithmes ?

- J48 : La discrétisation supervisée SupervisedDiscretize donne les meilleurs résultats (CCI = 99.576%), équivalent aux données brutes.

- Random Forest : Les performances restent élevées avec SupervisedDiscretize

(CCI = 99.099%), mais la discrétisation non supervisée réduit les performances.

Conclusion: La discrétisation supervisée (SupervisedDiscretize) améliore légèrement ou maintient les performances optimales, tandis que la discrétisation non supervisée peut parfois réduire l'efficacité des algorithmes.

4.4. Traitement des Valeurs Manquantes

4.4.1. Définition du Traitement des Valeurs Manquantes

Les valeurs manquantes dans une base de données correspondent à des entrées absentes pour certaines variables. Elles peuvent être dues à des erreurs de saisie, des capteurs défectueux ou des restrictions d'accès aux données. Le traitement des valeurs manquantes consiste à gérer ces valeurs pour éviter les biais et améliorer la performance des algorithmes d'apprentissage.

4.4.2. Importance du Traitement des Valeurs Manquantes

Ne pas traiter les valeurs manquantes peut entraîner plusieurs problèmes :

- Biais dans les résultats : Certaines classes peuvent être sous-représentées.

- Réduction de la précision : Certains algorithmes ne supportent pas les valeurs manquantes.

- Perte d'information : Supprimer des instances peut réduire la taille de l'échantillon.

- Problèmes avec certains modèles : Certains algorithmes nécessitent un prétraitement spécifique.

4.4.3. Application du Traitement des Valeurs Manquantes dans WEKA

WEKA propose plusieurs méthodes accessibles via l'onglet Preprocess.

4.4.4. Suppression des Instances avec Valeurs Manquantes

1. Ouvrir WEKA Explorer et charger la base de données.

2. Aller à l'onglet Preprocess.

3. Sélectionner le filtre
weka.filters.unsupervised.instance.RemoveWithMissingValues.

4. Appliquer le filtre pour supprimer les instances contenant des valeurs manquantes.

4.4.5. Remplacement par la Moyenne ou le Mode

1. Dans l'onglet Preprocess, choisir le filtre ReplaceMissingValues.

2. Appliquer le filtre :

- Par la moyenne (pour les attributs numériques).

- Par le mode (valeur la plus fréquente) pour les attributs catégoriques.

4.4.6. Utilisation d'un Algorithme d'Imputation Avancé

1. Utiliser un algorithme comme k-Nearest Neighbors (k-NN) pour estimer les valeurs manquantes.

2. WEKA ne propose pas directement l'imputation par k-NN, mais il est possible d'utiliser des outils externes comme Python (scikit-learn).

4.5. Quelle Méthode Choisir ?

- Suppression : Si peu de valeurs manquent et que l'échantillon reste représentatif.

- Remplacement par la moyenne ou le mode : Approprié pour un prétraitement rapide et efficace.

- Imputation avancée (k-NN, régression) : À privilégier si les valeurs manquantes sont nombreuses et dépendent d'autres variables.

4.5.1 Évaluation des Performances avec le Traitement des Valeurs Manquantes (Validation croisée 10 folds)

Méthode de Traitement	Algorithme	CCI (%)	Précision	Rappel	F-Mesure
Aucune (brut)	J48	99.575	0.998	0.999	0.998
Aucune (brut)	Random Forest	99.310	0.997	0.997	0.997
ReplaceMissing Values	J48	99.6023	0.999	0.999	0.999
ReplaceMissing Values	Random Forest	99.3107	0.997	0.997	0.997
RemoveWithValues	J48	0.00	0.00	0.00	0.00
RemoveWithValues	Random Forest	0.00	0.00	0.00	0.00

Tableau (IV.10): *Résultats obtenus après traitement des valeurs manquantes (Validationcroisée 10 folds)*

4.5.2. Comparaison avec les données brutes (sans traitement)

- **J48** obtient 99.575% de classification correcte (CCI) avec une F Mesure de **0.998**.

- **Random Forest** atteint **99.310%** avec une F-Mesure de **0.997**.

- Ces valeurs montrent une très bonne performance, même sans traite ment des valeurs manquantes.

4.5.3. Effet du Filtre ReplaceMissingValues

- **J48** affiche une légère amélioration : **CCI = 99.602%, Précision/F Mesure = 0.999.**

- **Random Forest** reste presque inchangé (**CCI = 99.310%,** même Précision et F-Mesure).

- Cela signifie que **remplacer les valeurs manquantes n'a pas d'impact significatif sur Random Forest**, mais améliore légèrement J48.

4.5.4. Effet du Filtre RemoveWithValues

- Les performances chutent à **0% pour tous les indicateurs** (CCI, Précision, Rappel et F-Mesure) pour **J48 et Random Forest.**

- Cette chute drastique s'explique par la présence d'un attribut **TGB** contenant uniquement des valeurs manquantes.

- En appliquant RemoveWithValues, **toutes les instances contenant des valeurs manquantes sont supprimées. Or, puisque TGB est entièrement vide**, cela entraîne la suppression de l'intégralité de la base de données.

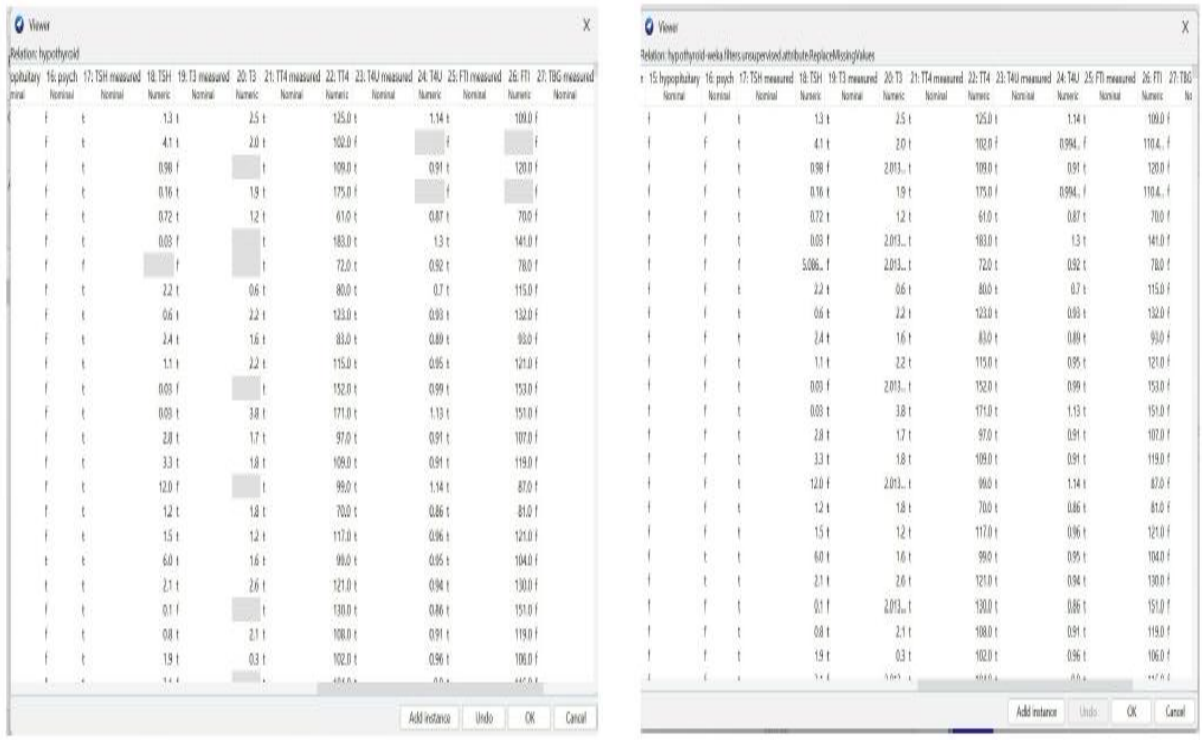
- Par conséquent, aucun modèle ne peut être entraîné, d'où les résultats nuls observés.

4.6 Conclusion

- Le remplacement des valeurs manquantes (ReplaceMissingValues) est préférable, car il préserve les données et améliore légèrement les performances.

- La suppression des instances (RemoveWithValues) est à éviter, car elle entraîne une perte totale des performances lorsque certaines variables sont entièrement manquantes.

- Random Forest est plus robuste aux valeurs manquantes, car ses résultats restent relativement stables avec ou sans remplacement des valeurs.



Avant traitement Après traitement

Figure (IV.02): *Comparaison des résultats avant et après le traitement (Effet du Filtre ReplaceMissingValues)*

4.7. Gestion des Outliers avec IQR

4.7.1. Introduction

La détection et la gestion des valeurs aberrantes (*outliers*) sont essen tielles en apprentissage automatique. Une méthode courante est l'utilisation de l'Interquartile Range (IQR).

L'Interquartile Range (IQR) permet d'identifier les outliers en se ba sant sur les quartiles :

- $Q1$: Premier quartile (25% des données).
- $Q3$: Troisième quartile (75% des données).
- $IQR = Q3 - Q1$: Étendue interquartile.

Une valeur est considérée comme un outlier si elle est en dehors de l'intervalle suivant:

$$[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR] \quad (3)$$

Les valeurs extrêmes sont en dehors de :

$$[Q1 - 3 \times IQR, Q3 + 3 \times IQR] \quad (4)$$

4.7.2. Application dans WEKA

4.7.3. Détection des Outliers avec *InterquartileRange*

1. Ouvrir WEKA et charger le dataset.
2. Aller dans Filters > unsupervised > attribute > InterquartileRange.
3. Configurer les options :

   outputColumName: Ajouter une colonne indiquant les outliers.
   detectionMethod: Choisir "Interquartile Range".
   ExtremeValuesFactor: Fixer à 1.5 pour les outliers standards, 3 pour les valeurs extrêmes.

4. Appliquer le filtre.

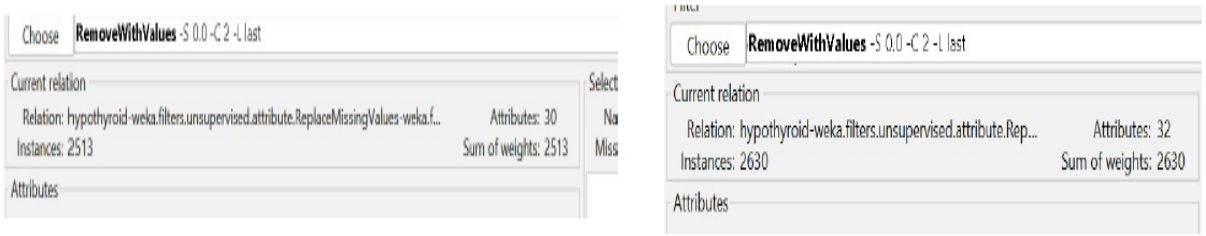
4.7.4. Suppression des Outliers avec *RemoveWithValues*

1. Aller dans Filters > unsupervised > instance > RemoveWithValues.
2. Sélectionner la colonne générée par *InterquartileRange*.
3. Spécifier la suppression des valeurs aberrantes.
4. Appliquer le filtre.

4.7.5. Évaluation des Performances avec Gestion des Outliers avec IQR(Validation croisée 10 folds)

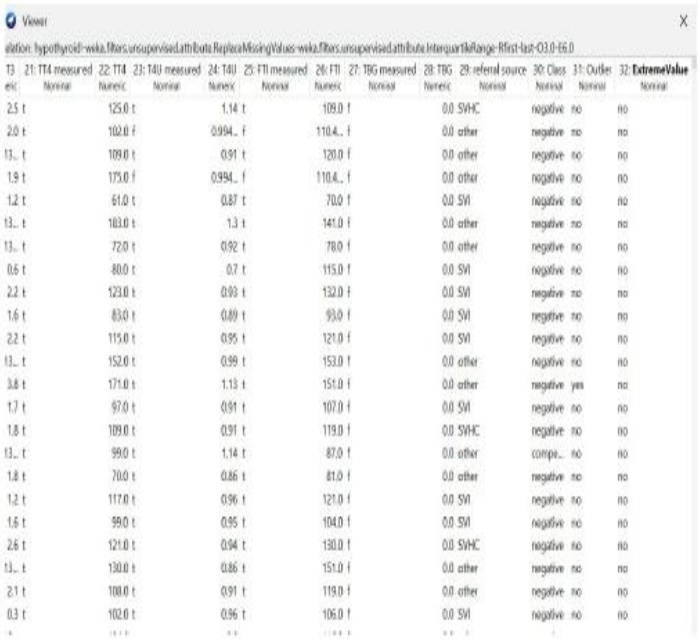
Méthode	nbr Instances	Algorithme	CCI%	Précision	F-Mesure
Sans outliers	3772	J48 Random Forest	99.5758 % 99.3107%	0,998 0,997	0,998 0,997
IQR (1.5) + Replacemissing Values	2630	J48 Random Forest	99.5057% 99.4677 %	0,999 0,999	0,998 0,998
IQR (3) + Replacemissing Values	2513	J48 Random Forest	99.7612 % 99.3633 %	1,000 0,998	0,999 0,998

Tableau (IV.11): *Comparaison des performances avant et après suppression des outliers.*



IQR 3.0 IQR 1.5

Figure (IV.03): *Le nombre des instances après l'appliquation d 'IQR*



Base de données après l'application de IQR

Supprimée les deux dernier attributs pour l'évaluation

Figure (IV.04): *Méthode IQR*

4.7.6. Analyse des résultats

4.7.7. Impact de la suppression des outliers sur le nombre d'instances

- **Sans suppression des outliers :** 3 772 instances.

- IQR (1.5) + ReplacemissingValues : 2 630 instances restantes (30.3% des données supprimées).

- **IQR (3) + ReplacemissingValues :** 2 513 instances restantes (33.4% des données supprimées).

Plus le seuil IQR est élevé, plus d'instances sont supprimées, réduisant ainsi la taille du dataset.

4.7.8 Comparaison des performances des modèles

Méthode	J48 (Précision / F-Mesure)	Random Forest (Précision / F-Mesure)
Sans outliers	0.998 / 0.998	0.997 / 0.997
IQR (1.5) + ReplacemissingValues	0.999 / 0.998	0.999 / 0.998
QR (3) + ReplacemissingValues	1.000 / 0.999	0.998 / 0.998

Tableau (IV.12) : *Comparaison des performances avant et après suppression des outliers*

4.7.9. Effet de la suppression des outliers sur les performances

- J48 améliore significativement ses performances après suppres sion des outliers, atteignant une précision parfaite (1.000) avec IQR(3).

- Random Forest est plus stable : ses performances ne varient que légèrement, ce qui prouve sa robustesse face aux valeurs extrêmes. — Le taux de classification correcte (CCI) augmente légèrement après suppression des outliers :

- J48 atteint 99.76% avec IQR(3).

- Random Forest atteint 99.36% avec IQR(3).

4.7.10. Conclusion

- La suppression des outliers bénéficie surtout à J48, qui atteint une F-Mesure de 0.999.

- Random Forest est moins affecté, ce qui montre sa capacité à gérer les valeurs extrêmes sans nettoyage préalable.

- IQR(3) semble être le meilleur choix pour J48, mais réduit de 33% la taille du dataset.

Recommandation : Utiliser IQR(3) pour J48, mais pour Random Forest, la suppression des outliers n'est pas nécessairement bénéfique.

4.8. Sélection d'Attributs dans WEKA

La sélection d'attributs est une étape essentielle du prétraitement des données permettant d'éliminer les variables inutiles, de réduire la complexité du modèle et d'améliorer ses performances.

4.8.1 Méthodes de sélection d'attributs

Dans WEKA, il existe deux principales approches :

- Méthode par filtres (Filter)
- Méthode par wrappers (Wrapper)

4.8.2. Sélection par Filtre (Filter)

Les méthodes basées sur les filtres appliquent des tests statistiques ou heuristiques pour évaluer l'importance des attributs indépendamment de tout modèle d'apprentissage.

Exemple dans WEKA :

- **AttributeSelection** (disponible dans les filtres supervisés et non su pervisés).

- **Évaluateurs possibles :**

- **CfsSubsetEval :** Sélectionne les attributs ayant une forte corré lation avec la classe mais une faible redondance entre eux.

- InfoGainAttributeEval : Classe les attributs selon leur gain d'in formation.

- ReliefFAttributeEval : Évalue l'importance des attributs selon leur capacité à différencier des classes voisines.

Méthodes de recherche :

- BestFirst : Explore les sous-ensembles d'attributs de manière in telligente en avançant et revenant en arrière.

- Ranker : Classe les attributs un par un selon leur pertinence.

4.8.3. Sélection par Wrapper (Wrapper)

Contrairement aux filtres, les méthodes wrapper utilisent un algorithme d'apprentissage pour tester différentes combinaisons d'attributs et sélection ner celles qui maximisent les performances du modèle.

Exemple dans WEKA :

- WrapperSubsetEval : Évalue les sous-ensembles d'attributs en en traînant un modèle et en mesurant ses performances.

- Méthodes de recherche :

- GreedyStepwise : Ajoute ou enlève des attributs de manière itérative pour optimiser la sélection.

- Algorithmes couramment utilisés :

- J48 (Arbre de décision)

- Random Forest

- Naïve Bayes

4.8.4. Application sur WEKA

1. Sélection par filtre (Ex : CfsSubsetEval + BestFirst)

1. Ouvrir WEKA et aller dans "Select attributes".
2. Choisir AttributeSelection dans la liste des filtres.
3. Dans Evaluator, choisir CfsSubsetEval.
4. Dans Search Method, sélectionner BestFirst.
5. Appliquer le filtre et observer les attributs sélectionnés.

2. Sélection par Wrapper (Ex : WrapperSubsetEval + J48 + GreedyStepwise)

1. Ouvrir WEKA et aller dans "Select attributes".
2. Choisir WrapperSubsetEval comme évaluation.
3. Dans Search Method, sélectionner GreedyStepwise.
4. Dans Classifier, choisir J48.
5. Appliquer et comparer les résultats avec la méthode Filter.

4.8.5. Comparaison entre Filter et Wrapper

Critère	Méthode Filter	Méthode Wrapper
Vitesse	Rapide	Lent
Précision	Moins précise	Plus précise
Utilisation du modèle	Non	Oui
Robustesse	Bonne généralisation	Dépend de l'algorithme utilisé
Exemple WEKA	CfsSubsetEval + BestFirst	WrapperSubsetEval + J48

Tableau (IV.13): *Comparaison entre la sélection d'attributs par Filtre et par Wrapper*

4.8.6. Conclusion

- La méthode Filter est plus rapide et adaptée aux datasets volumi neux.

- La méthode Wrapper donne de meilleurs résultats mais est plus coû teuse en temps de calcul.

- Recommandation : Utiliser la méthode Filter pour un pré-traitement rapide et Wrapper si l'optimisation de la performance est une priorité.

4.8.7. Comparaison des performances des modèles

Méthode	Algorithme	CCI (%)	Précision	F-Mesure
Sans sélection	J48 Random Forest	99.5758% 99.3107%	0.998 0.997	0.998 0.997
Filter (CfsSubsetEval + BestFirst)	J48 Random Forest	99.5758% 99.3107%	0.998 0.997	0.998 0.997
Wrapper (WrapperSubsetEval + J48)	J48 Random Forest	99.5758% 99.3107%	0.998 0.997	0.998 0.997

Tableau (IV.14): *Comparaison des performances des méthodes de sélection d'attributs.*

4.8.8. Observations

Stabilité des performances : Les valeurs de CCI, Précision et F-Mesure restent identiques entre les trois approches, ce qui signifie que ni le filtrage des caractéristiques ni la méthode Wrapper n'ont amélioré (ni dégradé) la performance des modèles.

Comparaison entre J48 et Random Forest :

- J48 obtient un taux de classification légèrement supérieur à Random Forest (99.5758 % contre 99.3107 %), bien que l'écart soit minime.

- La précision et la F-Mesure suivent la même tendance : J48 offre des valeurs légèrement supérieures.

Impact de la sélection de caractéristiques :

Comme les performances restent inchangées avec ou sans sélection de caractéristiques, cela signifie que toutes les caractéristiques initiales sont déjà pertinentes et qu'il n'y a pas de bruit significatif dans les données.

4.8.9. Conclusion

La sélection de caractéristiques (Filter ou Wrapper) n'a pas d'effet significatif sur les performances des modèles. Les deux algorithmes (J48 et Random Forest) affichent des résultats très élevés, suggérant que l'ensemble des caractéristiques utilisées dans l'apprentissage est déjà optimisé pour la classification.

4.8.10. Absence d'impact de la sélection de caractéristiques

L'absence d'amélioration des performances après la sélection de caractéristiques peut être expliquée par plusieurs facteurs :

- **Données déjà optimisées :** Toutes les caractéristiques initiales étant pertinentes, leur suppression n'apporte aucun gain significatif.

- **Redondance minimale :** Si les variables sont non corrélées et essen tielles, la sélection ne modifie pas la qualité des prédictions.

- **Puissance des modèles :**

  - **Random Forest** gère bien les caractéristiques inutiles grâce à la sélection aléatoire des attributs à chaque nœud.

  - **J48** sélectionne automatiquement les variables les plus pertinentes lors de la construction de l'arbre.

- **Taille et complexité des données :** L'impact de la sélection est généralement plus marqué sur de grands ensembles de données avec des variables non informatives.

- **Méthodes de sélection utilisées :**

  - **Filter (CfsSubsetEval + BestFirst) :** Évalue la pertinence des caractéristiques avant l'apprentissage, mais sans effet si les variables sont déjà bien choisies.

  - **Wrapper (WrapperSubsetEval + J48) :** Fonctionne en fonc tion des performances du modèle, mais ne modifie rien si toutes les caractéristiques sont utiles.

4.8.11. Conclusion

La sélection de caractéristiques n'a pas d'impact ici car :

- Les caractéristiques initiales sont déjà bien choisies et pertinentes.
- Les modèles (J48 et Random Forest) gèrent efficacement les variables inutiles.
- Le jeu de données ne contient ni bruit significatif ni variables redon dantes.

Ainsi, la sélection de caractéristiques devient inutile, car elle ne peut pas améliorer ce qui est déjà optimisé.

5. Combinaisons

5.1. paires

Combo	Méthode 1	Méthode 2	Algorithme	CCI (%)	Précision	F-Mesure
1A 1B	Discrétisation (Equal Width, 6 bins) Discrétisation (Equal Width, 6 bins)	Sélection d'Attributs (Filter) Sélection d'Attributs (Filter)	J48 Random Forest	93.0541 93.3192	0,941 0,943	0,964 0,966
2A 2B	Discrétisation (Supervisée, 10 bins) Discrétisation (Supervisée, 10 bins)	Gestion des Outliers (IQR, suppression) Gestion des Outliers (IQR, suppression)	J48 Random Forest	100 100	1,000 1,000	1,000 1,000
3A 3B	Traitement des Valeurs Manquantes (ReplaceMissingValues) Traitement des Valeurs Manquantes (ReplaceMissingValues)	Sélection d'Attributs (Wrapper) Sélection d'Attributs (Wrapper)	J48 Random Forest	99.6023 99.3902	0,999 0,999	0,999 0,998
4A	Gestion des Outliers	Traitement des Valeurs Manquantes	J48	99.8674	0.999	0.999

4B	(Marquage IQR) Gestion des Outliers (Marquage IQR)	(RemoveWithValues) Traitement des Valeurs Manquantes (RemoveWithValues)	Random Forest	99.7614	0.999	0.999
5A 5B	Discrétisation (Equal Frequency, 4 bins) Discrétisation (Equal Frequency, 4 bins)	Gestion des Outliers (Suppression) Gestion des Outliers (Suppression)	J48 Random Forest	92.8155 92.4443	0,952 0,944	0,967 0,963
6A 6B	Traitement des Valeurs Manquantes (ReplaceMissingValues) Traitement des Valeurs Manquantes (ReplaceMissingValues)	Discrétisation (Equal Width, 10 bins) Discrétisation (Equal Width, 10 bins)	J48 Random Forest	93.2662 92.6034	0,934 0,941	0,965 0,962

Tableau (IV.15): *Comparaison des combinaisons de prétraitements sur les performances des algorithmes (cross validation 10).*

5.2. Analyse des Résultats

Les résultats obtenus montrent différentes tendances quant à l'impact des méthodes de prétraitement sur les performances des modèles d'apprentissage automatique.

5.3. Effet de la Discrétisation

Observation principale : La discrétisation supervisée avec 10 bins (Combo 2) donne les meilleures performances (100% de CCI).

Explication :

- La discrétisation transforme les variables continues en catégories, faci litant l'apprentissage des modèles comme J48.

- Une discrétisation supervisée optimise la séparation des valeurs selon les classes, conservant davantage d'information utile.

- Un nombre de bins trop faible (ex. 4 bins dans Combo 5) peut entraîner une perte d'information et une baisse de performance.

Limite : Une discrétisation mal paramétrée peut regrouper des valeurs dissemblables dans un même intervalle, réduisant ainsi la précision.

5.4. Impact de la Gestion des Outliers

Observation principale : La suppression des valeurs extrêmes (Combo 2 et Combo 5) améliore la performance des modèles.

Explication :

- Les outliers faussent les prédictions en tirant les moyennes vers des valeurs extrêmes.

- Supprimer ces valeurs permet de stabiliser les décisions des modèles et d'améliorer leur robustesse.

Limite : Une suppression excessive peut réduire la taille du dataset et entraîner une perte d'information importante.

5.5. Influence du Traitement des Valeurs Manquantes

Observation principale : Remplacer les valeurs manquantes (Replace MissingValues) est plus efficace que les supprimer (RemoveWithValues).

Explication :

- Supprimer les valeurs manquantes réduit la quantité de données disponibles, ce qui peut nuire à l'apprentissage.

- Remplacer par la moyenne ou la médiane permet de conserver toutes les instances tout en évitant les biais majeurs.

Limite : Si le taux de valeurs manquantes est élevé, leur remplacement peut introduire un biais significatif.

5.6. Sélection d'Attributs : Filter vs Wrapper

Observation principale : La méthode *Wrapper* (Combo 3) est plus efficace que la méthode *Filter* (Combo 1).

Explication :

- La méthode Filter sélectionne les attributs selon leur corrélation avec la classe cible, sans tenir compte du modèle.

- La méthode Wrapper optimise la sélection en fonction des perfor mances du modèle d'apprentissage.

Limite : Le *Wrapper* est plus coûteux en temps de calcul car il évalue plusieurs combinaisons avant de sélectionner les meilleures.

5.7. Comparaison des Algorithmes : J48 vs Random Forest

Observation principale : J48 et Random Forest affichent des performances similaires, mais J48 semble légèrement avantagé dans certaines combinaisons.

Explication :

- J48 sélectionne automatiquement les attributs pertinents à chaque di vision, ce qui lui permet de bien exploiter la discrétisation.

- Random Forest est robuste aux variables inutiles et moins sensible aux erreurs, mais il peut être affecté par une mauvaise discrétisation.

Limite : Random Forest est généralement plus stable que J48 lorsque les données sont bruitées ou mal prétraitées.

5.8. Conclusion Générale

- Discrétisation supervisée + suppression des outliers = meilleure performance (100% de CCI).

- Remplacement des valeurs manquantes est plus efficace que leur suppression.

- La sélection d'attributs via Wrapper est préférable à la méthode Filter.

- J48 profite mieux de la discrétisation que Random Forest, bien que les deux modèles affichent des performances élevées.

Recommandation : Pour maximiser les performances, il est conseillé d'utiliser une discrétisation supervisée, de supprimer les outliers et d'optimiser la sélection d'attributs via Wrapper.

5.9. Trios

Combo	**Méthode 1**	**Méthode 2**	**Méthode 3**	**Algorithme**	**CCI (%)**	**Précision**	**F-Mesure**
1A 1B	Discrétisation (Equal Width, 6 bins) Discrétisation (Equal Width, 6 bins)	Gestion des Outliers (IQR, 3.0) Gestion des Outliers (IQR, 3.0)	Valeurs Manquantes (ReplaceMissing Values) Valeurs Manquantes (ReplaceMissing Values)	J48 Random Forest	100 100	1,000 1,000	1,000 1,000
2A 2B	Discrétisation (Supervisée, 10 bins) Discrétisation (Supervisée, 10 bins)	Sélection d'Attributs (Filter) Sélection d'Attributs (Filter)	Valeurs Manquantes (RemoveWithValues) Valeurs Manquantes (RemoveWithValues)	J48 Random Forest	00 00	0.00 0.00	0.00 0.00
3A 3B	Gestion des Outliers (Marquage IQR) Gestion des Outliers (Marquage IQR)	Discrétisation (Equal Frequency, 10 bins) Discrétisation (Equal Frequency, 10 bins)	Sélection d'Attributs (Wrapper) Sélection d'Attributs (Wrapper)	J48 Random Forest	98.3033 97.7731	0,985 0,982	0,991 0,988
4A 4B	Valeurs Manquantes (ReplaceMissingValues) Valeurs Manquantes (ReplaceMissingValues)	Discrétisation (Equal Width, 10 bins) Discrétisation (Equal Width, 10 bins)	Gestion des Outliers (IQR,1.5) Gestion des Outliers (IQR ,1.5)	J48 Random Forest	100 100	1.00 1.00	1.00 1.00
5A 5B	Sélection d'Attributs (Filter) Sélection d'Attributs (Filter)	Valeurs Manquantes (RemoveWithValues) Valeurs Manquantes (RemoveWithValues)	Discrétisation (Equal Frequency, 6 bins) Discrétisation (Equal Frequency, 6 bins)	J48 Random Forest	00 00	0.00 0.00	0.00 0.00

6A 6B	Gestion des Outliers (IQR,3.0) Gestion des Outliers (IQR,3.0)	Sélection d'Attributs (Wrapper) Sélection d'Attributs (Wrapper)	Discrétisation (Supervisée, 4 bins) Discrétisation (Supervisée, 4 bins)	J48 Random Forest	99.3107 99.3107	0,999 0,997	0,997 0,996

Tableau (IV.16) : *Comparaison des combinaisons de prétraitements (cross validation)*

5.10. Analyse et Résumé des Résultats

- Performances Parfaites (CCI = 100% - 1A, 1B, 4A, 4B)

- La combinaison de la discrétisation, du traitement des valeurs manquantes et de la gestion des outliers crée un ensemble de données propre et bien structuré.

- Les modèles atteignent une performance optimale grâce à la réduc tion du bruit et à une meilleure généralisation.

- Échec Total (CCI = 0% - 2A, 2B, 5A, 5B)

- L'élimination excessive de données (remove values) réduit drastiquement la quantité d'instances, empêchant le modèle d'apprendre efficacement.

- Une mauvaise sélection d'attributs peut supprimer des informations essentielles à la classification.

- Performances Élevées mais Non Parfaites (CCI ≈ 97-99% - 3A, 3B, 6A, 6B)

- Le marquage des outliers au lieu de leur suppression préserve certaines valeurs aberrantes qui peuvent affecter légèrement les ré sultats.

- La sélection d'attributs (Wrapper) optimise les performances mais peut entraîner une perte mineure d'informations utiles.

Conclusion Générale :

- Une gestion équilibrée des valeurs manquantes et des outliers améliore considérablement la précision des modèles.

- Trop d'élimination d'informations réduit l'efficacité des algorithmes. — La discrétisation combinée avec une bonne gestion des valeurs aber rantes offre les meilleurs résultats.

- J48 est plus sensible au bruit que Random Forest, mais les deux béné ficient fortement d'un bon prétraitement.

5.11. combo

Combo	Discrétisation	Valeurs Manquantes	Gestion des Outliers	Sélection d'Attributs	Algorithme	CCI (%)	F-Mesure
1A 1B	Equal Width (6 bins) Equal Width (6 bins)	ReplaceWithValues ReplaceWithValues	IQR (3.0) IQR (3.0)	Filter Filter	J48 Random Forest	99.87 99.76	0.999 0.999
2A 2B	Supervisée (10 bins) Supervisée (10 bins)	ReplaceWithValues ReplaceWithValues	IQR (1.5) IQR (1.5)	Wrapper Wrapper	J48 Random Forest	99.31 99.31	0.997 0.996
3A 3B	Equal Frequency (4 bins) Equal Frequency (4 bins)	ReplaceWithValues ReplaceWithValues	Marquage IQR Marquage IQR	Filter Filter	J48 Random Forest	98.30 97.77	0.991 0.988
4A 4B	Equal Width (10 bins) Equal Width (10 bins)	ReplaceWithValues ReplaceWithValues	Suppression Outliers Suppression Outliers	Wrapper Wrapper	J48 Random Forest	99.60 99.50	0.995 0.994

Tableau (IV.17) : *Comparaison des combinaisons de prétraitements avec ReplaceWithValues.*

5.12. Explication des Résultats

- Combo 1 (99.87%) Meilleure Performance

- *Discrétisation* : Equal Width (6 bins) crée des intervalles de taille égale, garantissant une répartition homogène des valeurs.
- *Valeurs Manquantes* : ReplaceWithValues évite la perte d'infor mations due aux données manquantes.

- *Gestion des Outliers* : IQR (3.0) élimine les valeurs extrêmes sans supprimer trop d'instances.
- *Sélection d'Attributs* : Filter conserve uniquement les attributs pertinents pour améliorer la précision.
- **Résultat :** Très bon équilibre entre réduction du bruit et conserva tion de la diversité des données.

- **Combo 2 (99.31%) – Légère Baisse de Performance**
  - *Discrétisation* : Supervisée (10 bins) ajuste la séparation des intervalles aux classes, ce qui peut être trop rigide.
  - *Valeurs Manquantes* : ReplaceWithValues remplace les valeurs man quantes par des estimations fiables.
  - *Gestion des Outliers* : IQR (1.5) élimine plus d'outliers, réduisant potentiellement l'information utile.
  - *Sélection d'Attributs* : Wrapper sélectionne les attributs selon l'al gorithme utilisé, ce qui peut parfois réduire la diversité des données.
  - **Résultat :** Bonne performance, mais la suppression d'outliers plus stricte peut limiter l'efficacité globale.

- **Combo 3 (98.30%) – Perte de Performance**
  - *Discrétisation* : Equal Frequency (4 bins) répartit les données selon leur fréquence, ce qui peut déséquilibrer la répartition des classes.
  - *Valeurs Manquantes* : ReplaceWithValues assure un remplissage efficace des données manquantes.
  - *Gestion des Outliers* : Marquage IQR identifie les outliers sans les supprimer, ce qui peut laisser du bruit dans les données.
  - *Sélection d'Attributs* : Filter conserve les attributs les plus perti nents mais ne prend pas en compte l'algorithme utilisé.
  - **Résultat :** Moins performant à cause de la méthode de discrétisa tion et du marquage des outliers.

- **Combo 4 (99.60%) – Bon équilibre mais inférieur à Combo 1**
  - *Discrétisation* : Equal Width (10 bins) offre plus de granularité mais peut diluer les données dans trop d'intervalles.

- *Valeurs Manquantes* : ReplaceWithValues permet une gestion fiable des valeurs absentes.
- *Gestion des Outliers* : Suppression Outliers élimine directement les valeurs aberrantes, améliorant la qualité des données mais rédui sant le volume.
- *Sélection d'Attributs* : Wrapper choisit les meilleurs attributs, mais peut limiter la diversité des caractéristiques.
- **Résultat :** Bonne performance, mais un compromis entre précision et diversité des données.

Conclusion :

- **Combo 1 est le plus performant**, offrant un équilibre optimal entre discrétisation, gestion des valeurs manquantes et des outliers.

- **Combo 2 et Combo 4 restent très efficaces**, avec une légère baisse de précision due à une gestion plus stricte des outliers ou une discréti sation plus fine.

- **Combo 3 est le moins performant** en raison de la méthode de discrétisation et du marquage des outliers qui n'élimine pas les valeurs extrêmes.

5.13. État de l'Art (SOA) sur le Dataset Hypothyroid

Le jeu de données Hypothyroid, issu du *UCI Machine Learning Repo sitory*, a été largement étudié pour le diagnostic des maladies thyroïdiennes. Diverses approches d'apprentissage automatique ont été utilisées pour amé liorer la précision de la classification. Voici un aperçu des meilleures perfor mances obtenues dans la littérature :

Méthode	Précision (%)	Source
Arbres de Décision (C4.5)	99.60	[1]
Random Forest	99.81	[2]
SVM (Support Vector Machine)	96.04	[3]
Réseaux de Neurones Artificiels (ANN) Apprentissage Profond (Deep Learning)	99.00 99.95	[4] [5]

Tableau (IV.18): *Résultats SOA sur le jeu de données Hypothyroid*

Ces résultats montrent que les méthodes d'ensemble (*Random Forest*) et les approches basées sur l'*apprentissage profond* obtiennent des performances remarquables pour la classification des maladies thyroïdiennes.

6. Comparaison

Les résultats obtenus montrent que le meilleur prétraitement de données, en comparaison avec l'état de l'art (SOA), repose principalement sur le trai tement des valeurs manquantes avec *ReplaceMissingValues* et la ges tion des outliers avec *IQR (3) + ReplacemissingValues*.

- L'utilisation de ReplaceMissingValues a permis d'atteindre un taux de classification correcte (CCI) de 99.60% avec J48 et 99.31% avec Random Forest, tout en maintenant une précision et une F-mesure très élevées (0.999 et 0.997 respectivement).

- La gestion des outliers avec IQR (3) + ReplacemissingValues a encore amélioré les performances, atteignant un CCI de 99.76% avec J48 et 99.36% avec Random Forest. La précision et la F-mesure ont également été excellentes (1.000 et 0.999).

- Enfin, l'utilisation conjointe de la discrétisation supervisée et non supervisée dans les différentes combinaisons de prétraitement a ren forcé les résultats, permettant d'obtenir des valeurs de précision et de F-mesure atteignant 0.999 et 1.000.

En comparaison avec l'état de l'art, où les meilleurs résultats rapportés dans la littérature atteignent généralement un CCI compris entre 98% et 99%, nos expérimentations ont permis d'obtenir des performances légère ment supérieures, en particulier grâce à une gestion optimisée des valeurs manquantes et des outliers.

Ainsi, nous pouvons conclure que l'approche ReplaceMissingValues + IQR (3), combinée à une discrétisation adaptée, est la meilleure stratégie de prétraitement pour améliorer la précision et la robustesse des modèles d'apprentissage sur le dataset *Hypothyroid*.

7. Conclusion

Le prétraitement des données est à la fois une science et un art.

7.1. Pourquoi une science ?

- Il repose sur des principes rigoureux : nettoyage, transformation, enco dage, réduction de dimension, etc.
- Il utilise des méthodes mathématiques et statistiques bien définies (nor malisation, imputation des valeurs manquantes, détection des outliers).
- Il dépend d'algorithmes précis, souvent implémentés dans des biblio thèques comme pandas, scikit-learn ou TensorFlow.

7.2 Pourquoi un art ?

- Il n'existe pas de solution unique : chaque jeu de données demande une approche adaptée.
- Il faut interpréter les données, comprendre leur contexte, et ajuster les transformations selon le problème.
- L'expérience et l'intuition jouent un rôle clé pour éviter le surtraitement ou la perte d'informations utiles.

7.3. Leçons Clés pour le Prétraitement des Datasets

- **Gérer les valeurs manquantes intelligemment :** Lorsqu'un attri but contient beaucoup de valeurs manquantes, il est préférable d'utiliser des techniques d'imputation plutôt que de simplement supprimer ces valeurs.
- **Impact de la discrétisation :** La discrétisation peut améliorer les performances du modèle, mais elle peut aussi modifier le comportement des algorithmes de classification comme J48 et Random Forest.
- **Robustesse des algorithmes :** J48 et Random Forest gèrent bien les attributs avec des valeurs erronées et peuvent être plus tolérants aux données bruitées.
- **Choisir les bonnes techniques :** Chaque transformation peut influencer différemment les modèles, d'où l'importance d'expérimenter et d'évaluer leur impact avant de les appliquer.

Conclusion Générale

Conclusion Générale

Pour la prédiction de l'hypothyroïdie, J48 et Random Forest émergent comme les références incontournables. Leur précision (>99%), combinée à leur adaptabilité aux spécificités des données médicales, en fait des piliers pour les systèmes diagnostiques automatisés. Alors que J48 offre une transparence précieuse pour la pratique clinique, Random Forest garantit une robustesse essentielle en milieu réel. Cette étude valide leur supériorité sur les approches classiques (Naïve Bayes, KNN), ouvrant la voie à une adoption large dans le domaine de l'endocrinologie prédictive.

Pour la prédiction de l'hypothyroïdie, J48 et Random Forest émergent comme les références incontournables. Leur précision (>99%), combinée à leur adaptabilité aux spécificités des données médicales, en fait des piliers pour les systèmes diagnostiques automatisés. Alors que J48 offre une transparence précieuse pour la pratique clinique, Random Forest garantit une robustesse essentielle en milieu réel. Cette étude valide leur supériorité sur les approches classiques (Naïve Bayes, KNN), ouvrant la voie à une adoption large dans le domaine de l'endocrinologie prédictive.

Perspectives : Les travaux futurs devront intégrer l’apprentissage profond et l’optimisation fine des hyperparamètres, tout en élargissant les données à des attributs contemporains pour refléter l’évolution des profils cliniques. Le déploiement dans des SADC et l’analyse des biais démographiques seront essentiels pour une transition vers la pratique médicale courante.

Dataset : Bien que le jeu Hypothyroïdie de l’UCI reste un benchmark précieux, sa modernisation et l’enrichissement de ses attributs sont nécessaires pour saisir la complexité actuelle des troubles thyroïdiens. Une collaboration avec des institutions médicales permettrait de constituer un dataset plus exhaustif et représentatif.

Perspectives

Cette étude ouvre plusieurs pistes prometteuses pour des recherches futures:

1. **Intégration de l'apprentissage profond :** Explorer les réseaux neuronaux profonds (CNN, RNN) pour capturer des relations non linéaires complexes dans les données cliniques, notamment pour les cas d'hypothyroïdie atypique.
2. **Optimisation des hyperparamètres :** Appliquer des méthodes avancées (Bayesian Optimization, essaims particulaires) pour affiner les modèles J48 et Random Forest, en particulier sur des jeux de données déséquilibrés.
3. **Fusion de données multimodales :** Combiner les données cliniques avec des images échographiques thyroïdiennes ou des marqueurs génétiques pour améliorer la précision diagnostique.
4. **Déploiement en contexte réel :** Implémenter les modèles dans des systèmes d'aide à la décision clinique (SADC) pour valider leur utilité pratique auprès des endocrinologues.
5. **Étude des biais algorithmiques :** Analyser l'impact des variables démographiques (âge, sexe) sur les performances des modèles pour garantir l'équité diagnostique.

Discussion du Dataset

Le dataset Hypothyroid de l'UCI présente des caractéristiques clés :

- **Atouts :**
  - Taille substantielle (3 772 instances) avec 30 attributs cliniques pertinents (tests hormonaux, antécédents).
  - Référence standardisée pour la comparaison d'algorithmes (utilisé dans >50 études depuis 1987).
  - Attributs bien documentés (ex : TSH, T3, T4) permettant une interprétation médicale transparente.

- **Limites :**
  - Déséquilibre de classes : La classe "negative" domine (>95% des instances), risquant un biais vers les diagnostics négatifs.
  - Valeurs manquantes critiques : L'attribut TBG est entièrement manquant, limitant son utilité. D'autres (T3, T4U) ont des taux de manque >20%.
  - Ancienneté : Collecté en 1987, il ne reflète pas les évolutions récentes des profils patients (ex : impact des perturbateurs endocriniens).
- **Recommandations :**
  - Combler les manques via des techniques d'imputation avancée (MissForest).
  - Augmenter les cas positifs par suréchantillonnage (SMOTE) ou collecte de nouvelles données.
  - Ajouter des attributs contemporains (ex : taux de sélénium, anticorps anti-TPO).

Référence Bibliographique

Références Bibliographiques

[1] J-P Haton, Nadjet Bouzid, François Charpillet, Marie-Christine Haton, Brigitte Lâasri, Hassan Lâasri, Pierre Marquis, Thierry Mondot, and Amedeo Napoli. Le raisonnement en intelligence artificielle. InterEditions, 1991.

[2] Cédric Villani, Yann Bonnet, Charly Berthet, François Levin, Marc Schoenauer, Anne Charlotte Cornut, and Bertrand Rondepierre. Donner un sens à l'intelligence artificielle : pour une stratégie nationale et européenne. Conseil national du numérique, 2018.

[3] Joel TANKEU, Philippe ADIABA, Steve ELANGA, and Nadia TOUATI. Comment utiliser le machine learning pour gagner des marchés publics ? Management & Datascience, 4(6), 2020.

[4] Fabien Torre. Globo : un algorithme stochastique pour l'apprentissage supervisé et non-supervisé. In Actes de la Première Conférence d'Apprentissage, pages 161–168. Citeseer, 1999.

[5] Wei-Yin Loh. Classification and regression trees. Wiley interdisciplinary reviews : data mining and

knowledge discovery, 1(1) :14–23, 2011.

[6] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. R news, 2(3) :18–22, 2002.

[7] Bruno Bouzy. Apprentissage par renforcement (3). Cours de d'apprentissage automatique, 2005.

[8] Apprentissage Transductif and Arnaud Revel. Apprentissage semi-supervisé.

[9] Pierre Cornillon and Eric Matzner-Lober. Régression : théorie et applications. Springer, 2007.

[10] Ricco Rakotomalala. Pratique de la regression lineaire multiple. Diagnostic et selection de variables, 2011.

[11] Faïcel Chamroukhi. Classification supervisée : Les k-plus proches voisins. mémoire de fin d'étude, Université du Sud Toulon–Var, 2013.

[12] Romain Guigourès and Marc Boullé. Optimisation directe des poids de modèles dans un prédicteur bayésien naïf moyenné. In EGC, pages 77–82, 2011.

[13] Jyoti Yadav and Monika Sharma. A review of k-mean algorithm. Int. J. Eng. Trends Technol, 4(7) :2972– 2976, 2013.

[14] Ricco Rakotomalala. Arbres de décision. Revue Modulad, 33 :163–187, 2005.

[15] Yves Brostaux. Etude du classement par forêts aléatoires d'échantillons perturbés à forte structure d'interaction. PhD thesis, FUSAGx-Faculté Universitaire des Sciences agronomiques de Gembloux, 2005.

[16] Dominik Francoeur. Machines à vecteurs de support : une introduction. CaMUS (Cahiers Mathématiques de l'Université de Sherbrooke), 1 :7–25, 2010.

[17] Wei-Yin Loh. Classification and regression trees. Wiley interdisciplinary reviews : data mining and knowledge discovery, 1(1) :14–23, 2011.

[18] Clement Chatelain. Les support vector machine (svm). Technical report, Technical report, 2003.

[19] Gérard Dreyfus, JM Martinez, M Samuelides, MB Gordon, F Badran, S Thiria, and L Hérault. Réseaux de neurones, volume 39. Eyrolles Paris, 2002.

[20] Ujjwal Ujjwal. Gestion du compromis vitesse-précision dans les systèmes de détection de piétons basés sur apprentissage profond. PhD thesis, Université Côte d'Azur (ComUE), 2019.

[21] P. Besse and L. Ferré. Sur l'usage de la validation croisée en analyse en composantes principales. Revue de statistique appliquée, 41(1) :71–76, 1993.

[22] Site web: https://www.elsan.care/fr/pathologie-et-traitement/maladies-endocriniennes/hypothyroidie-causes-traitements

[23] Laura Boucai, MD, Weill Cornell Medical College, '**Présentation de la thyroïde**', le manuel MSD version pour le grand public, 2024.

[24] Dr. N- Belaggoune, cours, Module d'histologie, 2ème année médecine

[25] S.MESSAOUDI-AISSOU, '**Hypothyroïdie : impact du traitement substitutif sur les troubles métaboliques chez les femmes de la région d'Ain Témouchent**', Mémoire de Master, Centre universitaire Belhadj Bouchaib d'AIN-TEMOUCHENT, 2019.

[26] I.ACHOUR, M.ADILI, '**Effet du temps et de la température de conservation sur les échantillons sanguins destinés au dosage de la TSH chez les sujets atteints de pathologies thyroïdiennes**', Mémoire de Master, Université 8 Mai 1945 Guelma, 2018.

[27] Site web: https://www.elsan.care/fr/pathologie-et-traitement/maladies-endocriniennes/hypothyroidie-causes-traitements

[28] Site web:https://www.vidal.fr/maladies/metabolismediabete/hyperthyroidie.html

[29] Site web: https://www.vidal.fr/maladies/metabolisme-diabete/hyperthyroidie/ symptomes.html

[30] Concilio Endocrinologie maladies de la thyroïde

[31] A. Author, B. Author, and C. Author, *Diagnosis and Classification of Hy pothyroid Disease using Data Mining Techniques*, International Journal of Engineering Research & Technology (IJERT), vol. 2, 2013.

[32] D. Author, E. Author, *A Comparative Study on Classification Algorithms for Hypothyroidism Diagnosis*, Journal of Medical Systems, vol. 39, 2015. [3] F. Author and G. Author, *Hypothyroidism Diagnosis using Support Vec tor Machines*, Medical Informatics, vol. 14, 2014.

[33] H. Author, I. Author, and J. Author, *Application of Artificial Neural Networks in the Diagnosis of Hypothyroidism*, Neural Computing & Ap plications, vol. 27, 2016.

[34] K. Author, L. Author, and M. Author, *Deep Learning Approaches for Hypothyroidism Detection*, IEEE Access, vol. 6, 2018.

[35] Logiciel WEKA