الجمهورية الجزائرية الديمقراطية الشعبية

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

École normale supérieure d'enseignement technologique. SKIKDA

Département de mathématiques et d'informatique

Mémoire

Présenté pour obtenir un diplôme : Professeur d'Enseignement Moyen

Application du Machine Learning pour le Diagnostic Automatique des maladies

Domaine d'études : Informatique

Présenté par

- Assem ALIA
- Laid CHEKAMBOU
- Ahmed CHEKAMBOU

Jury

Président Dr. Riadh BOUAITA

Superviseur Dr. Halima SALAH

Examinateurs Dr. Rida Mezghache

Dr. Samir Sellami

Année académique : 2024/2025

Remerciements

Loué soit Allah, qui nous a accordé le succès, la stabilité et qui nous a aidés à terminer ce travail après avoir œuvré pour mettre les points sur les i et arriver à la concrétisation de cette Mémoire.

Nous tenons à adresser nos sincères remerciements à notre directrice de projet, Dr. Halima Salah, pour son soutien constant, ses conseils précieux et sa vision scientifique qui nous ont grandement aidés à comprendre et à atteindre les objectifs de ce projet. Elle a joué un rôle déterminant dans l'accomplissement de ce travail.

Nous tenons également à remercier tous les médecins qui ont contribué par leurs consultations et informations précieuses, en particulier **Dr. Abd Ennour Daikha**, **Dr. Tarek Alia**, **Dr. Amine Lachlah** et **Dr. Ali Amiar**, qui n'ont ménagé aucun effort pour soutenir ce projet et nous apporter leurs observations et conseils médicaux, qui ont été essentiels dans l'avancement de ce travail. Leur générosité intellectuelle et leur attention sont inestimables.

Nous souhaitons exprimer notre profonde gratitude envers toutes les personnes qui nous ont soutenus, que ce soit par des mots bienveillants ou par une aide directe. Ce travail n'aurait pas été possible sans l'aide continue des personnes sincères.

Dédicace

Nous tenons à exprimer toute notre gratitude et nos sincères remerciements à nos parents, nos frères et nos sœurs pour leur soutien inébranlable et leur amour constant. Leur présence, leurs encouragements et leurs sacrifices nous ont permis de poursuivre nos études avec détermination et de mener à bien ce projet. Cette Mémoire est le fruit de leur soutien sans faille, et c’est grâce à eux que nous avons pu réaliser ce rêve. Que ce travail soit une petite reconnaissance de tout ce qu'ils nous ont donné.

Table des matières

Table des matières

Tableau des abréviations et acronymes

Abréviation / Acronyme	Définition
API	Application Programming Interface
API REST	Representational State Transfer API
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
DQN	Deep Q-Network
ECG	Électrocardiogramme
F1-Score	Mesure de performance combinée précision-rappel
FN	False Negative (Faux Négatif)
FP	False Positive (Faux Positif)
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
IA	Intelligence Artificielle
IMC	Indice de Masse Corporelle
IRM	Imagerie par Résonance Magnétique
IRMf	Imagerie par Résonance Magnétique fonctionnelle
KNN	k-Nearest Neighbors
L1 / L2	Régularisation Lasso / Ridge
MDP	Markov Decision Process
ML	Machine Learning (Apprentissage Automatique)
NLP	Natural Language Processing
PCA	Principal Component Analysis
RF	Random Forest
RNN	Recurrent Neural Network
SMOTE	Synthetic Minority Over-sampling Technique
SQL	Structured Query Language
SVM	Support Vector Machine
TN	True Negative (Vrai Négatif)
TP	True Positive (Vrai Positif)
UCI	University of California, Irvine (Machine Learning Repository)
XGBoost	Extreme Gradient Boosting

Liste des figures

Liste des tableaux

Introduction générale

Contexte du travail

L'évolution rapide des technologies d'intelligence artificielle (IA) a permis d'envisager des solutions innovantes dans de nombreux secteurs, notamment dans le domaine médical. Ce dernier exige un haut niveau de précision et de réactivité dans la prise de décision, notamment pour le diagnostic des maladies. L'apprentissage automatique (machine learning), sous-domaine de l'IA, s'impose aujourd'hui comme un levier puissant pour assister les professionnels de santé et améliorer la qualité des soins. En exploitant de grandes quantités de données médicales, il devient possible de construire des systèmes intelligents capables d'analyser, de prédire et de recommander des décisions adaptées à chaque situation clinique.

Problématique

Le diagnostic médical reste une tâche complexe, souvent confrontée à des difficultés telles que la diversité des symptômes, la similarité entre certaines maladies, ou encore le manque de données structurées. Ces contraintes peuvent conduire à des erreurs de diagnostic, à une perte de temps ou à une surcharge du personnel médical. Par ailleurs, les données médicales disponibles sont souvent déséquilibrées ou incomplètes, rendant difficile l'entraînement de modèles fiables. Il est donc nécessaire de mettre en place une solution capable de centraliser des données de qualité, de les analyser intelligemment, et de fournir des résultats pertinents, que ce soit pour les professionnels ou pour les patients.

Contribution

Dans ce contexte, notre projet de fin d'études propose la conception et le développement d'un **système intelligent de diagnostic médical basé sur l'apprentissage automatique**. Ce projet repose sur plusieurs contributions majeures :

- **Création d'une base de données médicale personnalisée**, issue de différentes sources, nettoyée, traitée et préparée pour servir de base d'apprentissage fiable et représentative ;
- **Comparaison de trois algorithmes de classification** (XGBoost, Random Forest et SVM) appliqués à des bases standards (Breast Cancer, Heart Disease), ainsi notre base développée dans le but d'identifier le modèle le plus performant pour notre système.

- des techniques comme **SMOTE** et **l'ajustement des poids** ont été utilisées afin de former des modèles plus équilibrés et efficaces.
- La **méthode sélectionnée**, combinée à notre **base de données développée**, a été intégrée dans deux types d'interfaces :
    - ➔ une **application de bureau** destinée aux professionnels de santé, pour l'aide à la décision médicale,
    - ➔ une **application mobile** à usage des patients, pour proposer un diagnostic initial et des conseils utiles.
- **Développement d'un système de recommandation**, proposant un diagnostic, des médicaments et des conseils médicaux adaptés, à partir des données saisies.

Organisation

Ce projet est structuré en trois chapitres :

- Le **premier chapitre** présente les généralités sur l'apprentissage automatique et la classification, ainsi que leurs principales applications dans le domaine médical.
- Le **deuxième chapitre** est consacré à la présentation des méthodes de classification utilisées, des bases de données exploitées (dont celles issues du *UCI Repository*) ainsi que le développement de notre propre base de données médicale.
- Enfin, le **troisième chapitre** porte sur l'implémentation des différents algorithmes, les expérimentations réalisées, l'évaluation des performances, ainsi que la présentation de notre application de diagnostic.

Chapitre I. Introduction à l'apprentissage automatique

I.1) Introduction à l'intelligence artificielle (IA)

L'intelligence artificielle (IA) est aujourd'hui très utilisée dans plusieurs domaines, notamment dans le domaine médical. Parmi les techniques les plus importantes de l'IA, on trouve **l'apprentissage automatique (ou machine learning)**. Cette méthode permet à un ordinateur d'apprendre à partir des données, sans être programmé directement pour chaque tâche.

Une des techniques de base en apprentissage automatique est la **classification**. Elle consiste à **classer des données dans des groupes** ou des catégories. Par exemple, on peut utiliser la classification pour détecter si un patient est malade ou non à partir de ses résultats médicaux.

Dans le domaine de la santé, ces techniques sont de plus en plus utilisées. Elles permettent par exemple d'**aider les médecins à poser un diagnostic**, de **prédire l'évolution d'une maladie**, ou encore de **détecter automatiquement des anomalies sur une radiographie**. [1] ,[2] ,[3] .

Dans ce chapitre, nous allons présenter :

- les bases de l'apprentissage automatique,
- les différents types d'apprentissage (supervisé, non supervisé),
- le principe de la classification,
- ainsi que quelques exemples d'applications dans le domaine médical.
- les études antérieures

I.2) Apprentissage automatique (Machine Learning)

I.2.1) Définition de l'apprentissage automatique

L'apprentissage automatique, ou Machine Learning, est une branche de l'intelligence artificielle (IA) qui permet aux systèmes informatiques d'apprendre et de s'améliorer automatiquement à partir de données, sans être explicitement programmés. Selon Arthur Samuel (1959), le Machine Learning est « le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés ».

De manière plus technique, Tom Mitchell (1997) définit l'apprentissage automatique comme suit :

« Un programme informatique est dit apprendre de l'expérience E par rapport à une tâche T et une mesure de performance P, si sa performance sur T, mesurée par P, s'améliore avec l'expérience E ».

Concrètement, l'apprentissage automatique repose sur l'utilisation de données appelées ensemble d'apprentissage pour construire des modèles capables de généraliser à de nouvelles données. Contrairement aux programmes traditionnels qui appliquent des procédures prédéfinies aux données pour produire des résultats, les algorithmes de Machine Learning analysent les données et les résultats pour générer automatiquement ces procédures.

Par exemple, dans le cadre d'un filtre anti-spam, le programme utilise un ensemble de courriels étiquetés comme spam et non-spam pour apprendre à détecter automatiquement les courriels indésirables dans le futur. Ici, la tâche T consiste à classer les courriels, l'expérience E est constituée des données d'apprentissage (courriels étiquetés), et la mesure de performance P peut être le taux de précision des classifications.

En résumé, l'apprentissage automatique constitue une discipline clé pour la résolution de problèmes complexes où il est difficile, voire impossible, de programmer explicitement toutes les règles nécessaires. Cette capacité d'apprentissage ouvre la voie à des applications variées, allant de la reconnaissance vocale à la recommandation de produits, en passant par la prédiction de comportements utilisateurs [4], [5], [6].

I.2.2) Étapes de développement de modèles d'apprentissage automatique

Le développement d'un modèle d'apprentissage automatique suit une série d'étapes systématiques afin de s'assurer de l'efficacité, de la précision et de la pertinence de celui-ci. Ces étapes peuvent être résumées comme suit :

I.2.2.1) Définition du problème et collecte des données

La première étape consiste à bien comprendre le problème que l'on veut résoudre et à bien identifier les objectifs (Tâche T, mesure de performance P) et à collecter les données nécessaires (expérience E). Un exemple de tâche de classification d'e-mails

consisterait à collecter des données en grande quantité par plusieurs exemples d'e-mails étiquetés comme **spam** ou **ham**.

I.2.2.2) Préparation et exploration des données

Cette étape consiste en un ensemble d'opérations de :

- Nettoyage des données : traitement des données manquantes, disparues ou aberrantes.
- Normalisation et mise en forme des données

Prise d'informations concernant les données grâce à une analyse exploratoire (données, visualisations, corrélations, etc.).

I.2.2.3) Partage des données

Les données sont globalement divisées en trois ensembles :

- **Ensemble d'entraînement (Training Set)** : pour entraîner le modèle.
- **Ensemble de validation (Validation Set)** : pour ajuster les hyperparamètres.
- **Ensemble de test (Test Set)** : pour évaluer la performance finale du modèle

I.2.2.4) Sélection du modèle

Le choix du modèle sera déterminé par la nature de la problématique (classification, régression, clustering, …) et en fonction des données à disposition dans chaque cas. Parmi les modèles les plus répandus dans le domaine, on peut par exemple citer les régressions linéaires, les forêts aléatoires, les réseaux de neurones, etc.

I.2.2.5) Entraînement du modèle

A ce stade, les paramètres du modèle sont optimisés via l'ensemble d'entraînement. Les algorithmes d'optimisation, tels que le gradient de descente, sont des acteurs majeurs dans l'amélioration de la performance du modèle.

I.2.2.6) Évaluation et validation

Le modèle est évalué via l'ensemble de validation sur sa capacité à généraliser sur de nouvelles données non vues. Les indicateurs de performance utilisés sont typiquement les suivants :

Précision (Accuracy) : Il est utilisé comme indicateur approximatif de la progression de la formation du modèle ou de la convergence sur des ensembles de données équilibrés. Cependant, il doit toujours être utilisé en conjonction avec d'autres mesures et évité sur des ensembles de données déséquilibrés en raison de son potentiel trompeur, telles que la Précision (Precision), le Rappel (Recall) et la F-mesure (F1-score), qui offrent une évaluation plus complète des performances du modèle, notamment en cas de déséquilibre des classes.

- Une fraction des échantillons correctement classés.

Accuracy=(TP+TN)/(TP+FP+TN+FN).

La matrice de confusion.

Précision moyenne (Precision): Parmi les cas classés comme positifs, quel pourcentage est réellement vrai ?

Precision= TP/(TP+FP)

Rappel moyen (Recall): Parmi les cas véritablement positifs, combien ont été correctement détectés ?

Recall=TP/(TP+FN)

F1 Score : Propose une mesure équilibrée entre précision et rappel, en particulier dans les contextes d'ensembles de données déséquilibrés étant donné que le compromis entre précision et rappel est au cœur de l'alternative à une précision seulement, tant redoutée.

F1-score=2×(Precision×Recall) / (Precision+Recall)

De sorte que :

- TP = True Positive (Le modèle prédit que le patient est malade → et c'est vrai)
- TN = True Negative (Le modèle prédit que le patient est non malade → et c'est vrai)
- FP = False Positive (Le modèle prédit que le patient est malade → mais il ne l'est pas)
- FN = False Negative (Le modèle prédit que le patient est non malade → mais il est malade)

I.2.2.7) Ajustement des hyperparamètres

Les hyperparamètres tels que le taux d'apprentissage ou la profondeur maximum d'un arbre de décision sont réajustés pour parfaire la performance globale.

I.2.2.8) Test final et déploiement

Le modèle est testé sur l'ensemble de test pour vérifier sa capacité à être éventuellement mis en pratique dans le monde réel. Une fois qu'il est validé, son déploiement pourra se faire au sein d'un environnement de production.

I.2.2.9) Surveillance et maintenance

Le processus ne s'arrête pas au déploiement, car les modèles doivent être monitorés en permanence pour détecter d'éventuelles dérives [5], [6], [7].

I.3) Types d'apprentissage automatique

I.3.1) Apprentissage supervisé

I.3.1.1) Définition de l'apprentissage supervisé

L'apprentissage supervisé constitue une forme d'apprentissage automatique dans laquelle on entraîne un modèle sur un jeu de données étiqueté. Les données d'entrée (ou exemples d'entraînement) sont accompagnées d'une sortie (ou label cible), et le modèle apprend à relier les entrées aux sorties afin de pouvoir fournir des réponses correctes lorsqu'il est confronté à de nouvelles données non étiquetées [5].

I.3.1.2) Principe de fonctionnement

Dans l'apprentissage supervisé, un modèle est fourni avec un ensemble de données d'entraînement sous la forme (xi,yi) où xi représente l'entrée et yi est le label ou la sortie associée. Pendant l'entraînement, le modèle tente de minimiser une fonction de perte, qui mesure l'écart entre ses prédictions et les valeurs réelles (les labels). Une fois ce processus terminé, le modèle peut être utilisé pour effectuer des prédictions sur des données inconnues [5].

I.3.1.3) Types de tâches

L'apprentissage supervisé est couramment utilisé pour résoudre deux types principaux de problèmes :

- **Classification :** Dans ce cas, le modèle attribue une catégorie ou une classe à chaque observation. Par exemple, il peut être utilisé pour distinguer les e-mails légitimes des spams.
- **Régression :** Le modèle prédit une valeur continue. Par exemple, il peut être utilisé pour prédire le prix d'un bien immobilier en fonction de ses caractéristiques (taille, emplacement, etc.) [4].

I.3.1.4) Exemples courants d'application :

L'apprentissage supervisé trouve de nombreuses applications dans divers domaines, parmi lesquels :

- **Vision par ordinateur :** Classification d'images (par exemple, reconnaître des visages ou des objets dans des photos).
- **Traitement du langage naturel (NLP) :** Analyse des sentiments, traduction automatique de textes.
- **Finance :** Prévision des tendances boursières, détection de fraudes [7].

I.3.1.5) Caractéristiques et avantages :

L'apprentissage supervisé, en effet, présente plusieurs avantages :

- **Efficacité :** il est particulièrement efficace lorsque l'on dispose d'un grand nombre de données étiquetées.

- **Précision :** permet d'obtenir des prédictions fiables dans des environnements bien définis, où les données d'entraînement sont représentatives des cas futurs auxquels le modèle sera confronté.
  Par exemple : dans un système de détection de spams, une haute précision signifie que la majorité des e-mails identifiés comme spam sont réellement des spams. Cela est crucial lorsqu'il faut éviter de classer des e-mails importants comme indésirables.
- **Large éventail d'applications :** on l'utilise dans quasiment tous les secteurs : la médecine, le commerce, l'industrie, l'art, etc.

Néanmoins, il est souvent limité en raison de la disponibilité des données étiquetées, de même que de ses risques de sur-apprentissage (overfitting) lorsque le modèle est trop spécialisé par rapport aux données d'apprentissage pour être efficace dans la prédiction des nouvelles données [4], [5], [7], [8], [9], [10].

I.3.2) Apprentissage non supervisé

I.3.2.1) Définition de l'apprentissage non supervisé

L'apprentissage non supervisé est une branche de l'apprentissage automatique où un modèle est entraîné sur des données non étiquetées. Contrairement à l'apprentissage supervisé, il n'y a pas de correspondance explicite entre les entrées et les sorties. Le modèle doit donc identifier des structures ou des motifs sous-jacents dans les données sans aucune indication préalable [5].

I.3.2.2) Principe de fonctionnement :

Dans le cadre de l'apprentissage non supervisé, le modèle traite un ensemble de données Xi, où chaque observation est considérée comme entrée sans label ou sortie. En somme, le modèle cherche à regrouper les observations similaires, ou à réduire la dimensionnalité des données pour extraire des informations pertinentes. Les objectifs récurrents sont, par exemple : identifier des groupes homogènes (clustering) ; réduire la complexité des données efficacement tout en conservant la variance (réduction de la dimensionnalité) ; découvrir des relations cachées ou des anomalies [5].

I.3.2.3) Types de tâches :

L'apprentissage non supervisé est fréquemment mis en œuvre pour :

Clustering ou segment induction : consiste à regrouper ensemble, les données en segments ou clusters homogènes. Exemple : segmentation de la clientèle en marketing ;

Réduction de dimensionnalité : Il s'agit de simplifier des données complexes pour en faciliter l'analyse et la visualisation. Exemple : analyse en composantes principales (PCA) sur données biologiques.;

Anomalies ou outlier detection : il s'agit de repérer des observations inhabituelles, rares ou inattendues.

Exemple : Détection de fraudes dans les transactions financières [11].

I.3.2.4) Exemples courants d'application

L'apprentissage non supervisé trouve des applications variées dans plusieurs domaines :

Vision par ordinateur : Segmentation d'images médicales, détection d'objets.

Traitement du langage naturel (NLP) : Analyse de sujets ou regroupement de documents similaires.

Santé : Identification de biomarqueurs dans les données génomiques, détection de clusters dans des ensembles de patients [7].

I.3.2.5) Caractéristiques et avantages

L'apprentissage non supervisé présente plusieurs atouts :

Flexibilité : Utilisation possible même sans labels ou annotations.

Découverte de structures cachées : Utile pour explorer des données complexes ou peu comprises.

Réduction des coûts : Moins coûteux car il ne nécessite pas d'étiquetage manuel des données.

Cependant, il présente aussi certaines limitations :

Interprétation des résultats : Les résultats obtenus peuvent être plus difficiles à analyser ou à valider.

Moins précis : L'absence de supervision peut limiter la fiabilité des prédictions [5], [7], [11].

I.3.3) Apprentissage semi-supervisé

I.3.3.1) Définition de l'apprentissage semi-supervisé

L'apprentissage semi-supervisé se définit comme une approche en apprentissage automatique qui combine des données étiquetées et non étiquetées pour entraîner le modèle visé. Cette approche est appropriée à des situations où l'étiquetage des données est long ou coûteux et à des situations où de grandes quantités de données non étiquetées existent. Le but fondamental de l'apprentissage semi-supervisé est d'exploiter le contenu informatif des données non étiquetées pour obtenir de meilleures performances qu'en apprentissage supervisé pur [5].

I.3.3.2) Principe de fonctionnement

Dans un cadre semi-supervisé, le modèle apprend à partir d'un petit échantillon étiqueté pour obtenir une structure initiale en vue d'améliorer les prédictions à partir des non-étiquetées en supposant que les données étiquetées et non étiquetées s'ordonnent selon la même structure (distribuée).

Exemple : En reconnaissance d'image, un petit ensemble d'images étiquetées (e.g., "chat", "chien") peut être complété par de nombreuses images non étiquetées pour enrichir les représentations du modèle.

Avantages de l'apprentissage semi-supervisé :

- **Réduction des coûts** : Réduit la dépendance à un grand volume de données étiquetées.
- **Efficacité** : Exploite la richesse des données non étiquetées pour améliorer les résultats.
- **Flexibilité** : Utile dans des domaines où l'étiquetage est difficile, comme la médecine ou la biologie, car l'acquisition de labels fiables peut nécessiter l'expertise humaine, être coûteuse ou prendre beaucoup de temps. L'apprentissage non supervisé permet alors d'exploiter les données sans avoir besoin de cette étape [5].

I.3.3.3) Limites de l'apprentissage semi-supervisé :

- Peut être sensible à la qualité des données non étiquetées.
- Nécessite des algorithmes robustes pour bien exploiter la structure sous-jacente des données [5].

I.3.4) Apprentissage par renforcement

I.3.4.1) Définition de l'apprentissage par renforcement

L'apprentissage par renforcement désigne la partie de l'apprentissage automatique où un agent apprend à choisir les meilleures actions possibles en interagissant avec un environnement, i.e. là où il ne sait pas a priori quelle serait la bonne action à prendre (pas de données étiquetées disponibles) et où il doit

se déterminer grâce aux signaux de récompense ou de punition qu'il reçoit à partir de ses actions [12].

I.3.4.2) Principe de fonctionnement

L'apprentissage par renforcement repose sur un cadre formel **appelé processus de décision markovien (MDP)**, où :

1. **Agent :** Le système ou programme qui apprend.
2. **Environnement :** Le monde avec lequel l'agent interagit.
3. **Action (A) :** Les choix possibles pour l'agent.
4. **État (S) :** La situation actuelle de l'agent dans l'environnement.
5. **Récompense (R) :** Un signal numérique indiquant la qualité d'une action.

L'objectif est de maximiser la somme cumulée des récompenses, souvent appelée valeur à **long terme** [12]**.**

I.3.4.3) Différents types d'algorithmes d'apprentissage par renforcement

1. **Apprentissage basé sur des politiques :** l'agent apprend directement une stratégie qui correspond les états avec les actions.
    - o Exemples : Algorithmes Policy Gradient.
2. **Apprentissage basé sur la valeur** : l'agent apprend une fonction valeur telle que Q(s,a).
    - o Exemple : Q-learning, Deep Q-Networks (DQN).
3. **Méthodes hybrides** : sont une combinaison des deux précédentes (classique et d'inspiration biologique) et sont illustrées par l'algorithme Actor-Critic [13].

I.3.4.4) Applications courantes

- **Robotique** : Entraînements de robots à réaliser des tâches variées (navigation, manipulation).
- **Jeux** : AlphaGo de DeepMind bat les meilleurs joueurs humains au jeu de Go, là où des méthodes et des techniques mixte avaient échouées auparavant.
- **Santé** : Stratégies personnalisées pour optimiser les traitements médicaux.
- **Finance** Apprentissage de stratégies pour le trading automatique [12].

I.4) Applications médicales de l'apprentissage automatique

I.4.1) Diagnostic précoce des maladies

- L'analyse d'images médicales (IRM, scanners, radiographies) via les CNN, avec une précision comparable à celle des radiologues .
- Les modèles (supervisés et non supervisés) permettent la prédiction de maladies chroniques (diabète, maladies cardiovasculaires) à partir de données cliniques.
- L'IA aide à la détection précoce des épidémies (grippe, COVID-19) en croisant données médicales et web.
- Applications spécifiques : détection des nodules pulmonaires en cancérologie, suivi de la progression d'Alzheimer via RNN.
  Impact : Réduction des erreurs, diagnostics plus précis et traitements personnalisés [5], [7], [14].

I.4.2) Prédire l'évolution des maladies

- Les modèles RNN et Transformers analysent les dossiers médicaux pour suivre l'évolution des maladies chroniques.
- Personnalisation des traitements selon les profils cliniques et génétiques (notamment en cancérologie).
- Prédiction des rechutes (dépression, maladies auto-immunes) pour mieux planifier le suivi.
- Utilisation de capteurs portables pour détecter en temps réel des anomalies (rythme cardiaque, glycémie) [4], [5], [7].

I.4.3) Conception de Médicaments et Traitements

- Identification de molécules cibles via les réseaux neuronaux profonds.
- Génération de nouvelles molécules grâce aux GANs et à l'apprentissage par renforcement.
- Simulation des interactions médicamenteuses avec des SVM et CNN pour limiter les tests en laboratoire.
- Médecine de précision : adaptation des traitements au profil génétique du patient [4], [5], [7], [14].

I.4.4) Gestion des Hôpitaux et Soins de Santé

- Prévision des ressources (personnel, matériel, lits) à partir de données historiques.
- Traitement automatique des dossiers médicaux avec le NLP pour structurer les données.
- Recommandation de soins personnalisés selon l'historique du patient.
- Optimisation des coûts par analyse des pratiques hospitalières [5], [7], [12], [14].

Type d'apprentissage	Applications médicales
Apprentissage supervisé	**- Diagnostic automatisé (IRM, radios via CNN)** **- Prédiction de maladies chroniques (diabète, maladies cardiaques)** **- Identification des patients à risque (sepsis)- Personnalisation des traitements (profil génétique)** **- Sélection des participants pour les essais cliniques**
Apprentissage non supervisé	**- Clustering de patients similaires (profils moléculaires)** **- Réduction de dimension pour l'analyse d'images (PCA, autoencodeurs)** **- Découverte de biomarqueurs en génomique** **- Détection d'anomalies dans les résultats médicaux**
Apprentissage par renforcement (RL)	**- Ajustement dynamique des doses médicamenteuses (ex : diabète)** **- Gestion des soins intensifs (ventilation, vasopresseurs)** **- Recommandations cliniques (détection précoce de septicémie)** **- Planification à long terme pour maladies chroniques (ex : cancer)**

Tableau 1 *Résumé des applications médicales des principaux types d'apprentissage automatique*

I.5) Classification

I.5.1) Définition de la classification (Classification)

La classification consiste à prédire à quelle catégorie ou classe appartient une instance de données à partir de ses caractéristiques. Elle constitue une tâche d'apprentissage automatique supervisée sur un ensemble de données étiquetées, dont les classes de nouvelles données non étiquetées, dites non vues, peuvent être prédites. Elle fait partie des problèmes d'apprentissage supervisé parmi les plus courants et elle trouve de nombreuses applications telles que :

- le diagnostic médical.
- la classification automatique d'images.
- la détection de fraude.
- l'analyse des sentiments.

I.5.2) Différents types de classification

Classification binaire (Binary Classification) :Un échantillon est classé dans l'une des deux classes seulement, par exemple : (malade / non malade).

Classification multiclasse (Multiclass Classification) : Classification multiclasse (Multiclass Classification) : Le modèle affecte chaque échantillon à une seule catégorie parmi plusieurs possibles.

Par exemple : différents types de maladies (diabète, hypertension, asthme, etc.).

Classification multi-étiquettes (Multilabel Classification) :Chaque échantillon peut prendre plusieurs étiquettes simultanément, qui ne sont pas discrètes, par exemple un examen médical indiquant plusieurs diagnostics.

Classification multi-sorties (Multi-output Classification) :Semblable au multilabel, mais le modèle prédit simultanément plusieurs variables, chacune ayant son propre type de sortie par exemple : déterminer la maladie et le type d'appareil de diagnostic [5].

I.6) Études antérieures

I.6.1) Résumé des recherches les plus importantes dans le domaine

L'intelligence artificielle appliquée à l'auscultation médicale a beaucoup évolué ces dernières années. De nombreuses études ont mis en évidence l'intérêt d'utiliser des algorithmes d'IA afin d'améliorer la qualité, la rapidité et l'efficacité du diagnostic.

- **Détection des maladies pulmonaires**

Les travaux de Rajpurkar et al. (2017) appliquent les méthodes d'apprentissage profond et plus particulièrement la technique des réseaux de neurones convolutifs pour

le diagnostic d'affections pulmonaires par analyse des radiographies thoraciques avec une précision comparable à celle des radiologues [15].

- **Diagnostic des tumeurs cérébrales**

Menze et al. (2015) proposent des techniques de segmentation utilisant le deep learning pour classifier et circonscrire des tumeurs cérébrales à partir d'images IRM qui a fait référence dans l'utilisation des réseaux neuronaux étendus à l'oncologie [16].

- **Détection précoce des troubles neurologiques**

Zhang et al. (2018) étudient l'interprétation des IRMs fonctionnelles (IRMf) par apprentissage supervisé afin de faire ressortir les activités anormales observées dans des pathologies comme la maladie d'Alzheimer [17].

I.6.2) Techniques d'intelligence artificielle les plus couramment utilisées

De nombreuses techniques d'intelligence artificielle ont été retenues pour établir un diagnostic des maladies, dans les nombreux cas cliniques observés :

- **Réseaux de neurones profonds (Deep Neural Networks)**

Pouvant être utilisés prioritairement pour le traitement des images médicales (radiographies, IRM, …), en raison de leur capacité à identifier des structures complexes dans les données.

- **Forêts aléatoires (Random Forests)**

Adaptées pour l'analyse des données tabulaires colligées au sujet des patients, notamment pour la prédiction des facteurs de risque ou des diagnostics multiples.

- **Machines à vecteurs de support (SVM)**

Utilisées pour classifier les maladies à partir de données structurées ou semi-structurées comme dans le cas de prédiction du cancer ou du diabète.

- **Apprentissage non supervisé (Clustering)**

Appelée aussi apprentissage non supervisé, qui regroupe les patients aux symptômes proches, permettant ainsi d'expliquer les sous-groupes de maladies.

- **Réseaux GAN (Generative Adversarial Networks)**

Qui servent à la génération de données médicales synthétiques, intéressants dans des contextes dont les données réelles sont rares ou difficiles à collecter.

Synthèse des avancées technologiques

Les techniques d'intelligence artificielle ont démontré leur capacité à :

- Améliorer la précision des diagnostics.
- Réduire la charge de travail des cliniciens.
- Offrir des solutions dans des contextes où les ressources médicales sont limitées [7], [15], [16], [17].

I.6.3) Lacunes et problèmes pouvant être améliorés

I.6.3.1) Problèmes de données manquantes

Les données médicales sont souvent incomplètes, ce qui pose un défi important pour les systèmes d'intelligence artificielle. Les raisons possibles de cette incomplétude sont :

- **Sources des données manquantes**

  - Erreurs humaines au niveau de la saisie de l'information.
  - Données non enregistrées pour des raisons technologiques et organisationnelles.
  - Problèmes de partage entre les systèmes de santé.

- **Impacts des données manquantes**
  - **Biais des modèles** : Systèmes d'IA formés sur des données incomplètes et particulièrement incapables de tracker un type de données absent par nature [18].
  - **Diminution de la précision** : Algorithmes qui n'arrivent plus à exploiter l'abondance future de l'information.

  - **Difficulté à interpréter les résultats** : Pourra être rendue compliquée sous des données non complètes.

- **Solutions possibles**
  - **Imputation des données** : l'aide de plusieurs techniques d'imputation du k-plus proches voisins (KNN) ou des modèles probabilistes [19].
  - **Collecte de données standardisée** via des protocoles uniformes au niveau de la saisie et du stockage des données médicales [20].
  - **Modèles robustes** : des algorithmes capables de s'adapter aux données manquantes [7].

I.6.3.2) Précision limitée des modèles actuels

Les modèles d'intelligence artificielle, malgré leur puissance, peuvent présenter certaines limites, notamment dans le domaine médical :

- **Sources de limitation :**
  - **Hétérogénéité des données** Les données médicales proviennent de sources variées (imagerie, données cliniques, génomiques…) et le traitement analytique est délicat.
  - **Biais des données** : Les ensembles de données ne représentent pas nécessairement de manière équitable toutes les populations, ce qui accentue les inégalités de la prédiction.
- **Conséquences :**
  - **Précision réduite :** modèles en mesure d'identifier tout changement d'une certaine normalité pour une donnée, mais incapables d'obtenir le bon diagnostic
  - **Manque de généralisation** : un modèle IA formé sur un jeu de données ne parvient pas à reproduire les mêmes performances avec un autre échantillon.
- **Améliorations proposées**
  - **Augmentation des données** : Création de données synthétiques via des réseaux antagonistes génératifs (GANs) pour élargir les ensembles d'entraînement.
  - **Apprentissage fédéré** : Formation de modèles à partir de données réparties sur plusieurs sources tout en préservant la confidentialité des patients.
- **Optimisation des algorithmes** : Utilisation de mécanismes comme l'apprentissage par transfert pour améliorer les performances des modèles sur des ensembles limités [21], [22], [23], [24].

I.7) Conclusion

Dans ce premier chapitre, nous avons introduit les concepts de base liés à l'intelligence artificielle, et plus particulièrement à l'apprentissage automatique et à la classification.

Nous avons également présenté les différents types d'apprentissage (supervisé, non supervisé), ainsi que l'intérêt croissant de ces techniques dans le domaine médical. Aujourd'hui, l'apprentissage automatique joue un rôle important dans le soutien au diagnostic, la prédiction des maladies, et l'amélioration de la qualité des soins.

Dans les chapitres suivants, nous allons approfondir les différentes méthodes de classification utilisées dans notre travail, ainsi que le développement de la base de données exploitée pour nos expérimentations.

Chapitre II. Méthodologie de classification et préparation des données

II.1) Introduction

Les techniques d'apprentissage automatique, et plus particulièrement la classification, jouent un rôle essentiel dans le domaine médical, notamment pour le diagnostic assisté par ordinateur. Dans ce chapitre, nous nous intéressons à trois méthodes de classification largement reconnues pour leur efficacité : **SVM (Support Vector Machine), XGBoost** et **Random Forest.** L'objectif est de comparer leurs performances afin d'identifier celle qui s'adapte le mieux à notre cas d'usage médical.

Nous présenterons également deux bases de données issues du **UCI Machine Learning Repository** : le **Heart Disease Dataset** et **l**e **Breast Cancer Wisconsin Dataset,** qui seront utilisées pour nos expérimentations. Les étapes de nettoyage, de traitement et d'amélioration des données y seront abordées afin d'optimiser leur qualité. Enfin, nous décrirons le processus de développement de notre propre base de données médicale, en mettant l'accent sur les techniques utilisées pour assurer une préparation efficace et une analyse fiable.

II.2) Méthodes de classification utilisée

II.2.1) Machine à Vecteurs de Support (SVM)

II.2.1.1) Définition de SVM

La SVM (Support Vector Machine) s'avère être l'un des algorithmes de classification les plus puissants de l'apprentissage automatique qui repose sur la recherche du meilleur hyperplan entre différentes classes de données dans l'espace des caractéristiques selon un critère de séparation par la marge d'un hyperplan, c'est-à-dire que l'algorithme vise à maximiser la distance entre cet hyperplan et les points des classes les plus proches (les vecteurs de support) or, l'entraînement de ce modèle correspond à un problème d'optimisation mathématique où, dans la majorité des cas, seule une infime fraction des données (les vecteurs de support) est prise en compte pour obtenir la solution et réaliser le modèle.

- **Paramètres essentiels du SVM :**

Lors de la configuration d'un modèle **SVM (Support Vector Machine)**, plusieurs paramètres doivent être ajustés :

- **C** : Contrôle la régularisation et l'équilibre entre marge et erreurs de classification.

- **Kernel** : Détermine la fonction de séparation (linéaire, polynomial, RBF, sigmoïde).
- **Gamma** : Impacte d'influence des points de données (pour rbf et poly).
- **Degree :** Définit le degré des polynômes (pour poly).
- **Class Weight** : Ajuste l'importance des classes en cas de déséquilibre.
- **Tol** : Seuil d'arrêt de l'optimisation.
- **Max Iter** : Nombre maximal d'itérations du modèle [25].

II.2.1.2) Caractéristiques de SVM (Support Vector Machine)

- La précision du modèle est améliorée grâce à **la maximisation de la marge** entre les différentes classes, à la minimisation des erreurs et à l'amélioration de la généralisation.
- Repose uniquement sur un nombre limité de points critiques (**vecteurs de support**), ce qui le rend plus simple et plus efficace.
- Fonctionne bien dans les **espaces de grande dimension grâce** à l'utilisation de fonctions noyau, sans calcul de coordonnées.
- Propose un système permettant de **contrôler la complexité** du modèle tout en **minimisant les erreurs** par maximisation de la marge.
- **Gère les données non linéairement séparables** en autorisant certaines erreurs grâce à la notion de marge souple.
- Fait preuve d'une **grande précision dans les tâches de classification**, notamment dans le domaine du traitement de texte, de l'analyse d'images et de l'évaluation génomique.
- Repose sur des **bases mathématiques solides** telles que l'optimisation convexe et la théorie de Vapnik-Chervonenkis.
- Peut être appliqué dans de nombreux domaines comme la médecine, la cybersécurité, l'agriculture intelligente et le traitement du langage naturel [25].

II.2.1.3) Pourquoi l'avons-nous choisie pour ce projet ?

Parce qu'elle est très efficace pour distinguer deux classes (malade / non malade), en particulier dans les cas médicaux simples où les caractéristiques sont bien définies. La SVM repose sur la maximisation de la marge entre les classes, ce qui en fait un choix idéal lorsque les données présentent une séparation nette. Cela nous permet de commencer l'évaluation du modèle dans un environnement contrôlé, où les résultats peuvent être interprétés plus facilement. Grâce à sa robustesse et à sa précision dans les tâches de classification binaire, elle constitue une base solide avant de passer à des situations plus complexes ou à des algorithmes plus avancés .

II.2.1.4) Types de données pour lesquelles SVM est efficace

- Efficace lorsque les données sont clairement étiquetées (notamment pour la classification binaire).
- Capable de traiter des **données séparables de manière non linéaire** grâce aux fonctions noyaux.
- Se connecte aux **données de grande dimension**, telles que les données génétiques en bio-informatique.
- A produit d'excellents résultats lorsque les classes étaient divisées par des **marges claires**.
- Utilise le concept de « **marge souple** » pour tolérer un certain niveau de bruit.

Avec de nombreuses caractéristiques, il fonctionne bien même avec peu d'exemples, car il ne dépend que des vecteurs de support [25].

II.2.2) Xgboost

II.2.2.1) Définition de xgboost

XGBoost est un outil important dans le domaine de l'apprentissage supervisé, offrant des performances de pointe pour les tâches de classification, de régression et de classement. Il s'agit d'une implémentation d'un algorithme de gradient boosting généralisé qui est devenu un choix privilégié dans les compétitions de machine learning.

Malgré ses bonnes performances par rapport aux autres implémentations de gradient boosting existantes, XGBoost peut être très coûteux en temps d'exécution. Les tâches courantes peuvent prendre des heures, voire des jours, à s'exécuter. La construction de modèles très précis à l'aide du gradient boosting nécessite également un réglage approfondi des hyperparamètres. Dans ce processus, l'algorithme doit être exécuté de nombreuses fois pour explorer l'effet de paramètres tels que le taux d'apprentissage (learning rate) et les termes de régularisation L1/L2 sur la précision de la validation croisée.

Paramètres essentiels de XGBoost

Lors de la configuration d'un modèle XGBoost, plusieurs paramètres doivent être ajustés pour optimiser ses performances :

- **n_estimators** : Nombre d'arbres dans le modèle.
- **learning_rate** : Taux d'apprentissage contrôlant l'impact de chaque arbre.
- **max_depth** : Profondeur maximale des arbres, influençant la complexité du modèle.
- **min_child_weight** : Limite minimale du poids des observations dans un nœud avant la division.

- **gamma** : Paramètre de régularisation contrôlant la réduction de la perte requise pour scinder un nœud.
- **subsample** : Fraction des données utilisées pour entraîner chaque arbre, limitant le sur-ajustement.
- **colsample_bytree** : Proportion des caractéristiques sélectionnées à chaque arbre.
- **lambda** : Paramètre de régularisation L2 (Ridge), réduisant la complexité des arbres.
- **alpha** : Paramètre de régularisation L1 (Lasso), favorisant la sélection de caractéristiques pertinentes.
- **scale_pos_weight** : Ajuste l'influence des classes déséquilibrées, utile pour les données fortement biaisées [26], [27].

II.2.2.2) Caractéristiques de XGBoost

- XGBoost **améliore la précision des prédictions** en corrigeant les erreurs cumulées des modèles faibles.
- Elle utilise des techniques de **régularisation L1 et L2** pour éviter le risque de surapprentissage, ce qui lui confère une efficacité optimale pour les jeux de données complexes.
- Elle **arrête automatiquement l'apprentissage** lorsque les performances ne s'améliorent plus, économisant ainsi du temps et des ressources.
- Elle calcule la contribution de chaque attribut à la prédiction, ce qui facilite l'interprétation du modèle.
- Elle **gère automatiquement les valeurs manquantes**, ce qui la rend idéale pour le traitement de données réelles avec des valeurs manquantes.
- Elle utilise un **traitement parallèle** pour construire les arbres de manière efficace, ce qui est particulièrement efficace pour les jeux de données volumineux.
- Elle est également responsable des tâches de classification et de régression, et peut être mise en œuvre dans différents langages de programmation, ce qui renforce sa **flexibilité** [26].

II.2.2.3) Pourquoi l'avons-nous choisie pour ce projet ?

Les raisons les plus solides d'opter pour XGBoost sont :

- Des performances prédictives élevées :
- Gestion des données déséquilibrées
- Résistance à surapprentissage (l'Overfitting)
- Importance des caractéristiques
- Rapidité et efficacité

II.2.2.4) Types de données pour lesquelles XGBoost est efficace

- XGBoost gère les **variables numériques** continues et discrètes, ce qui le rend idéal pour l'analyse de grands ensembles de données numériques.
- Il peut gérer les **variables catégorielles** sans recourir à une méthode d'encodage complexe comme le codage One-Hot.
- Il gère efficacement les **valeurs manquantes** et les **données lucides**, ce qui le rend idéal pour les systèmes de recommandation ou le traitement de texte.
- Il peut être utilisé avec des **données temporelles**, à condition qu'elles soient reformulées pour l'apprentissage supervisé.
- Largement utilisé dans les projets de science des données, il est très efficace avec les **données structurées** et tabulaires [26].

II.2.3) Forêt aléatoire (Random Forest)

II.2.3.1) Définition de Forêt aléatoire (Random Forest)

La forêt d'arbres de décision (Forêt aléatoire, RF) désigne une méthode d'apprentissage d'ensemble qui regroupe plusieurs arbres de décision. En effet, un arbre est un classificateur faible en raison de sa grande variance. Alors pour pallier ce problème, la RF forme une forêt générée par un très grand nombre d'arbres de décision. Le but de la RF est de faire croître chaque arbre de décision indépendamment des autres, grâce à la méthode de rééchantillonnage (bagging).

Paramètres essentiels de Forêt aléatoire

Lors de la configuration d'un modèle **Forêt aléatoire**, plusieurs paramètres doivent être ajustés pour optimiser ses performances :

- **n_estimators** : Nombre d'arbres dans la forêt.
- **max_depth** : Profondeur maximale des arbres, influençant la complexité du modèle.
- **min_samples_split :** Nombre minimal d'échantillons requis pour diviser un nœud.
- **min_samples_leaf** : Nombre minimal d'échantillons dans une feuille.
- **max_features** : Nombre de caractéristiques prises en compte à chaque division.
- **bootstrap** : Active ou désactive l'échantillonnage avec remise.
- **criterion** : Fonction utilisée pour mesurer la qualité d'une scission (gini ou entropy).
- **class_weight** : Ajuste l'importance des classes en cas de déséquilibre.
- **random_state** : Fixe la graine aléatoire pour la reproductibilité des résultats.

II.2.3.2) Caractéristiques de Forêt aléatoire

- Fusionner plusieurs arbres de décision pour obtenir une meilleure précision et réduire le surapprentissage.
- Utilise l'échantillonnage aléatoire (**ensachage**) pour créer des arbres diversifiés.
- Crée un modèle à maillage plus solide et généralisable à travers l'agrégation des prévisions.
- Met en évidence les variables les plus déterminantes de la prise de décision.
- Diminue les effets des données bruitées ou aberrantes.
- Il est aussi applicable à des tâches **de classification** ainsi qu'à une **régression** à différents types de données.
- En donnant des bons résultats en **détection des vrais positifs**, ce qui est particulièrement utile pour des situations sensibles telles que la maintenance prédictive [28].

II.2.3.3) Pourquoi l'avons-nous choisie pour ce projet Forêt aléatoire

Nous avons choisi l'algorithme Forêt aléatoire car il est particulièrement adapté à une application de diagnostic des maladies, pour les raisons suivantes :

- **Excellente performance en classification.**
- **Résistance au surapprentissage.**
- **Capacité à gérer des données médicales complexes** : Il peut traiter efficacement à la fois des données numériques (comme les analyses) et catégorielles (comme le sexe ou les symptômes).
- **Grande flexibilité.**

En résumé, le choix de Forêt aléatoire pour ce projet a été motivé par ses hautes performances, sa robustesse, sa capacité à gérer des données complexes et l'interprétabilité qu'il offre, ce qui en fait une option appropriée pour relever les défis de la maintenance prédictive des ponts [28].

II.2.3.4) Types de données pour lesquelles la Forêt aléatoire est efficace

- Cela fonctionne bien pour les tâches de **classification et de régression**, que les données soient continues ou discrètes.
- Utile avec des données de **haute dimension**, surtout quand il y a peu de variables qui ont un impact significatif.
- Gère bien les **valeurs manquantes** sans perte de performance.
- Capable de créer des modèles qui montrent comment les variables interagissent de **façon non linéaire.**
- Vous aide à **trouver les variables les plus importantes**, ce qui est utile pour comprendre et prendre des décisions.
- Moins susceptible au surapprentissage que les arbres de décision individuels [29].

II.3) Bases de données médicales

II.3.1) Bases de UCI

II.3.1.1) Première base de données « Breast Cancer Wisconsin (Diagnostic) »[30]

A. Description de la base de données

Nom de la base : Breast Cancer Wisconsin (Diagnostic Dataset)

Source : UCI Machine Learning Repository

Domaine : Diagnostic du cancer du sein

Nombre d'instances : 569 échantillons

Nombre de caractéristiques : 30 caractéristiques numériques

Type de tâche : Classification binaire (tumeur bénigne / tumeur maligne)

Les caractéristiques sont extraites à partir d'images numérisées d'aspirations à l'aiguille fine (FNA) de masses mammaires. Dix caractéristiques sont calculées pour chaque noyau cellulaire, donnant lieu à 30 variables au total (moyenne, erreur standard, et « worst » pour chacune).

Exemples de caractéristiques :

- Radius (moyenne des distances du centre au contour)
- Texture (écart-type des niveaux de gris)
- Perimeter
- Area
- Smoothness (variation locale du rayon)
- Compactness, Concavity, Symmetry, Fractal dimension, etc.

Classe cible :

- 'M' pour tumeur maligne
- 'B' pour tumeur bénigne

Observations :

- Les valeurs sont toutes numériques (réelles)
- Pas de valeurs manquantes

B. Nettoyage et préparation de la base de données Breast Cancer Wisconsin

La base de données **Breast Cancer Wisconsin** contient 569 échantillons avec 30 caractéristiques numériques, utilisée pour une tâche de classification binaire (tumeur bénigne ou maligne). Avant l'entraînement des modèles, il est essentiel d'effectuer

des opérations de nettoyage et de préparation afin d'assurer la qualité des données et la fiabilité des résultats.

1- Suppression des colonnes non pertinentes

- Le premier attribut est un identifiant unique (ID) pour chaque patient.
- Cette colonne ne porte pas d'information utile pour la modélisation et est donc supprimée afin d'éviter toute confusion ou biais lors de l'entraînement.

Figure 1 *Suppression des colonnes non pertinentes.*

2- Encodage de la variable cible (diagnostic)

- La deuxième colonne contient le diagnostic sous forme de texte : 'M' pour tumeur maligne, 'B' pour tumeur bénigne.
- Cette variable est convertie en numérique : 1 pour maligne, 0 pour bénigne, afin d'être compatible avec les algorithmes d'apprentissage.

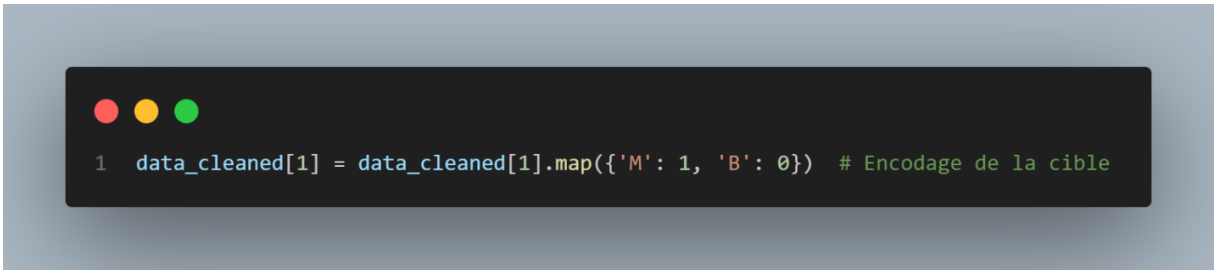


Figure 2 *Encodage de la variable cible (diagnostic).*

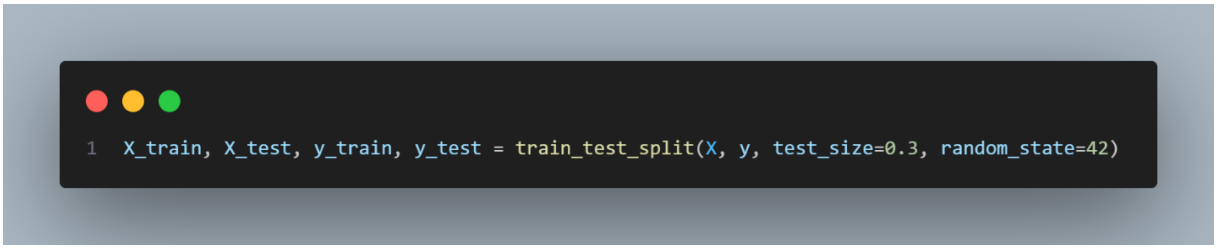
3- Séparation des variables explicatives et de la cible

- X représente les caractéristiques cliniques (toutes les colonnes sauf la cible).
- y correspond à la variable cible (diagnostic).

Figure 3 *Séparation des variables explicatives et de la cible.*

4- Division des données en ensembles d'entraînement et de test

- Pour garantir une évaluation objective, les données sont divisées en 70% pour l'entraînement et 30% pour le test.
- L'utilisation du paramètre random_state assure la reproductibilité de cette division.

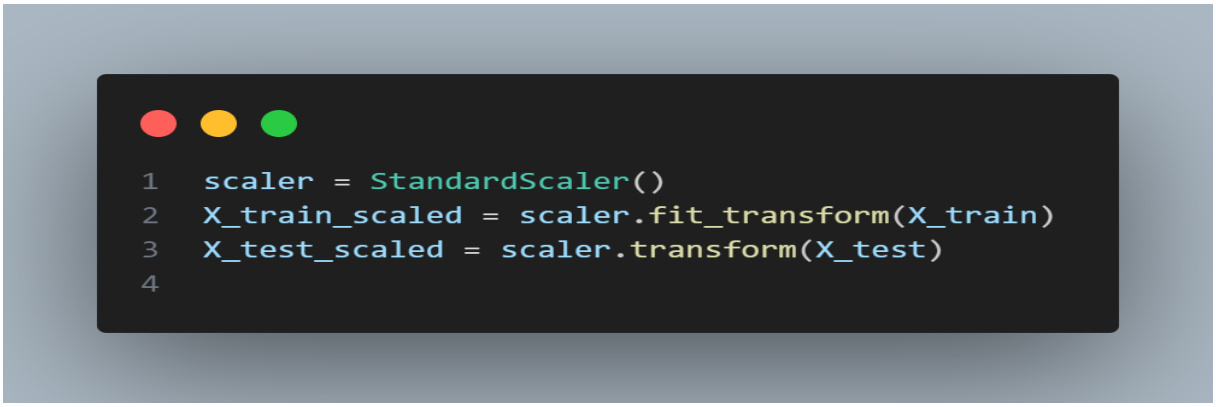


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 4 *Division des données en ensembles d'entraînement et de test.*

5- Normalization des variables

- Certaines méthodes, notamment la SVM, sont sensibles à l'échelle des variables.
- Une normalisation est donc appliquée pour que chaque caractéristique ait une moyenne nulle et un écart-type égal à un.
- La normalisation est ajustée sur les données d'entraînement, puis appliquée aux données de test.



```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figure 5 *Normalisation des variables.*

• Remarques sur le nettoyage dans ce contexte

- La base Breast Cancer Wisconsin est connue pour ne pas contenir de valeurs manquantes dans ce format, ce qui explique l'absence de traitement explicite des données manquantes.
- Si des valeurs manquantes apparaissent dans d'autres bases, il faudra alors prévoir un traitement adapté.
- Les variables explicatives retenues pour l'apprentissage sont toutes numériques. La variable cible "diagnostic", de type nominal au départ ('M' ou 'B'), a été encodée en binaire (1/0) pour être compatible avec les algorithmes d'apprentissage.

II.3.1.2) Deuxième base de données « Heart Disease Dataset (cleveland)» [31]

A. Description de la base de données

Nom de la base : Heart Disease Dataset

Source : UCI Machine Learning Repository

Domaine : Diagnostic des maladies cardiaques

Nombre d'instances : 303 échantillons

Nombre de caractéristiques : 13 caractéristiques cliniques

Type de tâche : Classification multi-classes (5 classes)

Principales caractéristiques :

Age : âge du patient (années).

Sexe : sexe (1 = homme, 0 = femme).

Chest Pain Type (cp) : type de douleur thoracique.

Resting Blood Pressure (trestbps) : pression artérielle au repos (mmHg).

Serum Cholesterol (chol) : taux de cholestérol sérique (mg/dl).

Fasting Blood Sugar (fbs) : glycémie à jeun (> 120 mg/dl).

Resting ECG (restecg) : électrocardiogramme au repos.

Max Heart Rate Achieved (thalach) : fréquence cardiaque maximale atteinte.

Exercise Induced Angina (exang) : angine provoquée par l'exercice.

Oldpeak : dépression du segment ST.

Slope : pente du segment ST.

Number of Major Vessels (ca) : nombre de vaisseaux principaux colorés.

Thalassemia (thal) : présence de thalassémie.

Observations :

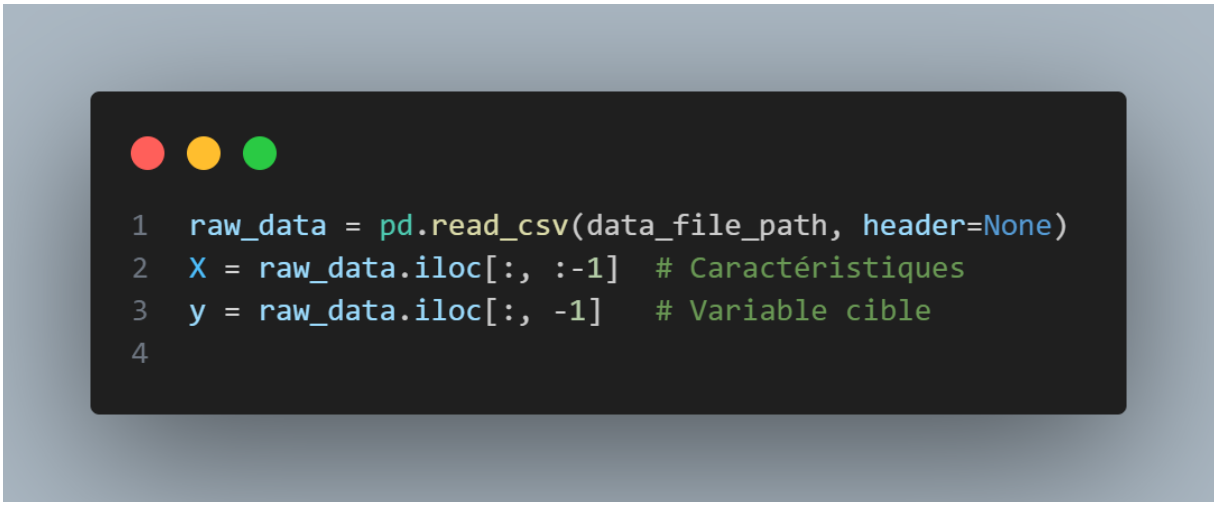
La variable cible (target) prend les valeurs de 0 à 4, représentant différents niveaux de sévérité de la maladie cardiaque.

B. Nettoyage et préparation de la base de données (Heart Disease Dataset)

La base de données **Heart Disease Dataset** comprend 303 échantillons avec 13 caractéristiques cliniques, destinés à une tâche de classification multi-classes (5 classes de sévérité).

1- Chargement des données

- Les données sont chargées à partir d'un fichier CSV brut.
- La variable cible se trouve dans la dernière colonne, tandis que toutes les autres colonnes représentent les caractéristiques explicatives.



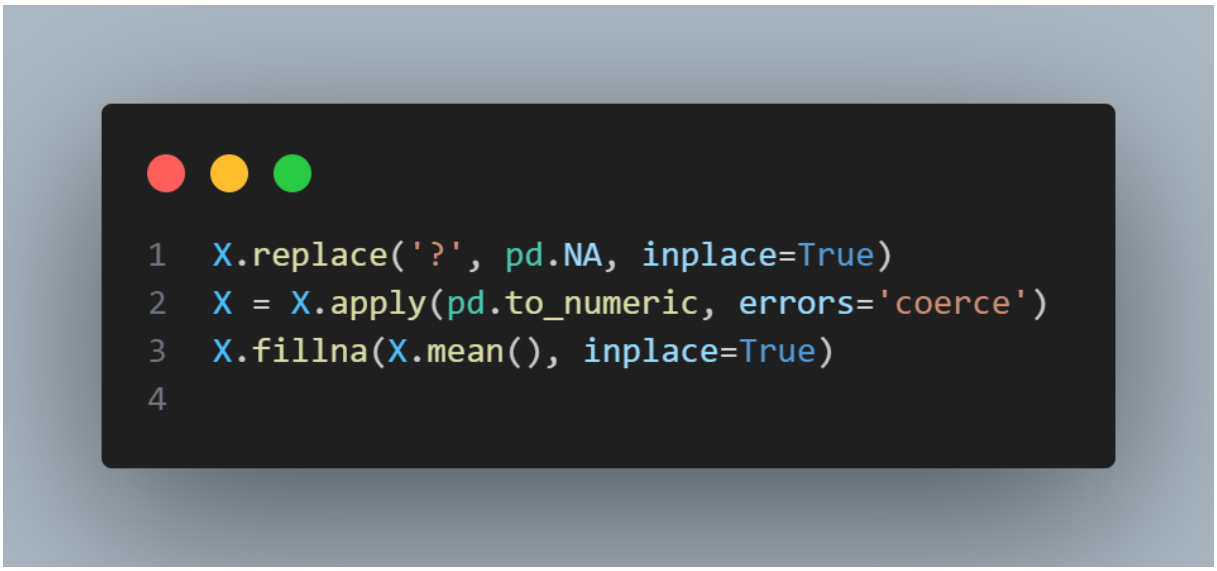
```
raw_data = pd.read_csv(data_file_path, header=None)
X = raw_data.iloc[:, :-1]  # Caractéristiques
y = raw_data.iloc[:, -1]   # Variable cible

```

Figure 6 *Chargement des données*

2- Gestion des valeurs manquantes et aberrantes

- Certaines colonnes contiennent des valeurs manquantes représentées par le caractère '?' ou des données non numériques.
- Ces valeurs sont d'abord remplacées par des NaN (pd.NA), puis converties en valeurs numériques avec pd.to_numeric (les erreurs deviennent NaN).
- Les valeurs manquantes sont ensuite imputées en remplaçant par la moyenne de chaque colonne, ce qui permet de conserver toutes les observations.



```
X.replace('?', pd.NA, inplace=True)
X = X.apply(pd.to_numeric, errors='coerce')
X.fillna(X.mean(), inplace=True)

```

Figure 7 *Gestion des valeurs manquantes et aberrantes*

3- Équilibrage des classes avec SMOTE

- Étant donné le déséquilibre potentiel des classes dans la variable cible, la technique **SMOTE** (Synthetic Minority Over-sampling Technique) est utilisée pour générer des exemples synthétiques des classes minoritaires.
- Cela permet d'améliorer la qualité de l'apprentissage, en évitant que le modèle ne soit biaisé vers la classe majoritaire.

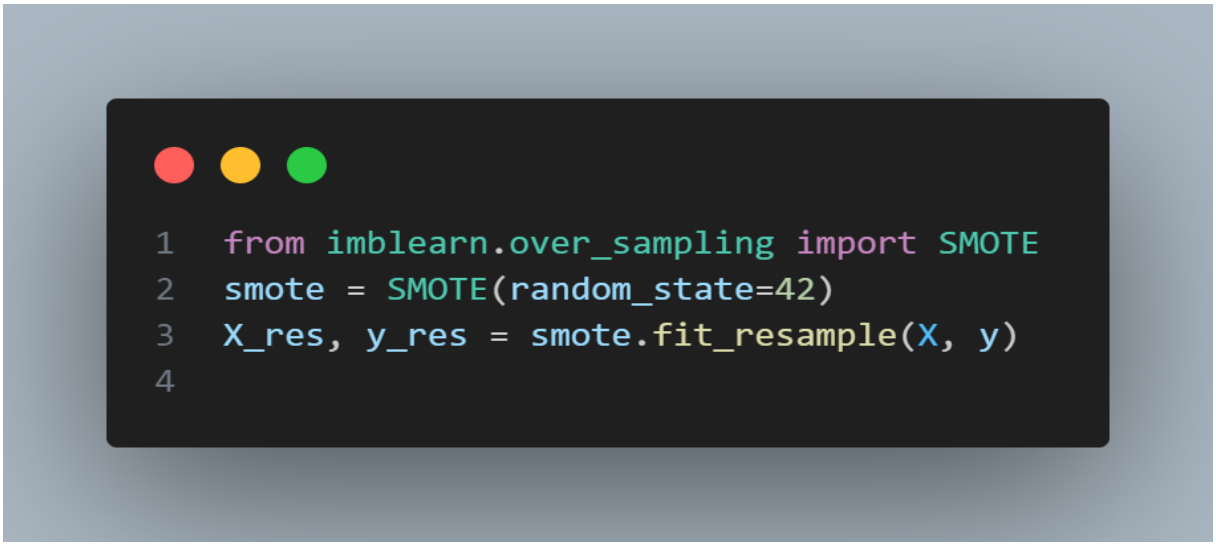


Figure 8 Équilibrage des classes avec SMOTE

4- Division des données en ensembles d'entraînement et de test

- Les données équilibrées sont divisées en 80% pour l'entraînement et 20% pour le test.
- La division est aléatoire mais reproductible grâce au paramètre random_state.

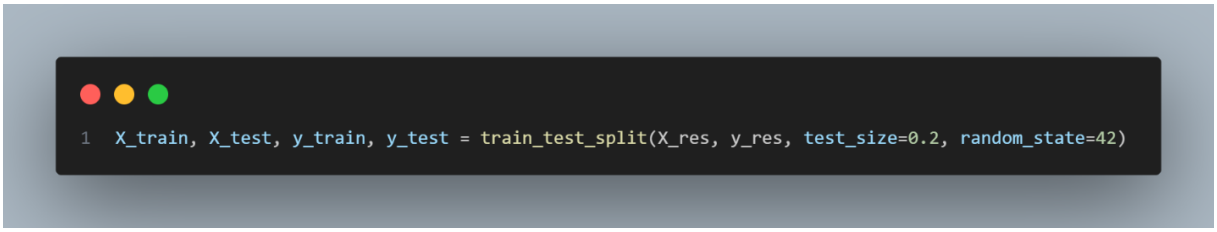
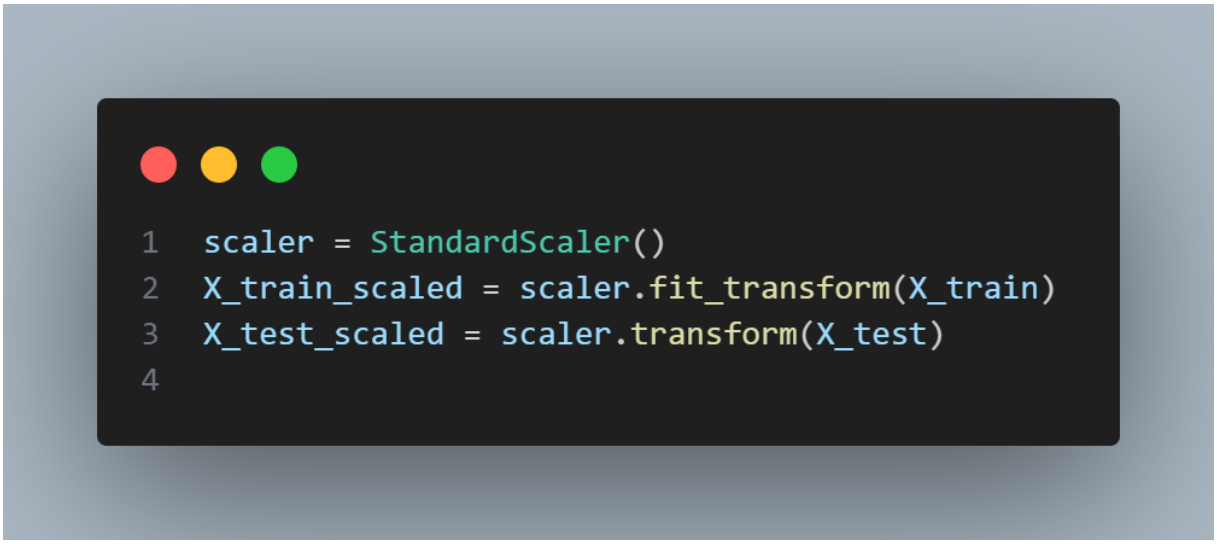


Figure 9 Division des données en ensembles d'entraînement et de test

5- Normalisation des caractéristiques

- La normalisation est appliquée aux données d'entrée afin d'assurer une meilleure convergence des algorithmes, surtout pour les modèles sensibles à l'échelle des données comme le SVM.
- Le scaler est ajusté sur l'ensemble d'entraînement et appliqué à l'ensemble de test.



```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figure 10 *Normalisation des caractéristiques*

Remarques importantes :

- Cette procédure de nettoyage et de préparation garantit que les modèles apprendront à partir de données fiables, équilibrées et comparables.
- Le recours à SMOTE est particulièrement utile pour les tâches multi-classes où certaines classes peuvent être sous-représentées.
- La normalisation des données améliore la performance des modèles basés sur les distances, notamment le SVM.

II.3.2) Développement de notre base de donne

II.3.2.1) Introduction et définition de la base de données

Le but de la base de données médicale développée dans cette étude est de constituer une source à la fois fiable et réaliste pour entraîner des modèles d'intelligence artificielle dans le domaine de la santé, en s'appuyant sur des données de santé diverses et complètes, dans le cadre d'un effort d'intégration. Cette base de données a été alimentée dans le cadre d'une collaboration directe avec de nombreux médecins et cliniques spécialisées, supervisée par le Dr Amine Lachlah (médecin généraliste), et le Dr Tarek Alia (spécialiste en odontologie).

La collecte des données a été réalisée en ayant recours à de multiples sources : dossiers de patients imprimés, documents manuscrits ainsi que des bases de données électroniques exportées au format SQL depuis les archives des médecins. Les données ont ensuite été soigneusement triées et nettoyées, avec une sélection précise de **500 dossiers médicaux** par maladie étudiée, ce qui a permis d'exclure les cas présentant des informations incomplètes ou erronée. La sélection des maladies à étudier était hiérarchisée en fonction de la disponibilité d'un nombre suffisant de dossiers complets pour chaque des maladies étudiée, afin de garantir une représentation réaliste et exhaustive de chaque cas dans la base de données.

La confidentialité et la sécurité des données ont été scrupuleusement respectées, toutes les informations personnelles identifiables des patients ayant été supprimées,

afin de préserver la confidentialité et de concentrer l'entraînement des modèles sur les aspects cliniques et symptomatiques, évitant ainsi tout biais lié aux données personnelles.

II.3.2.2) Caractéristiques générales de la base de données :

- **Nombre de médecins participants** : 25 médecins issus de différentes spécialités ont contribué à la collecte des données.
- **Nombre de maladies incluses** : 98 maladies communes et chroniques.
- **Nombre d'enregistrements** : 500 dossiers médicaux par maladie, soit un total de 49000 enregistrements.
- **Nature des données** : Données démographiques (âge, sexe, état civil), antécédents médicaux, symptômes, diagnostics, traitements suivis, et mode de vie.
- **Format :** fichiers CSV tabulaires.
- **Sources des données** :
    - Dossiers imprimés et manuscrits provenant de différentes cliniques.
    - Bases de données électroniques au format SQL extraites des archives médicales.
- **Traitement des données** : Nettoyage et correction des données collectées, élimination des valeurs manquantes et incorrectes, préparation homogène pour une analyse scientifique rigoureuse.
- **Supervision scientifique** : Le Dr Amin Lachlah et le Dr Tarek Alia ont supervisé l'ensemble du processus pour garantir la qualité et la fiabilité des données utilisées à des fins de recherche.
- **Protection de la vie privée** : Toutes les données personnelles sensibles ont été supprimées conformément aux normes éthiques et réglementaires en matière de confidentialité.

II.3.2.3) Structure de la base de données

Pour comprendre la nature des données et les analyser de manière efficace, il est essentiel de fournir des informations sur l'architecture de la base de données ainsi que sur les caractéristiques (features) qu'elle contient. Le choix et la sélection des caractéristiques doit s'appuyer sur deux critères : leur pertinence clinique pour le diagnostic et leur accessibilité constante dans les dossiers médicaux. Le but ici est d'énoncer les principales caractéristiques des données collectées et d'indiquer le type de chaque caractéristique dans le but de permettre une analyse et une modélisation rigoureuses.

Des caractéristiques composant notre base

La base de données est composée d'un ensemble de colonnes fondamentales représentant les informations personnelles et médicales de chaque patient, notamment :

- **Âge (Âge)** : variable quantitative représentant l'âge du patient en années, influençant la probabilité de survenue et le profil des symptômes.
- **Sexe (Sexe)** : variable qualitative binaire (homme/femme) impactant la manifestation de certaines maladies et la réponse aux traitements.
- **Grossesse (Enceinte)** : variable qualitative indiquant l'état de grossesse chez les femmes, ayant une influence médicale particulière.
- **Taille (Taille (cm))** : variable quantitative exprimant la taille du patient en centimètres, utilisée pour calculer des indicateurs de santé comme l'IMC.
- **Poids (Poids (kg))** : variable quantitative indiquant le poids du patient en kilogrammes.
- **Indice de masse corporelle (IMC**) : variable quantitative calculée à partir de la taille et du poids, reflétant l'état nutritionnel et la santé globale.
- **Groupe sanguin (Groupe Sanguin)** : variable qualitative représentant le groupe sanguin, pouvant être liée à certaines maladies ou réactions médicamenteuses.
- **État civil (Statut Matrimonial)** : variable qualitative décrivant la situation matrimoniale du patient (marié, célibataire, divorcé, veuf).
- **Pratique sportive (Sport)** : variable qualitative indiquant si le patient pratique régulièrement une activité physique, facteur influant sur la santé générale.
- **Profession (Profession)** : variable textuelle reflétant la profession exercée, pouvant influencer l'exposition à certains facteurs de risque.
- **Maladies chroniques (Maladies Chroniques)** : variable textuelle listant les maladies chroniques dont souffre le patient, influençant diagnostic et traitement.
- **Prise régulière de médicaments (Prend Médicaments)** : variable qualitative indiquant si le patient prend régulièrement des médicaments spécifiques aux maladies chroniques.
- **Liste des médicaments (Liste Médicaments)** : variable textuelle détaillant les médicaments consommés par le patient.
- **Allergies médicamenteuses (Allergies Médicamenteuses**) : variable textuelle identifiant d'éventuelles allergies à certains médicaments.
- **Antécédents chirurgicaux (Antécédents Chirurgicaux)** : variable textuelle décrivant les interventions chirurgicales antérieures.

- **Dispositifs médicaux implantés (Dispositif Médical Implanté)** : variable textuelle indiquant la présence de dispositifs médicaux implantés tels que pacemakers ou pompes à insuline.
- **Tabagisme (Fumeur)** : variable qualitative précisant l'état de fumeur.
- **Nombre de cigarettes par jour (Cigarettes/Jour)** : variable quantitative précisant la consommation journalière de cigarettes.
- **Consommation d'alcool (Alcool) :** variable qualitative indiquant si le patient consomme de l'alcool.
- **Antécédents familiaux (Antécédents Familiaux) :** variable textuelle précisant la présence de maladies héréditaires ou chroniques dans la famille.
- **Symptômes récurrents (Symptômes Récurrents**) : variable textuelle enregistrant les symptômes apparaissant de manière continue ou répétée.
- **Qualité du sommeil (Sommeil (heures))** : variable quantitative indiquant le nombre d'heures de sommeil quotidien.
- **Symptômes actuels (Symptômes Actuels)** : variable textuelle listant les symptômes dont souffre actuellement le patient.
- **Intensité des symptômes (Intensité Symptômes)** : variable qualitative décrivant la gravité des symptômes (légère, modérée, sévère).
- **Durée d'apparition des symptômes (Durée Apparition Symptômes (jours))** : variable quantitative indiquant le nombre de jours depuis le début des symptômes.
- **Diagnostic (Diagnostic)** : variable textuelle contenant le diagnostic final de la maladie.
- **Traitement prescrit (Traitement Prescrit)** : variable textuelle détaillant les médicaments ou interventions thérapeutiques prescrits.
- **Classe des médicaments (Classe Médicaments)** : variable textuelle décrivant les familles médicamenteuses utilisées sans mentionner les noms spécifiques.
- **Conseils médicaux (Conseils Médicaux)** : variable textuelle comprenant des recommandations générales au patient telles que le régime alimentaire, la période de repos, etc.
- **Méthode de traitement (Méthode de traitement)** : variable textuelle décrivant la méthode générale de traitement pour chaque maladie.

Critères de sélection des caractéristiques

La sélection des caractéristiques repose sur deux critères essentiels :
- Leur pertinence clinique liée à l'examen médical pratiqué pour établir le diagnostic.
- Et leur disponibilité dans les données recueillies lors de l'enquête sur la pratique des médecins.

Ces caractéristiques sont constituées des données sociodémographiques, des antécédents médicaux connus, des symptômes présents ou passés, ainsi que de données sur le mode de vie et des facteurs environnementaux. Elles donnent au médecin une image fidèle de l'état du patient, participent à la qualité de la base de données et contribuent à la performance des modèles prédictifs en médecine [32], [33].

II.3.2.4) Processus de création de la base de données

A. Sources des données initiales

Les données ont été collectées directement à partir des dossiers de patients réels auprès d'un groupe de médecins et de cliniques spécialisées. Ces dossiers comprenaient différents types de sources, notamment des dossiers imprimés, des documents manuscrits ainsi que des bases de données électroniques obtenues au format SQL à partir des archives des médecins. Vingt cinq médecins de différentes spécialités ont participé à la collecte des données afin d'assurer la diversité et l'exhaustivité des informations.

B. Choix des distributions d'âge, des symptômes et des diagnostics

Les patients ont été sélectionnés parmi différentes tranches d'âge afin de garantir une représentation complète et équilibrée de toutes les catégories démographiques, reflétant ainsi la diversité naturelle des maladies. Les symptômes et diagnostics proviennent de rapports cliniques réels et ont été préparés avec rigueur, reflétant la condition médicale exacte des patients. La validation des diagnostics a été assurée par une vérification médicale directe et une supervision scientifique par une équipe médicale spécialisée.Dr Amine Lachlah (médecin généraliste), et le Dr Tarek Alia (spécialiste en odontologie)).

C. Étapes de nettoyage et de traitement des données

Les données incomplètes ou partiellement manquantes ont été exclues afin d'assurer la qualité des informations. Des corrections d'erreurs et des incohérences ont été effectuées à travers plusieurs revues et des techniques de correction spécialisées. En unifiant les maladies et les symptômes et en résolvant les problèmes d'incompatibilité des symptômes avec la maladie. Après le nettoyage, les données ont été formatées et les variables textuelles ont été encodées en formats numériques adaptés à l'analyse statistique et à l'entraînement des modèles.

D. Garantie de la qualité et du réalisme des données

Le processus de nettoyage et de validation médicale a été supervisé par le **Dr. Amine Lachlah**, médecin généraliste, et le **Dr.Tarek Alia**, spécialiste en odontologie, afin de garantir la précision et la fiabilité des données. Un contrôle continu a été mis en place pour s'assurer de l'absence de biais ou d'erreurs susceptibles d'affecter les résultats

de l'analyse. Des critères stricts ont conduit à l'exclusion des cas incomplets ou non cohérents, assurant ainsi la qualité optimale de la base.

E. Considérations éthiques et confidentialité

Afin de protéger la vie privée des patients, toutes les informations personnelles sensibles ont été supprimées de la base de données. Un engagement total envers l'éthique médicale et les normes légales en matière de collecte et d'utilisation des données a été respecté, garantissant la confidentialité des informations et le respect des droits des patients.

II.3.2.5) Nettoyage et préparation des données

A. Opérations de nettoyage et vérification des valeurs

Cette étape comprend plusieurs opérations essentielles visant à garantir la qualité des données et leur adéquation à l'analyse :

- **Gestion des valeurs manquantes :**
  Les données sont scrupuleusement examinées pour détecter toute valeur manquante ou incomplète. Les enregistrements contenant de telles valeurs sont supprimés afin d'assurer la précision du modèle d'apprentissage.
- **Correction des données non logiques :**
  Des règles logiques sont appliquées pour vérifier que les valeurs saisies sont conformes aux plages attendues, telles que l'âge supérieur à zéro, et des tailles et poids dans des limites normales, ainsi que la cohérence temporelle des données médicales.
- **Formatage des données :**
  Uniformisation des unités et des mesures dans les colonnes (par exemple, taille en centimètres, poids en kilogrammes), réorganisation des colonnes et suppression des doublons.

Exemple de code pour les opérations de nettoyage :

```python
import pandas as pd

# Chargement des données
df = pd.read_csv('chemin/vers/mediAi.csv')

# Suppression des enregistrements avec valeurs manquantes
df = df.dropna()

# Vérification des valeurs logiques
df = df[(df['Âge'] > 0) & (df['Taille (cm)'] > 20) & (df['Poids (kg)'] > 3)]

# Uniformisation des unités ou autres formats à appliquer ici

# Suppression des doublons
df = df.drop_duplicates()

# Sauvegarde des données nettoyées
df.to_csv('chemin/vers/donnees_nettoyees.csv', index=False)
```

Figure 11 *Code pour les opérations de nettoyage.*

B. Encodage des données

La conversion des données textuelles et catégorielles en données numériques constitue une étape essentielle dans la préparation des données pour l'entraînement des modèles d'apprentissage automatique. En effet, la plupart des algorithmes de machine learning ne peuvent traiter que des données numériques, d'où la nécessité d'utiliser différentes méthodes d'encodage adaptées à la nature des variables.

Résumé des principales méthodes d'encodage :

- **Encodage binaire (Label Encoding)** : utilisé pour les variables dichotomiques (exemple : sexe, fumeur) en leur assignant les valeurs 0 et 1.
- **Encodage one-hot (One-Hot Encoding) :** utilisé pour les variables catégorielles ayant plusieurs modalités sans ordre naturel (exemple : groupe sanguin, état civil). Cette méthode crée une colonne binaire distincte pour chaque modalité.
- **Encodage ordinal (Ordinal Encoding)** : appliqué aux variables catégorielles ayant un ordre naturel (exemple : intensité des symptômes : légère, moyenne, sévère) en leur attribuant des codes numériques ordonnés.
- **Encodage multi-étiquette (Multi-hot Encoding)** : utilisé pour les variables textuelles contenant plusieurs valeurs possibles simultanément (exemple : liste

des maladies chroniques, symptômes actuels). Chaque modalité devient une colonne binaire indiquant sa présence ou son absence.

- **Encodage de la variable de diagnostic**

  Dans cette base, la variable Diagnostic est un label unique par observation, représentant la maladie finale diagnostiquée. Elle sera encodée en tant que variable catégorielle à une seule étiquette (single-label classification) à l'aide d'un encodage numérique simple (Label Encoding) attribuant un code numérique unique à chaque maladie.

Chaque méthode d'encodage est choisie en fonction de la nature des données pour préserver l'intégrité des informations et assurer une interprétation correcte par les algorithmes. Par exemple, l'encodage one-hot évite d'introduire un ordre implicite entre les catégories, tandis que l'encodage ordinal préserve la hiérarchie naturelle des modalités. Le multi-hot encoding permet de représenter les variables multi-étiquettes, essentielles pour capturer la présence simultanée de plusieurs caractéristiques.

Le tableau suivant résume les types de variables présentes dans la base de données, les colonnes concernées ainsi que la méthode d'encodage recommandée pour chacune, afin d'assurer une préparation optimale des données pour l'entraînement des modèles.

Type de variable	Colonnes incluses	Méthode d'encodage recommandée
Variables quantitatives	Âge, Taille (cm), Poids (kg), IMC, Cigarettes/Jour, Sommeil (heures), Durée Apparition Symptômes (jours)	Utilisation directe, normalisation possible
Variables binaires (dichotomiques)	Sexe, Enceinte, Fumeur, Prend Médicaments, Alcool, Sport	Encodage binaire (Label Encoding : 0 ou 1)
Variables catégorielles multi-étiquettes (multi-label)	Maladies Chroniques, Liste Médicaments, Allergies Médicamenteuses, Antécédents Chirurgicaux, Dispositif Médical Implanté, Symptômes Récurrents, Symptômes Actuels, Traitement Prescrit, Classe Médicaments, Antécédents Familiaux	Multi-hot encoding (colonnes binaires multiples)
Variables catégorielles nominales	Groupe Sanguin, Statut Matrimonial, Profession	One-Hot Encoding (colonnes binaires par catégorie)
Variables ordinales	Intensité Symptômes	Encodage ordinal (Label Encoding ordonné)
Variable cible (diagnostic unique)	Diagnostic	Label Encoding (encodage numérique simple)
Variables textuelles (optionnelles)	Conseils Médicaux, Méthode de traitement	Texte brut ou encodage simplifié selon besoin

Tableau 2 *Méthodes d'Encodage des Variables selon leur Type.*

Encodage des données avec Python et scikit-learn

L'encodage est essentiel pour convertir les variables catégorielles en valeurs numériques exploitables par les algorithmes d'apprentissage automatique. La bibliothèque scikit-learn propose plusieurs outils efficaces pour ce traitement.

Importation des bibliothèques nécessaires :

```python
import pandas as pd
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder
```

Figure 12 Importation des bibliothèques nécessaires.

1. Encodage binaire (Label Encoding)

Pour les variables dichotomiques (exemple : Sexe, Fumeur, Enceinte, Prend Médicaments).

```python
le = LabelEncoder()

# Encodage du sexe : "Homme" -> 0, "Femme" -> 1
df['Sexe_encoded'] = le.fit_transform(df['Sexe'])

# Encodage du tabagisme : "Non" -> 0, "Oui" -> 1
df['Fumeur_encoded'] = le.fit_transform(df['Fumeur'])

# Encodage de la grossesse (uniquement pour les femmes) : "Non" -> 0, "Oui" -> 1
df['Enceinte_encoded'] = le.fit_transform(df['Enceinte'])

# Encodage de la prise régulière de médicaments : "Non" -> 0, "Oui" -> 1
df['Prend_Medicaments_encoded'] = le.fit_transform(df['Prend Médicaments'])
```

Figure 13 Variables dichotomiques.

2. Encodage nominal (One-Hot Encoding) Pour les variables catégorielles sans ordre (exemple : Groupe Sanguin, Statut Matrimonial, Profession)

```
# Transformation des colonnes catégorielles en colonnes binaires (one-hot)
df = pd.get_dummies(df, columns=['Groupe Sanguin', 'Statut Matrimonial', 'Profession'], drop_first=False)

```

Figure 14 *Variables catégorielles sans ordre.*

3. Encodage ordinal:

Pour les variables catégorielles ordonnées (exemple : Intensité des symptômes)

```
ordinal_enc = OrdinalEncoder(categories=[['Légère', 'Moyenne', 'Sévère']])

# Transformation de la colonne 'Intensité Symptômes' en valeurs numériques ordonnées
df['Intensite_encoded'] = ordinal_enc.fit_transform(df[['Intensité Symptômes']])


```

Figure 15 *Variables catégorielles ordonnées.*

4. Encodage multi-étiquette (Multi-hot Encoding)

Pour les colonnes contenant plusieurs valeurs simultanées (exemple : Maladies Chroniques, Liste Médicaments)

```python
# Remplissage des valeurs manquantes avec une chaîne vide
df['Maladies Chroniques'] = df['Maladies Chroniques'].fillna('')

# Extraction de la liste unique des maladies chroniques dans la base
all_conditions = set()
df['Maladies Chroniques'].str.split(',').apply(all_conditions.update)

# Création d'une colonne binaire pour chaque maladie indiquant sa présence (1) ou absence (0)
for condition in all_conditions:
    condition = condition.strip()
    df['Maladie_' + condition] = df['Maladies Chroniques'].apply(lambda x: int(condition in x))

# Suppression de la colonne originale
df.drop(columns=['Maladies Chroniques'], inplace=True)

```

Figure 16 *Colonnes contenant plusieurs valeurs simultanées.*

5. Encodage de la variable cible (Diagnostic)

```python
# Réutilisation de LabelEncoder pour la colonne 'Diagnostic'
df['Diagnostic_encoded'] = le.fit_transform(df['Diagnostic'])

```

Figure 17 *Encodage de la variable cible.*

Classification à étiquette unique, chaque diagnostic est converti en un code numérique unique.

6.Exemple de codage : tableau des diagnostics avec leurs codes attribués:

Diagnostic	Diagnostic_encoded
Arthrose	5
Diabète de type 2	27
Anémie ferriprive	2

Sinusite	83
Asthme allergique	7
Hypertension artérielle	34
Rage	80
Gingivite	32
Hépatite A	37
Céphalée de tension	21
Pancréatite	70
Laryngite	44
Dermatite de contact	23
Pharyngite	75
Maladie de Parkinson	53
Troubles anxieux diagnostiqués	90
Cancer du poumon	14
Thalassémie	87
Pancréatite aiguë	71
Arthrite septique	4
Maladie de Crohn	52
COVID-19	11
Sinusite chronique	84
Lèpre	49
Asthme bronchique	8
Maladie fongique profonde	54
Tonsillite	88
Cancer du côlon	13
Caries dentaires	15
Neurodermatitis	59
Stéatose hépatique	85
Sclérodermie	82
Eczéma diagnostiqué	31
Pemphigus	74
Conjonctivite	19
Néphrite aiguë	62
Vascularite des petits vaisseaux	95
Cystite diagnostiquée	20
Inflammation oreille interne	43
Paludisme	69
Stéatose hépatique non alcoolique	86
Vascularite des vaisseaux moyens	96
Hypothyroïdie	36
Otite externe	67
Névralgie du trijumeau	63
Neuropathie	60
Charcot-Marie-Tooth	16
Pneumonie	77
Polyarthrite rhumatoïde	78
Colite ulcéreuse	18
Lichen Planus	46
Asthme	6
Diabète gestationnel	28

Neurodermatite	58
Tuberculose	91
Otite	66
AVC	0
Diarrhée bactérienne	29
Hépatite C	39
Calculs rénaux	12
Arthrite psoriasique	3
Grippe	33
Maladie inflammatoire de l'intestin	55
Pharyngite virale	76
Tuberculose systémique	92
Pancréatite kystique	73
Névrite optique	64
Néphrite	61
Dermatophytose	25
Hépatite auto-immune	40
Borréliose	9
Dermatite séborrhéique	24
Migraine	56
Colite	17
Hyperthyroïdie	35
Immunodéficience	41
Infection sexuellement transmissible	42
Oreillons	65
Tumeur cutanée	93
Maladie d'Alzheimer	51
Acné diagnostiquée	1
Diabète de type 1	26
Dacryocystite	22
Psoriasis	79
Hépatite B	38
Toxoplasmose	89
Dépression diagnostiquée	30
Brucellose	10
Pancréatite chronique	72
Rougeole	81
Mononucléose	57
Vascularite des gros vaisseaux	94
Leishmaniose	45
Lupus	47
Maladie cardiaque ischémique	50
Épilepsie	97
Lymphadénite	48
Otite moyenne	68

Tableau 3 *Diagnostics et leurs codes numériques.*

II.3.2.6) Analyse descriptive des données

Cette section vise à fournir un aperçu global des caractéristiques de la base de données utilisée, à travers des résumés statistiques et des graphiques facilitant la compréhension des distributions des variables clés et leurs relations potentielles.

A. Résumé statistique des données numériques

La base de données contient 49 000 dossiers médicaux. L'âge moyen des patients est de 40,5 ans avec un écart-type de 21,5 ans. L'histogramme ci-dessous (Figure 18) illustre la distribution des âges au sein de l'échantillon, montrant une large répartition allant de 1 à 90 ans.

La moyenne du poids est de 75,7 kilogrammes, la taille moyenne est de 164,1 centimètres, tandis que l'indice de masse corporelle (IMC) reflète la diversité des états nutritionnels parmi les patients.

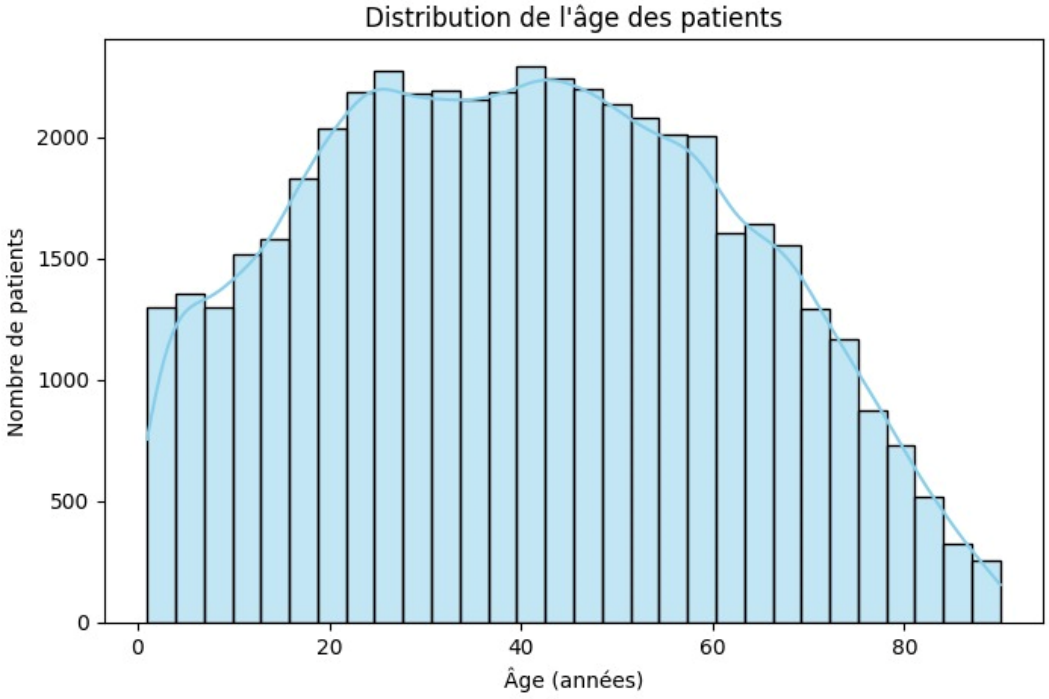


Figure 18 *Résumé statistique des âges des patients.*

B. Répartition des variables catégorielles

La proportion de femmes est de 49,8 %, tandis que celle des hommes est de 50,2 %, comme le montre le diagramme circulaire suivant :

La distribution de l'état civil révèle que la majorité des patients sont célibataires (33,5 %), suivis par les veufs (22,4 %) et les divorcés (22,1 %), tandis que les mariés représentent environ 22 % de l'échantillon, comme présenté dans le diagramme à barres ci-dessous (Figure 19) :

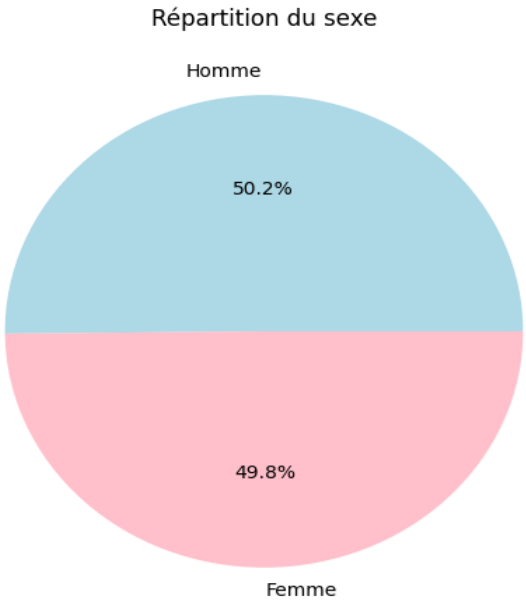


Figure 19 *Répartition des patients selon le sexe.*

Distribution de l'état civil révèle que la majorité des patients sont **Célibataire**, avec la répartition des autres catégories présentée dans le diagramme à barres ci-dessous(Figure 20) :

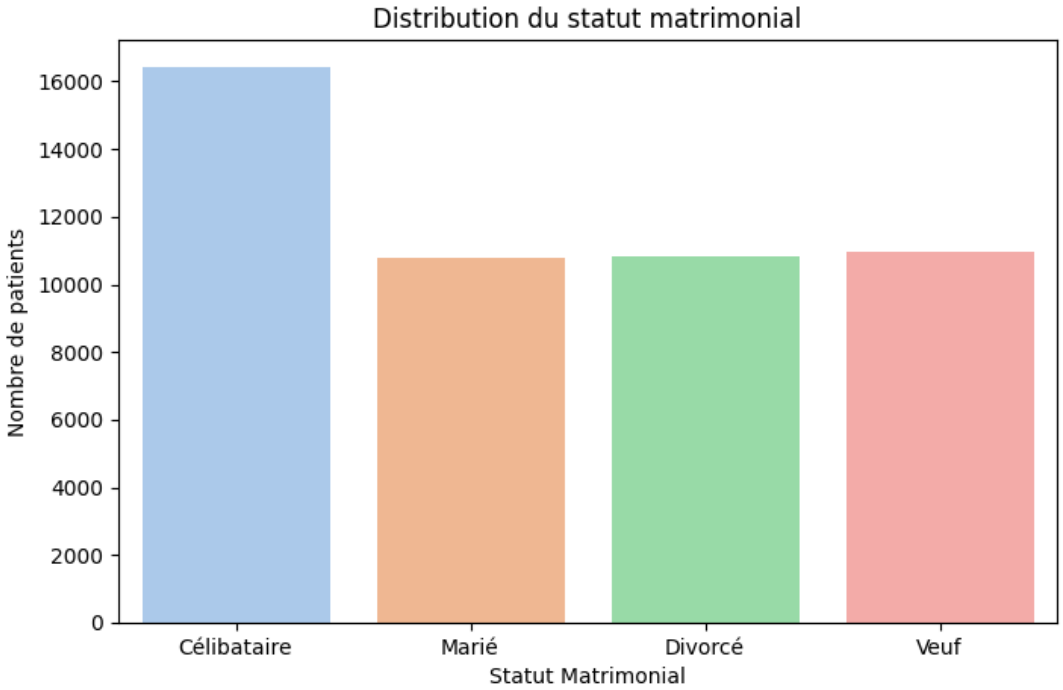


Figure 20 *Distribution du statut matrimonial des patients.*

C. Analyse des symptômes

Les symptômes les plus communément rapportés parmi les patients incluent : Fièvre, chroniques Fatigue. La répartition des intensités des symptômes est assez équilibrée avec 33,4 % des symptômes classés comme légers, 33,5 % comme modérés, et 33,1 % comme sévères, illustrée dans le graphique suivant.

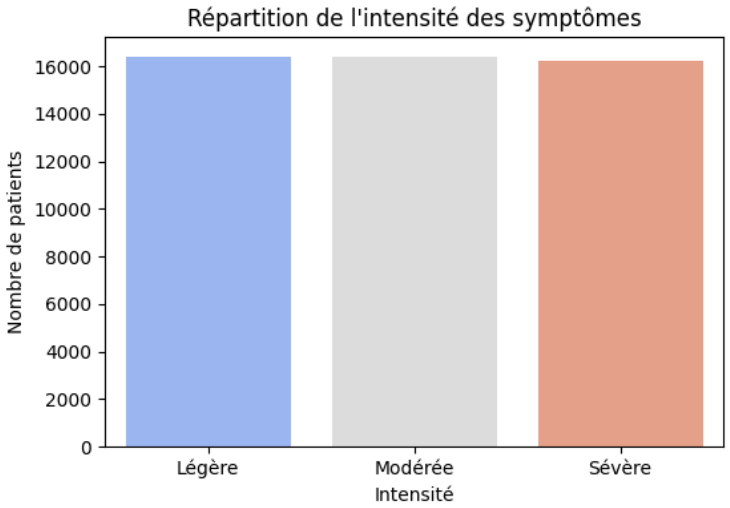


Figure 21 *Répartition de l'intensité des symptômes.*

D. Corrélations initiales entre variables

Une analyse des corrélations entre les variables numériques clés telles que l'âge, le poids, l'IMC, et la durée d'apparition des symptômes a été réalisée. La matrice de corrélation ci-dessous met en évidence des relations significatives, notamment une corrélation positive modérée entre l'âge et le poids ($r \approx 0,42$), le poids et la taille ($r \approx 0,54$), ainsi qu'entre le poids et l'IMC ($r \approx 0,78$).

Des analyses complémentaires ont été menées pour explorer les relations entre le sexe, les diagnostics, ainsi que l'impact du tabagisme sur la prévalence des maladies chroniques, apportant des informations précieuses pour la modélisation prédictive.

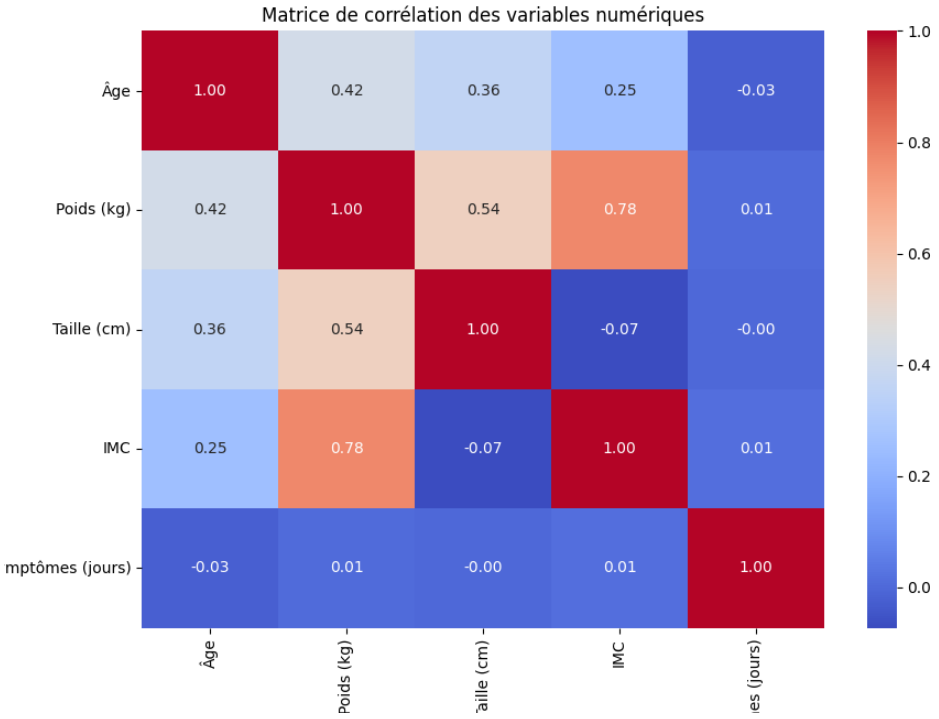


Figure 22 *Corrélations initiales entre variables.*

II.4) Conclusion

Dans ce chapitre, nous avons exploré trois méthodes de classification couramment utilisées dans le domaine médical : SVM, XGBoost et Random Forest, en vue d'identifier celle qui serait la plus performante pour notre application de diagnostic. Nous avons également présenté deux bases de données de référence issues du UCI Machine Learning Repository : le Heart Disease Dataset et le Breast Cancer Wisconsin Dataset, qui serviront à l'évaluation des modèles. Par ailleurs, nous avons détaillé la construction de notre propre base de données médicale, ainsi que les différentes étapes de nettoyage, de traitement et de préparation nécessaires à une analyse fiable. Ces éléments constituent les fondations nécessaires pour passer, dans le chapitre suivant, à l'expérimentation, l'évaluation comparative et l'interprétation des résultats obtenus.

Chapitre III. Entraînement des modèles et analyse des résultats

III.1) Introduction

Dans ce chapitre, nous passons à la phase d'implémentation pratique des modèles de classification étudiés. À partir des bases de données préparées dans le chapitre précédent, nous allons appliquer trois méthodes d'apprentissage automatique : **XGBoost**, **Random Forest** et **SVM**.

Ces modèles seront d'abord testés sur les bases **Heart Disease** et **Breast Cancer**, déjà nettoyées et prêtes à l'emploi. L'objectif est d'évaluer leurs performances. Ensuite, ces mêmes algorithmes seront testés sur notre propre base de données créée pour ce projet, afin de vérifier leur efficacité dans un cas reel et d'identifier la méthode la plus adaptée à nos données.

Nous ferons aussi attention à deux problèmes fréquents : le surapprentissage et le déséquilibre des données.

Enfin, la méthode jugée la plus performante sera intégrée dans une **interface mobile conviviale**, pensée pour une utilisation simple et intuitive par l'utilisateur final, notamment les patients.

III.2) Analyse des classificateurs

Dans cette section, nous appliquerons les trois méthodes de classification sur deux bases de données : **Breast Cancer** et **Heart Disease,** afin d'analyser la performance de chaque méthode sur des jeux de données de nature différente, à savoir binaire (biclasse) pour la première et multi classe pour la seconde.

III.2.1) Application et réglages des modèles

Dans cette section, nous présentons les paramètres principaux utilisés pour l'application des trois modèles sélectionnés dans cette étude : **XGBoost, Random Forest** et **SVM**. Ces réglages ont été choisis pour optimiser la performance des modèles sur les différentes bases de données étudiées.

Résumé des paramètres que nous avons utilisés selon la base de données (UCI bases)

Modèle	Breast Cancer Wisconsin	Heart Disease Dataset
XGBoost	n_estimators=50, max_depth=3	n_estimators=50, max_depth=3
Random Forest	n_estimators=100, poids équilibré non utilisé	n_estimators=100, class_weight='balanced'
SVM	Recherche Grid Search sur C et kernel	class_weight='balanced', C=1, noyau par défaut

Tableau 4 *:Synthèse des réglages par base de données*

Cette configuration garantit que chaque modèle est entraîné avec des paramètres adaptés à la nature de la base de données et aux exigences de classification, maximisant ainsi la performance et la robustesse des modèles.

III.2.2) Implémentation des modèles et analyse des résultats

Cette partie présente les résultats obtenus à l'issue de l'entraînement des trois modèles d'apprentissage automatique sélectionnés — **SVM, Random Forest** et **XGBoost** — sur différentes bases de données médicales, ainsi qu'une analyse comparative approfondie de leurs performances.

III.2.2.1) Entraînement et analyse des résultats sur la base Breast Cancer (classification binaire)

La base de données **Breast Cancer Wisconsin** constitue un problème de classification binaire, distinguant entre tumeurs bénignes et malignes. Les modèles ont été entraînés en utilisant des techniques d'optimisation des hyper paramètres afin de maximiser leurs performances.

Les étapes suivantes montrent le déroulement et le paramétrage des différentes méthodes utilisées compris le prétraitement des données de Breast cancer

1. **Importation des bibliothèques** : Importe SVC, RandomForestClassifier, XGBClassifier, GridSearchCV, StandardScaler, et autres.
2. **Chargement des données**
3. **Prétraitement des données** : Supprime les valeurs manquantes, sépare X et y, divise en train/test.
4. **Mise à l'échelle** : Normalise avec StandardScaler().

5. **Entraînement XGBoost** : Configure avec n_estimators=50, max_depth=3, entraîne. 6. **Prédiction XGBoost** : Prédit sur X_test. 7. **Évaluation XGBoost** .	5. **Entraînement Random Forest** : Configure avec random_state=42, entraîne. 6. **Prédiction Random Forest** : Prédit sur X_test. **7. Évaluation Random Forest .**	5. **Optimisation SVM** : Utilise GridSearchCV avec kernel=['rbf', 'poly'], gamma=['scale', 'auto']. 6. **Entraînement et prédiction SVM** : Entraîne et prédit sur X_test_scaled. 7. **Évaluation SVM** .

Le tableau suivant montre les résultats obtenus après l'application des trois méthodes sur la base **Breast Cancer**

Analyse des performances

Modèle	Précision (Accuracy)	Précision moyenne (Precision)	Rappel moyen (Recall)	F1-score moyen
SVM	98,24 %	98 %	98 %	98 %
Random Forest	96,49 %	97 %	96 %	96 %
XGBoost	96,49 %	96 %	96 %	96 %

Tableau 5 *Analyse des performances sur la base Breast Cancer*

- Le modèle SVM se distingue par sa précision globale supérieure et un excellent équilibre entre précision et rappel.
- Les modèles **Random Forest** et **XGBoost** présentent des performances très compétitives, avec une robustesse notable.

III.2.2.2) Entraînement et analyse des résultats sur la base Heart Disease (classification multi-classes)

La base de données **Heart Disease Dataset** correspond à un problème de classification multi-classes avec cinq catégories, ce qui représente un défi supérieur pour les modèles.

Les étapes suivant montrent le déroulement et le paramétrage des différentes méthodes utilisées compris le prétraitement des données de **la base Heart Disease**

1. **Importation des bibliothèques** : Importe SVC, RandomForestClassifier, XGBClassifier, accuracy_score, et classification_report.
2. **Chargement des données**
3. **Prétraitement des données** Sépare X et y, divise en train/test avec train_test_split(X, y, test_size=0.3, random_state=42).
4. **Mise à l'échelle Normalise** avec StandardScaler() sur X_train et X_test.

5. **Entraînement du SVM** Configure avec kernel='rbf', class_weight='balanced', entraîne.
6. **Prédiction du SVM** Prédit sur X_test_scaled.
7. **Évaluation du SVM** .

5. **Entraînement du Random Forest** Configure avec n_estimators=100, random_state=42, entraîne.
6. **Prédiction du Random Forest**
7. **Évaluation du Random Forest**

5. **Entraînement de XGBoost** Configure avec use_label_encoder=False, eval_metric='logloss', n_estimators=50, max_depth=3, entraîne.
6. **Prédiction de XGBoost**
7. **Évaluation de XGBoost**

8. **Affichage des résultats** : Affiche les performances pour chaque modèle.

Les résultats de Précision, Précision moyenne, Rappel moyen et F1-score moyen de l'application des trois algorithmes cités précédemment sur la base Heart disease ont expérimenté dans le tableau suivant

Analyse des performances

Modèle	Précision (Accuracy)	Précision moyenne (Precision)	Rappel moyen (Recall)	F1-score moyen
SVM	80,48 %	81 %	81 %	80 %
Random Forest	87,19 %	88 %	87 %	87 %
XGBoost	85,97 %	87 %	86 %	86 %

Tableau 6 *table d'Analyse des performances heart*

- Le Random Forest obtient la meilleure performance globale, montrant une capacité de généralisation élevée dans un contexte multi-classes.
- XGBoost démontre également une forte capacité d'apprentissage sur cette tâche complexe.
- SVM souffre d'une baisse de performance significative, en raison de ses limitations dans la gestion des problèmes multi-classes.

III.2.2.3) Comparaison des algorithmes et conclusions

- Pour la classification binaire (Breast Cancer), SVM est le modèle le plus performant, grâce à sa capacité à maximiser la marge entre deux classes.

- Pour la classification multi-classes (Heart Disease), **Random Forest** et **XGBoost** surpassent nettement **SVM**, offrant plus de flexibilité et une meilleure adaptation aux données complexes.

- Étant donné que la base de données spécifique au projet contient **98 classes** (maladies), il est recommandé d'écarter l'utilisation de SVM pour la phase suivante, et de privilégier **Random Forest** et **XGBoost** qui sont plus adaptés aux tâches multi-classes avec un grand nombre de catégori

III.3) L'utilisation de la base développée

III.3.1) Entraînement des modèles sur la base de données spécifique au projet

Dans le chapitre précédent, toutes les opérations de nettoyage, de préparation et de codage de la base de données spécifique au projet ont été complétées de manière rigoureuse. Ces étapes ont permis de traiter les valeurs manquantes, de corriger les anomalies et de convertir les variables catégorielles en formats numériques adaptés, rendant ainsi les données prêtes à être utilisées directement pour l'entraînement des modèles.

Dans ce chapitre, nous entraînons les modèles sélectionnés sur cette base de données préparée, en utilisant dans un premier temps les paramètres par défaut fournis par les bibliothèques d'apprentissage automatique. Cette approche permet d'obtenir une évaluation initiale des performances des modèles avant d'envisager toute optimisation ou réglage des hyperparamètres.

III.3.1.1) Entraînement des modèles sélectionnés

- Chargement des données

Les données prétraitées et codées sont chargées à partir d'un fichier CSV, prêtes pour l'entraînement :

```python
import pandas as pd

data_prepared = pd.read_csv('prepared_project_database.csv')

X = data_prepared.drop(columns=['Diagnostic_encoded'])
y = data_prepared['Diagnostic_encoded']

```

Figure 23 *Chargement des données*

-Séparation des données en ensembles d'entraînement et de test

Les données sont divisées en deux ensembles : 80% pour l'entraînement et 20% pour le test, en respectant la répartition des classes grâce à un échantillonnage stratifié :

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

Figure 24 *Séparation des données en ensembles d'entraînement et de test*

-Entraînement des modèles avec paramètres par défaut

Les modèles **Random Forest** et **XGBoost** sont entraînés en utilisant leurs réglages par défaut, offrant ainsi une première estimation de leurs performances sur la base de données spécifique.

- Entraînement du modèle **Random Forest** :

```python
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(random_state=42, class_weight='balanced')
rf_model.fit(X_train, y_train)

```

Figure 25 *Entraînement du modèle Random Forest*

Entraînement du modèle **XGBoost** :

```python
from xgboost import XGBClassifier

xgb_model = XGBClassifier(
    use_label_encoder=False, eval_metric='logloss', random_state=42)
xgb_model.fit(X_train, y_train)

```

Figure 26 *Entraînement du modèle XGBoost*

Après cette phase d'entraînement initiale avec les paramètres par défaut, l'évaluation des performances des modèles et l'analyse des résultats seront présentées dans la

section suivante, avec la possibilité de procéder à des optimisations ultérieures pour améliorer les résultats.

III.3.1.2) Évaluation initiale des modèles avec paramètres par défaut

A. Résultats de l'évaluation initiale

Les résultats obtenus sur l'ensemble de test montrent des indicateurs de performance parfaits, les deux modèles ayant atteint une précision (Accuracy) de 100 %, ainsi que des valeurs maximales pour la précision (Precision), le rappel (Recall) et le score F1 (F1-score) pour toutes les classes, comme illustré dans le tableau suivant :

Modèle	Accuracy	Precision	Recall	F1-score
Random Forest	1.00	1.00	1.00	1.00
XGBoost	1.00	1.00	1.00	1.00

Tableau 7 Résultats de l'évaluation initiale

B. Discussion des résultats

Ces résultats témoignent de la capacité des modèles à distinguer avec une grande précision les différentes classes de maladies, ce qui est en partie attendu compte tenu de l'équilibre des données entre les classes. La présence d'un nombre équivalent d'échantillons (environ 500 par classe) réduit le biais lié aux classes sous-représentées et permet aux modèles d'apprendre de manière équitable et homogène pour chaque catégorie, facilitant ainsi l'atteinte de performances de classification excellentes.

C. Étapes futures

Il est essentiel de vérifier l'absence de surapprentissage (overfitting) du modèle, car une grande précision sur l'ensemble de test ne garantit pas nécessairement une bonne capacité de généralisation. Le surapprentissage peut survenir lorsque le modèle mémorise des détails spécifiques au jeu d'entraînement, ce qui peut nuire à sa performance sur de nouvelles données. Cette étape de vérification est cruciale pour assurer la fiabilité du modèle et sa capacité à fonctionner de manière optimale.

III.3.1.3) Vérification du surapprentissage (Overfitting)

La vérification du surapprentissage est une étape essentielle dans la construction d'un modèle performant et fiable. Elle survient lorsque le modèle s'adapte excessivement aux détails spécifiques des données d'entraînement, compromettant ainsi sa capacité à généraliser sur de nouvelles données. Dans cette section, nous avons appliqué des méthodes telles que la validation croisée et l'analyse des courbes d'apprentissage pour vérifier la présence de surapprentissage dans les modèles Forêt Aléatoire (Random Forest) et XGBoost.

A. Importance de la Vérification du Surapprentissage

Il est crucial de souligner qu'une précision parfaite sur l'ensemble de test ne garantit pas nécessairement la capacité du modèle à se généraliser efficacement à de nouvelles données. Le surapprentissage peut se produire, réduisant ainsi la performance du modèle sur des données non vues. Par conséquent, la vérification de l'absence de surapprentissage est une étape incontournable pour assurer la fiabilité et la capacité de généralisation du modèle, en particulier dans des domaines sensibles tels que les applications médicales, où une haute précision diagnostique est indispensable.

B. Méthodes de vérification du surapprentissage

Évaluation par validation croisée (Cross-validation)

La validation croisée est une méthode courante pour évaluer les performances des modèles en apprentissage automatique. Les données sont divisées en un nombre défini de plis (dans ce cas, 5 plis, soit k=5) pour garantir une évaluation stable et fiable. Le processus se déroule comme suit :

1. **Division des données** : Les données sont divisées en 5 parties approximativement égales.
2. **Entraînement et test** : 4 parties sont utilisées pour entraîner le modèle, tandis que la cinquième est utilisée pour le test. Ce processus est répété 5 fois, chaque partie ayant son tour en tant que groupe de test.
3. **Calcul de la précision** : La précision moyenne (accuracy) est calculée sur l'ensemble des plis pour obtenir une évaluation globale du modèle.
4. **Analyse de la variance** : L'écart-type (standard deviation) est également calculé pour comprendre la stabilité des performances.

Analyse des courbes d'apprentissage (Learning Curves)

Les courbes d'apprentissage permettent d'évaluer la performance d'un modèle en fonction de la taille des données d'entraînement, en comparant les résultats sur les ensembles d'entraînement et de test. Voici comment cela fonctionne :

1. **Génération des tailles d'entraînement** : Les données sont divisées en différentes tailles croissantes (par exemple, de 0,1 à 1 avec 10 intervalles).
2. **Évaluation par validation croisée** : Pour chaque taille, le modèle est entraîné et testé à l'aide d'une validation croisée (ici, avec 5 plis, cv=5), et les scores de précision sont calculés.
3. **Calcul des moyennes et écarts-types** : Les scores moyens et les écarts-types sont calculés pour les ensembles d'entraînement et de test afin de mesurer la performance et la stabilité.

4. **Visualisation** : Les résultats sont tracés sous forme de courbes, où la précision d'entraînement (généralement plus élevée) est comparée à la précision de validation. Un grand écart entre ces courbes peut indiquer un surapprentissage.

C. Résultats de la vérification

- **Résultats de la validation croisée pour le modèle Forêt Aléatoire (Random Forest)**

La validation croisée a été effectuée pour le modèle Forêt Aléatoire avec 5 divisions, et les résultats ont montré une précision parfaite sur toutes les divisions.

Les résultats de la validation croisée sont les suivants :

Précision de la validation croisée	**[1.0, 1.0, 1.0, 1.0, 1.0]**
Précision moyenne	**1.0**
Écart-type	**0.0000**

Tableau 8 *résultats de la validation croiseé pour Random Forest*

Cela indique que le modèle atteint une précision parfaite sur toutes les données divisées, sans aucune variation ou instabilité dans la performance.

- **Résultats de la validation croisée pour le modèle XGBoost**

Pour le modèle XGBoost, les résultats de la validation croisée ont également montré une précision parfaite, avec un léger écart dans une des expériences. Les résultats sont les suivants :

Précision de la validation croisée	**[1.0, 1.0, 1.0, 0.9999, 1.0]**
Précision moyenne	**0.999999**
Écart-type	**0.0000**

Tableau 9 *résultats de la validation croiseé pour XGBoost*

XGBoost montre également une performance stable, mais avec une légère variation dans une expérience, ce qui signifie que la performance globale est toujours bonne mais légèrement inférieure à une précision parfaite.

- **Courbe d'apprentissage pour le modèle Forêt Aléatoire (Random Forest)**

En analysant la courbe d'apprentissage pour le modèle Forêt Aléatoire, il est apparu qu'il y a une cohérence entre la précision d'entraînement et celle de validation à mesure que le nombre d'exemples d'entraînement augmente. Cela indique que le modèle ne souffre pas de surapprentissage majeur.

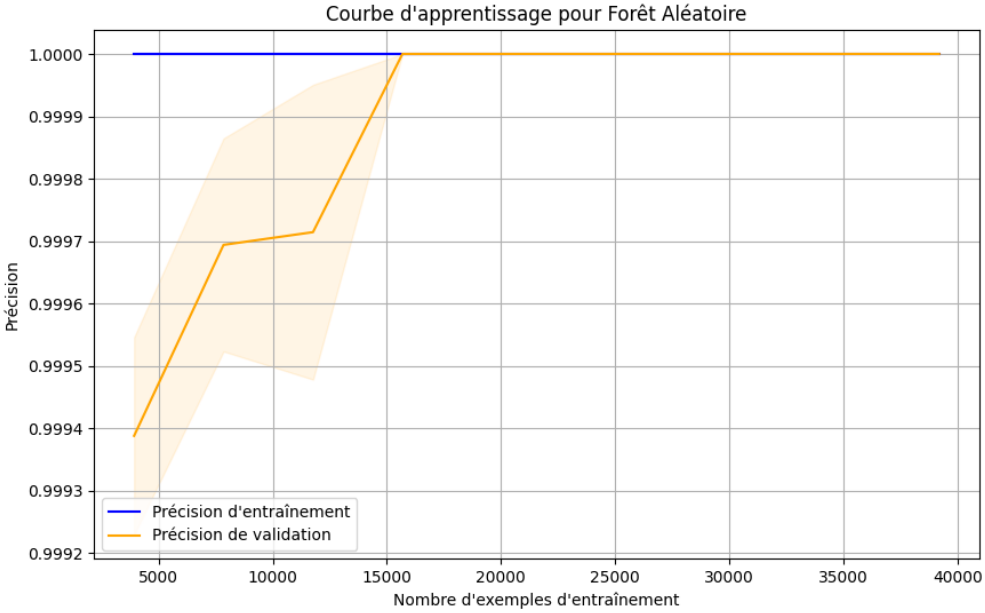


Figure 27 Courbe d'apprentissage pour le modèle Forêt Aléatoire

- **Courbe d'apprentissage pour le modèle XGBoost**

Concernant la courbe d'apprentissage pour XGBoost, les résultats montrent également une cohérence entre la précision d'entraînement et celle de validation. Lorsque le nombre d'exemples d'entraînement augmente, il n'y a pas de grande différence entre les performances sur les données d'entraînement et les données de test.

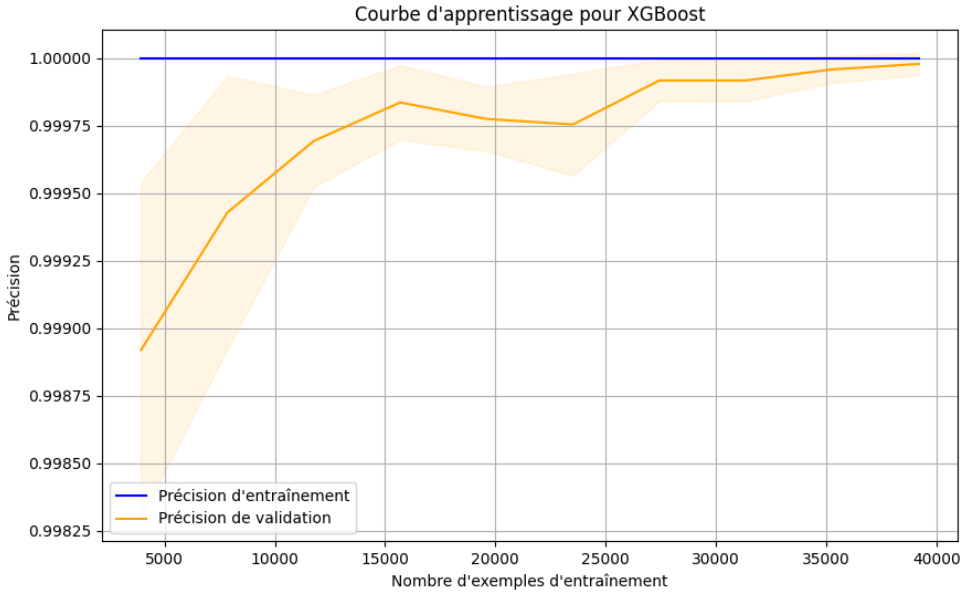


Figure 28 Courbe d'apprentissage pour le modèle XGBoost

D. Analyse des résultats

‘En se basant sur les résultats de la validation croisée et des courbes d'apprentissage, nous pouvons conclure qu'il n'y a pas de signe clair de **surapprentissage** dans les deux modèles. Les modèles montrent une haute précision constante sur toutes les divisions et sur les données de test.

E. Étapes futures

Bien que les deux modèles présentent une haute précision sans signe évident de surapprentissage, les données équilibrées utilisées dans cette étude peuvent ne pas représenter fidèlement les données réelles où les classes sont souvent déséquilibrées. **L'étape suivante** consiste à adapter l'entraînement pour traiter les données déséquilibrées, en utilisant des techniques telles que **SMOTE** (Synthetic Minority Over-sampling Technique) ou des ajustements de poids pour compenser le biais des classes majoritaires.

III.3.1.4) Entraînement du modèle sur des données déséquilibrées

Dans de nombreuses applications réelles, y compris dans les applications médicales, les données sont souvent **déséquilibrées**. Dans ce cas, certaines catégories sont plus dominantes que d'autres, ce qui amène les modèles à préférer les catégories majoritaires. Cela peut entraîner une réduction de la précision du modèle pour les catégories moins représentées. Ce défi était présent dans les données utilisées pour entraîner notre modèle, ce qui a rendu nécessaire l'atteinte de l'équilibre pendant l'entraînement. Ainsi, en cas de **réentraînement du modèle sur des données déséquilibrées**, le modèle sera toujours capable de maintenir une bonne performance sans négliger les catégories moins représentées.

A. Techniques pour traiter les données déséquilibrées

- **Technique SMOTE (Synthetic Minority Over-sampling Technique)**

SMOTE est une méthode utilisée pour équilibrer les ensembles de données en augmentant le nombre d'échantillons des classes sous-représentées. Voici comment elle fonctionne :

1. **Identification des minorités :** SMOTE identifie les échantillons appartenant à la classe minoritaire.
2. **Génération d'échantillons synthétiques :** Pour chaque échantillon de la classe minoritaire, SMOTE sélectionne un ou plusieurs de ses voisins les plus proches et crée de nouveaux points de données en interpolant entre eux.
3. **Application :** Cette interpolation est réalisée en utilisant une combinaison linéaire aléatoire, avec un paramètre random_state pour assurer la reproductibilité (**random_state**=42).
4. **Résultat :** Le nouvel ensemble de données contient plus d'échantillons synthétiques, ce qui aide le modèle à mieux généraliser pour les classes rares.

- **Utilisation des poids pour ajuster l'impact des classes majoritaires**

L'ajustement des poids des classes pendant l'entraînement permet de donner plus d'importance aux classes sous-représentées, améliorant ainsi la performance des modèles sur celles-ci. Voici comment cela fonctionne :

1. **Définition des poids** : Des algorithmes comme RandomForestClassifier et XGBClassifier permettent de spécifier un paramètre class_weight='balanced' ou scale_pos_weight pour pondérer automatiquement les classes en fonction de leur fréquence.
2. **Application** : Pour RandomForest, le paramètre class_weight='balanced' ajuste les poids inversement proportionnels à la fréquence des classes. Pour XGBoost, scale_pos_weight peut être ajusté manuellement ou calculé (par exemple, en utilisant le rapport entre le nombre d'échantillons des classes majoritaires et minoritaires).
3. **Entraînement** : Le modèle est entraîné sur les données équilibrées, ce qui réduit le biais envers les classes majoritaires.
4. **Résultat** : Cette approche aide le modèle à mieux généraliser en tenant compte des classes rares, améliorant la précision globale.

B. Entraînement des modèles sur les données modifiées

Des techniques comme **SMOTE** et l'ajustement des poids ont été utilisées pour entraîner les modèles sur des données déséquilibrées, en veillant à ce que chaque classe prenne une place appropriée dans le processus d'entraînement. Nous allons maintenant entraîner les **forêts aléatoires (Random Forest)** et **XGBoost** sur les données modifiées.

```python
# Entraînement du modèle RandomForest avec les données équilibrées
model_rf = RandomForestClassifier(random_state=42, class_weight='balanced')
model_rf.fit(X_res, y_res)

# Entraînement du modèle XGBoost avec les données équilibrées
model_xgb = XGBClassifier(random_state=42, scale_pos_weight=class_weights)
model_xgb.fit(X_res, y_res)

```

Figure 29 Entraînement des modèles sur les données modifiées

C. Évaluation des modèles après modification des données

Une fois les modèles entraînés sur les données modifiées, il est nécessaire d'évaluer leur performance sur un jeu de test distinct afin de vérifier leur capacité à généraliser correctement. Cela inclut l'examen de la précision du modèle et d'autres mesures de performance comme la **précision (Precision)**, le **rappel (Recall)** et le **F1-score** pour chaque classe.

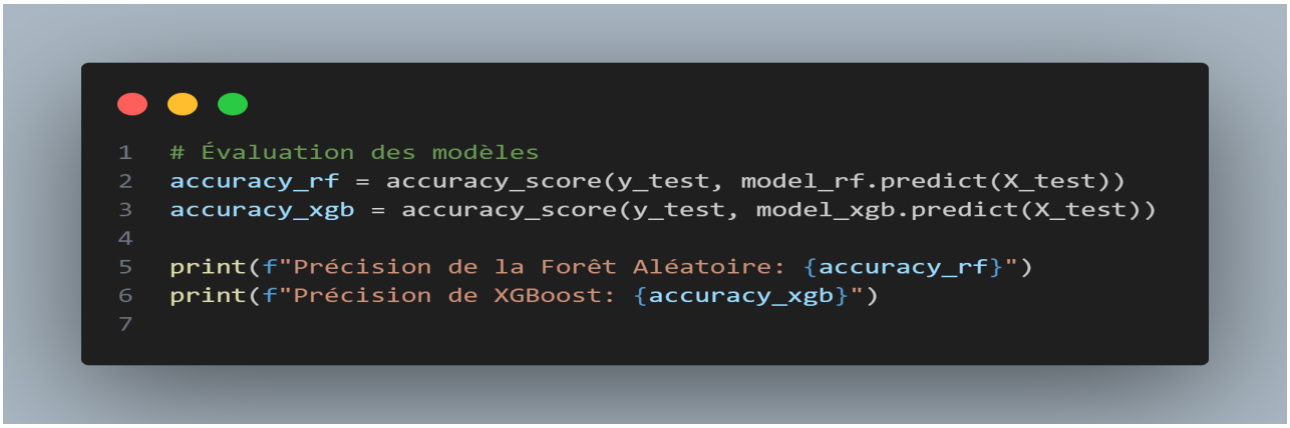


Figure 30 Évaluation des modèles après modification des données

D. Perspectives sur la Réponse du Modèle aux Données Déséquilibrées

Bien que le modèle n'ait pas encore été testé sur des données non équilibrées, il est raisonnable de s'attendre à ce qu'il réagisse efficacement lors de l'entraînement sur de telles données. En effet, la version initiale du modèle a été formée sur des données équilibrées, ce qui lui a permis d'apprendre les caractéristiques importantes des différentes catégories. Par conséquent, l'application de techniques pour traiter les déséquilibres, telles que SMOTE, devrait améliorer la capacité du modèle à gérer ces données de manière plus performante et adaptée.

E. Étapes futures

Le modèle pourrait rencontrer des difficultés à l'avenir si de **nouvelles catégories** ou **nouveaux diagnostics** apparaissent et ne sont pas inclus dans les données d'origine. Dans ce cas, il serait nécessaire de **ré-entraîner le modèle** pour inclure ces nouvelles catégories. Par exemple, le modèle actuel contient seulement **98 catégories de maladies**, et il pourrait bénéficier d'un **entraînement continu** lorsque de nouveaux diagnostics ou maladies non mentionnées apparaissent.

Le réentraînement du modèle avec de nouvelles catégories ou des données déséquilibrées sera une étape essentielle pour garantir que le modèle reste performant et à jour dans des environnements réels.

III.3.1.5) Rentraîner le modèle et ajouter de nouvelles catégories

A. Réentraîner avec de nouvelles données

Le réentraînement d'un modèle avec de nouvelles données est essentiel pour maintenir sa pertinence face à l'évolution des données, notamment l'apparition de nouvelles catégories ou diagnostics. Voici comment cela fonctionne :

1. **Chargement des nouvelles données** : Les nouvelles données sont chargées (par exemple, depuis un fichier CSV) et fusionnées avec les données existantes.
2. **Mise à jour des données** : Les ensembles X (caractéristiques) et y (cibles) sont mis à jour en combinant les anciennes et nouvelles données, puis en supprimant les colonnes inutiles si nécessaire.
3. **Division en ensembles d'entraînement et de test** : Les données combinées sont divisées en ensembles d'entraînement et de test (par exemple, avec un ratio de 80/20 et un random_state pour reproductibilité).
4. **Équilibrage des classes** : Une technique comme SMOTE est appliquée pour équilibrer les classes dans les données d'entraînement, en générant des échantillons synthétiques pour les classes minoritaires.
5. **Réentraînement du modèle** : Le modèle (par exemple, RandomForest) est réentraîné sur les données équilibrées avec des poids ajustés (class_weight='balanced'), puis évalué sur l'ensemble de test.
6. **Sauvegarde du modèle** : Une fois réentraîné, le modèle mis à jour est sauvegardé pour une utilisation future.

B. Ajouter de nouvelles catégories aux données

L'intégration de nouvelles catégories dans les données est essentielle pour maintenir la pertinence d'un modèle face à l'évolution des diagnostics. Voici comment cela fonctionne :

1. **Création des nouvelles données** : Une nouvelle catégorie (par exemple, "Nouvelle Maladie") est ajoutée sous forme de données synthétiques avec des caractéristiques (feature1, feature2) et une cible (Diagnostic_encoded).
2. **Fusion avec les données existantes** : Les nouvelles données sont combinées avec les anciennes données pour former un ensemble mis à jour.
3. **Mise à jour des ensembles X et y** : Les caractéristiques (X) et les étiquettes (y) sont extraites des données combinées, en supprimant les colonnes inutiles si nécessaire.
4. **Division en ensembles d'entraînement et de test** : Les données mises à jour sont divisées en ensembles d'entraînement et de test (par exemple, avec un ratio de 80/20 et un random_state pour reproductibilité).

5. **Équilibrage des classes** : SMOTE est appliqué pour équilibrer les classes dans les données d'entraînement en générant des échantillons synthétiques pour les nouvelles catégories.
6. **Réentraînement du modèle** : Le modèle (par exemple, XGBoost) est réentraîné sur les données équilibrées avec des poids ajustés (scale_pos_weight='balanced'), puis évalué sur l'ensemble de test.
7. **Sauvegarde du modèle** : Le modèle mis à jour est sauvegardé pour une utilisation future.

C. Évaluation après ajout de nouvelles catégories

Après l'ajout de nouvelles catégories et le réentraînement des modèles, il est nécessaire d'évaluer la performance du modèle pour vérifier qu'il fonctionne correctement avec les nouvelles données tout en maintenant la précision pour les anciennes catégories.

D. Développement Continu

- Bien que le modèle actuel montre de bonnes performances et soit capable de gérer les nouvelles catégories, **le développement ne s'arrête pas** ici.
- De nouvelles catégories ou modifications dans les données peuvent apparaître, nécessitant un réentraînement.
- **L'entraînement continu** est essentiel pour maintenir la précision du modèle et l'adapter aux évolutions futures.

III.3.1.6) Choix du modèle optimal pour notre base de données médicale

Dans les applications médicales modernes, où les volumes de données peuvent atteindre des millions de points d'information, le choix du modèle approprié est essentiel. Après avoir évalué les performances des modèles **Random Forest** et **XGBoost**, plusieurs critères ont été pris en compte pour déterminer lequel serait le mieux adapté à notre contexte, notamment la précision, le temps de formation, et la capacité à gérer de grandes quantités de données.

A. Critères de sélection du modèle

Les données médicales dans notre étude comprennent actuellement environ **49000 enregistrements**, avec des prévisions de croissance continue. Un modèle efficace pour ces données massives doit être capable de **gérer rapidement** des volumes importants tout en maintenant un haut niveau de performance.

B. Performances et temps d'entraînement

Bien que les résultats de précision entre **XGBoost et Random Forest** soient comparables, un critère décisif dans notre choix a été le **temps d'entraînement**. **XGBoost** est généralement plus rapide dans l'entraînement des modèles sur de grandes bases de données en raison de son efficacité en termes de calcul. Cette

caractéristique est particulièrement importante dans un environnement médical où le traitement rapide des données et la mise à jour fréquente des modèles sont essentiels.

Critère	XGBoost	Random Forest
Précision	100% (similaire)	100% (similaire)
Temps d'entraînement	Moins de 10 minutes	Plus de 15 minutes
Capacité à traiter de grandes quantités de données	Très performant pour les grandes bases	Moins efficace avec de très grands ensembles de données
Interprétabilité	Moins interprétable	Facile à interpréter

Tableau 10 *Comparaison des performances et du temps d'entraînement entre XGBoost et Random Forest.*

C. Choix final du modèle : XGBoost

En tenant compte de la taille des données et du besoin de traiter rapidement de grands volumes d'informations, **XGBoost** a été choisi comme modèle optimal pour cette application. Bien que **Random Forest** offre également de bonnes performances, **XGBoost** se distingue par sa **rapidité d'entraînement** et sa capacité à **gérer efficacement les données massives**, ce qui en fait le modèle le mieux adapté à un environnement en constante évolution avec des augmentations continues de données.

III.4) Développement et évaluation d'une API et d'applications mobiles pour le diagnostic médical avec XGBoost :

III.4.1) Création du fichier de modèle (.pkl)

Le modèle XGBoost, sélectionné pour ses performances, est sauvegardé en fichier .pkl via joblib pour une utilisation pratique. Les données de test (X_test, y_test) sont chargées (80 % entraînement, 20 % test), et le modèle est sauvegardé avec joblib.dump pour un accès rapide sans réentraînement.

Remarques techniques : Joblib est choisi pour son efficacité. Les données de test doivent respecter le format d'entraînement. Une compression (**ex. joblib.dump(model, 'model.pkl', compress=3)**) est recommandée pour les modèles volumineux.

III.4.2) Choix de l'environnement logiciel

Nous choisirons à ce stade les outils et environnements de programmation que nous allons utile dans le développement et l'intégration du système dans divers environnements applicatifs : ces outils et environnements sont choisis en fonction de l'intérêt qu'ils présentent pour le projet et la garantie de l'efficacité et des performances. Les outils choisis sont : Visual Studio Code/Python, Flutter, FastAPI et Firebase: une présentation de chaque outil/framework utilisé, ainsi que les raisons de leur choix, figure ci-dessous.

- **Visual Studio Code / Python**

Python est un langage de programmation puissant et facile à apprendre. Il possède des structures de données de haut niveau et gère la programmation orientée objet, ce qui le rend idéal pour le développement rapide d'applications sur diverses plateformes[34] .

Visual Studio CodeVS Code est un éditeur de code open source léger et adaptable qui prend en charge plusieurs langages, dont Python, et fournit un environnement de développement.

Raison de la sélection: VS Code a été choisi pour son intégration réussie avec Python, qui inclut l'utilisation de bibliothèques comme scikit-learn et XGBoost pour expérimenter des modèles et développer des API avec FastAPI. Son intégration avec divers outils et la prise en charge des extensions en font l'environnement idéal pour développer rapidement des applications.

- **Flutter**

Flutter est un framework d'interface utilisateur open source créé par Google qui permet la création d'applications natives, contemporaines et réactives pour les ordinateurs de bureau, le Web, Android et iOS à partir d'une seule source de code utilisant le langage de programmation Dart et le moteur graphique Skia.

Raison de la sélection: Flutter est privilégié pour le développement mobile multiplateforme en raison de sa rapidité, de sa flexibilité et de sa capacité à créer des interfaces utilisateur élégantes tout en réduisant le temps de développement. De plus, Flutter est désormais également utilisé pour le développement d'applications de bureau, permettant de créer des applications multiplateformes performantes pour Windows, macOS et Linux, avec une interface cohérente et un code partagé entre différentes plateformes [35].

- **FastAPI**

FastAPI est un framework moderne conçu pour créer des API Web faciles à entretenir et performantes, tout en gérant le traitement asynchrone, la validation des données et l'authentification.

Raison de la sélection: FastAPI a été choisi en raison de sa rapidité, de sa facilité d'utilisation et de sa capacité à gérer efficacement de nombreuses requêtes, ce qui le rend idéal pour développer des API médicales interactives et évolutives [36].

- **Firebase**

Firebase est une plateforme de développement prise en charge par Google qui permet la création d'applications Web et mobiles avec des outils d'analyse, de gestion des erreurs et d'optimisation des performances.

Raison de la sélection: Firebase a été choisi en raison de sa fonctionnalité d'authentification intégrée, qui facilite le développement des processus de connexion et de gestion des utilisateurs. Son système sécurisé permet de simplifier l'enregistrement et la sauvegarde des données utilisateur, tout en assurant une intégration fluide et une évolution rapide. De plus, **Firestore** permet de stocker les données utilisateur de manière sécurisée et évolutive, ce qui en fait la solution idéale pour gérer et synchroniser les informations des utilisateurs dans l'application [37].

- **Swagger**

Swagger est une norme de spécification largement utilisée pour la documentation des API Web. Elle permet d'automatiser et de rationaliser de nombreuses tâches de développement logiciel, telles que la visualisation et le test des API RESTful.

Raison de la sélection: Swagger a été retenu pour générer une documentation claire et interactive de l'API du projet. Il permet aux développeurs de tester les requêtes, visualiser les réponses et assurer une intégration fluide avec d'autres systèmes grâce à une documentation toujours à jour [38].

III.4.3) Développement de l'API

Le développement d'une interface de programmation d'application (API) à l'aide de FastAPI vise à connecter le modèle de diagnostic XGBoost aux applications des utilisateurs (mobile, bureau, site web) de manière sécurisée et évolutive. L'API facilitera une communication efficace, la gestion des requêtes asynchrones et

répondra aux besoins des patients, des médecins , tout en offrant une possibilité d'extension à d'autres plateformes.

III.4.3.1) Structure de l'API multi-usages

A. Définition des exigences de l'API

L'API est conçue pour faciliter l'interaction entre le système et les modèles entraînés avec différents acteurs, tels que **les utilisateurs** (patients)et **les médecins**. Plusieurs **points de terminaison (Endpoints)** seront créés pour garantir une interaction fluide et sécurisée avec l'API.

B. Points de terminaison (Endpoints)

- **/predict:**
  - **But** : Ce point de terminaison est destiné aux utilisateurs pour soumettre leurs données (telles que les symptômes, l'âge, le poids) et obtenir un diagnostic potentiel.
  - **Fonctionnement :** Ce point enverra les données au modèle entraîné via FastAPI, qui analysera les informations fournies et renverra les trois diagnostics les plus probables, accompagnés de la famille des médicaments associés et des conseils médicaux pour chaque diagnostic. Si la probabilité d'un diagnostic potentiel est inférieure à 10 %, il ne sera pas affiché comme résultat, même s'il figure parmi les trois diagnostics les plus probables. Le système trie toutes les probabilités des diagnostics et renvoie uniquement les trois premiers avec les probabilités les plus élevées.

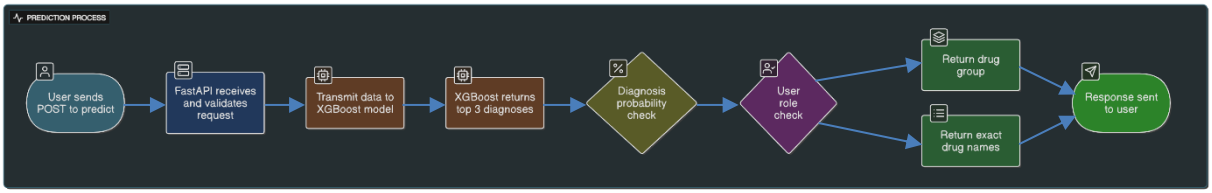


Figure 31 *Prediction process*

Exemple de Requête de Prédiction :

L'image montre une structure de données JSON contenant les détails du patient (par exemple, âge, poids, symptômes..) soumis via un point de terminaison API (/predict). Ces données sont chiffrées et envoyées à un modèle entraîné via **FastAPI**, qui les traite et renvoie des diagnostics potentiels à l'utilisateur, qu'il soit médecin ou patient.

```
{
  "Âge": 35,
  "Sexe": "Femme",
  "Enceinte": "Non",
  "Taille_cm": 175,
  "Poids_kg": 70,
  "IMC": 22.86,
  "Groupe_Sanguin": "A+",
  "Statut_Matrimonial": "Marié",
  "Sport": "Oui",
  "Profession": "Ingénieur informatique",
  "Maladies_Chroniques": "Hypertension",
  "Prend_Medicaments": "Oui",
  "Liste_Medicaments": "Lisinopril",
  "Allergies_Medicamenteuses": "Pénicilline",
  "Antecedents_Chirurgicaux": "Appendicectomie",
  "Dispositif_Medical_Implante": "Non",
  "Fumeur": "Non",
  "Cigarettes_Jour": 0,
  "Alcool": "Non",
  "Antecedents_Familiaux": "Diabète type 2",
  "Symptomes_Recurrents": "Brûlures d'estomac",
  "Sommeil_heures": 7,
  "Symptomes_Actuels": "Fièvre, Toux, Douleurs musculaires",
  "Intensite_Symptomes": "Modérée",
  "Duree_Apparition_Symptomes_jours": 3,
  "user_role": "patient"
}
```


Figure 32 Exemple de Requête de Prédiction

Exemple de Réponse de l'API /predict

L'image montre une réponse JSON de l'API /predict contenant plusieurs diagnostics possibles (maladie cardiaque ischémique, hypothyroïdie, dermatite de contact) avec des détails pour chaque diagnostic : une prédiction encodée, une classe de confiance, et une méthode de traitement. Les données sont traitées et renvoyées par un modèle via FastAPI après réception et chiffrement.

- Si l'utilisateur est un patient, seuls les groupes de médicaments (ex. bêta-bloquants, statines, lévothyroxine) sont affichés, sans noms spécifiques de médicaments.
- Si l'utilisateur est un médecin, les noms exacts des médicaments (ex. aspirine, bêta-bloquants, statines) sont également inclus. Cette distinction dépend du rôle de l'utilisateur défini dans la requête (ici, "user_role": "patient").

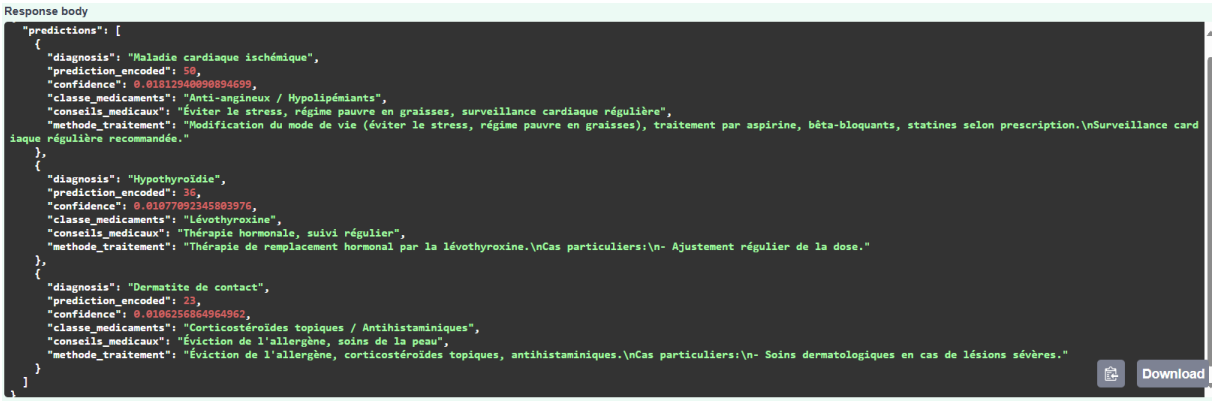


Figure 33 *Exemple de Réponse de l'API /predict*

- **/upload :**
  - But : Permettre aux médecins de télécharger un fichier CSV afin de mettre à jour les données d'entraînement et de réentraîner le modèle XGBoost. Cette fonctionnalité leur offre la possibilité de réajuster et d'améliorer le modèle en fonction de nouvelles données, afin d'optimiser la précision et la pertinence des diagnostics fournis.
  - Fonctionnement : Réception d'un fichier CSV, vérification de son format, fusion des données avec la base existante, traitement avec SMOTE, réentraînement du modèle via XGBClassifier, et sauvegarde du modèle et des encodeurs. Retourne un message de remerciement avec le pourcentage de précision ou une erreur en cas d'échec du téléchargement.

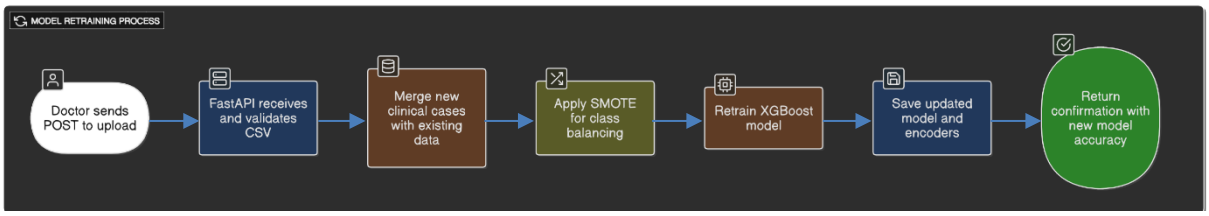


Figure 34 *Model re training process*

III.4.4) Intégration de l'API avec plusieurs plateformes pour une gestion adaptée des utilisateurs

L'API est intégrée avec diverses plateformes, offrant plusieurs méthodes d'interaction pour répondre aux besoins spécifiques des utilisateurs. Dans notre cas, un **application mobile** sera dédiée aux **patients**, tandis qu'une **application de bureau** sera conçue spécifiquement pour les **médecins**, garantissant ainsi une expérience adaptée à chaque type d'utilisateur.

III.4.4.1) Développement de l'application mobile avec API

Objectif : L'application mobile a été développée spécifiquement pour les patients afin de leur permettre de diagnostiquer leur état de santé avant de consulter un médecin.

Technologie : L'application a été conçue à l'aide de Flutter, assurant une compatibilité multiplateforme (iOS et Android).

Intégration : Elle est intégrée à l'API, en particulier avec le point de terminaison (/predict) permettant aux utilisateurs de soumettre leurs données (symptômes, âge, poids, etc.) et de recevoir des diagnostics potentiels.

Cycle de vie d'une application

- **Connexion ou inscription :**

L'utilisateur ou le patient peut se connecter à l'application via la première interface **«a»** s'il possède déjà un compte, ou bien procéder à l'inscription à travers l'interface **« b »** s'il n'a pas encore de compte. Pour cette dernière option, l'authentification est gérée de manière sécurisée grâce à **FirebaseAuth**, qui facilite le processus d'inscription et de connexion, tout en assurant la protection des données personnelles de l'utilisateur. FirebaseAuth permet une intégration fluide et rapide du système d'authentification avec diverses méthodes telles que l'email et le mot de passe, ou d'autres options comme l'authentification par Google compte.

Figure 35 *"a"Se connecter*

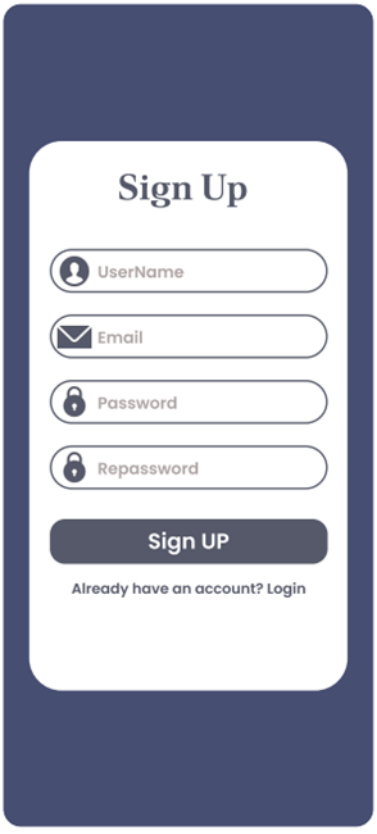


Figure 36 *"b" Créer un compte*

- **Formulaire d'inscription :**

Dans le cas où le patient est nouveau et s'inscrit pour la première fois, il sera redirigé vers la page « **c1** »,où il devra saisir ses informations personnelles, telles que la taille, le poids et d'autres données pertinentes. Ensuite, le patient passera successivement par les pages « **c2** » et « **c3** », où il complétera son historique médical et d'autres informations nécessaires. Toutes ces données seront stockées dans Firebase Firestore, permettant un accès sécurisé et distant. Le patient pourra y accéder à tout moment ou les modifier via les paramètres de l'application. Ces informations sont essentielles pour le processus de diagnostic ultérieur et garantissent un suivi médical précis et personnalisé.

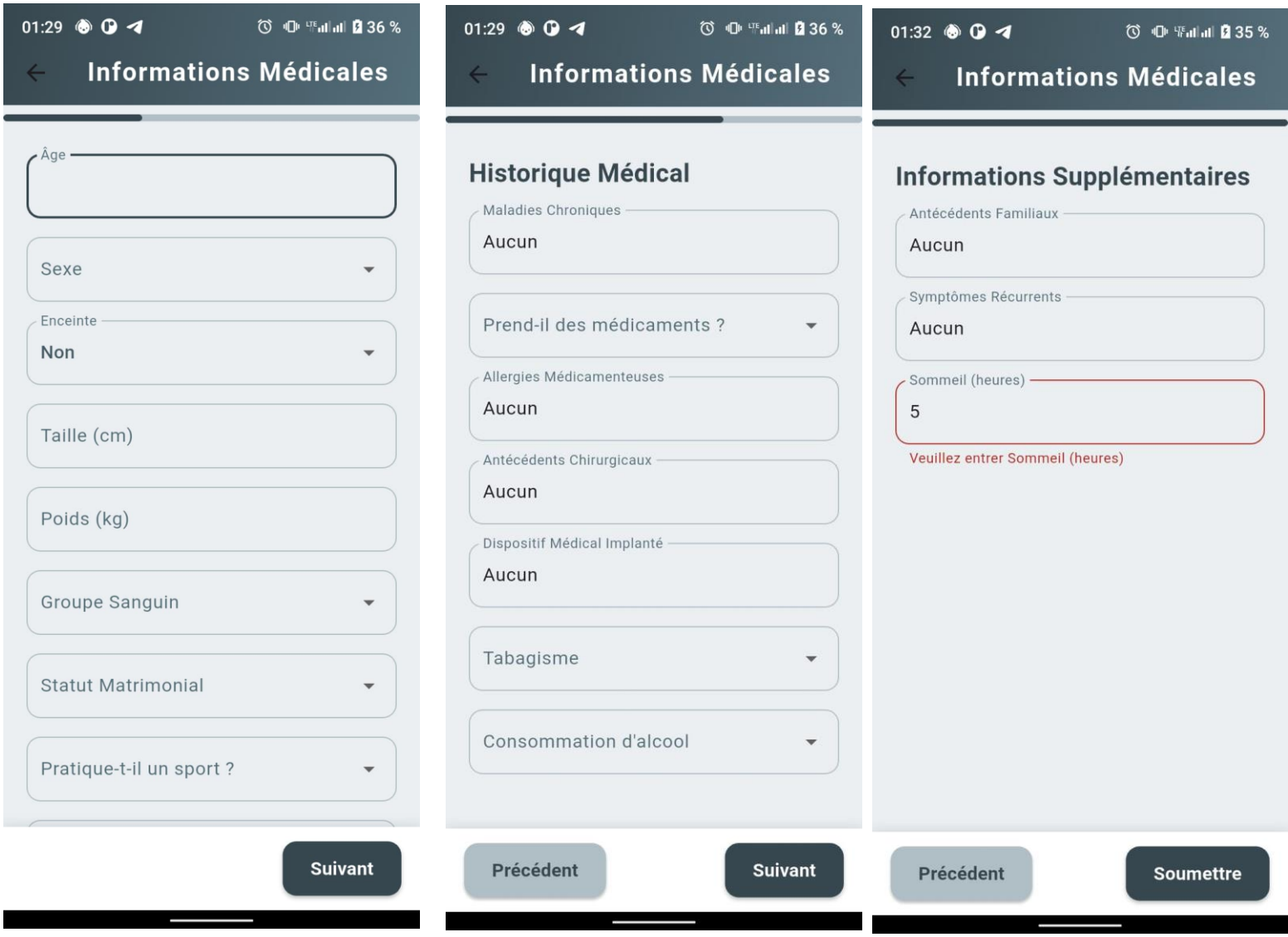


Figure 37 *Informations Médicales*

(c1) **(c2)** **(c3)**

Accueil

La **page d'accueil** « d » représente un tableau de bord contenant une série de services et affiche le profil du patient. À partir de cette page, l'utilisateur peut accéder à la page de Diagnostic médical via une option dédiée, dans le but d'obtenir un nouveau diagnostic. Cette fonctionnalité permet au patient de consulter les services disponibles et de passer à l'étape suivante pour recevoir un diagnostic précis basé sur ses données médicales.

Figure 38 "d" Accueil

Traitement et résultats du diagnostic :

Dans la **page de Diagnostic médical** « e », lorsqu'on clique sur « **Choisir un symptômes** », une liste de symptômes apparaît, permettant au patient de sélectionner ceux qui correspondent à ses conditions. Lors de la sélection de chaque symptôme, le patient peut également indiquer **l'intensité des symptômes** et **le nombre de jours** pendant lesquels il a ressenti ces symptômes. Une fois que tous les symptômes sont sélectionnés et que les informations sont complètes, le patient peut cliquer sur «**obtenir un diagnostic** », ce qui enverra une demande de diagnostic. Cette demande contiendra non seulement les informations fournies lors de l'inscription, mais aussi les données collectées sur la page de diagnostic médical, assurant ainsi un diagnostic personnalisé et précis.

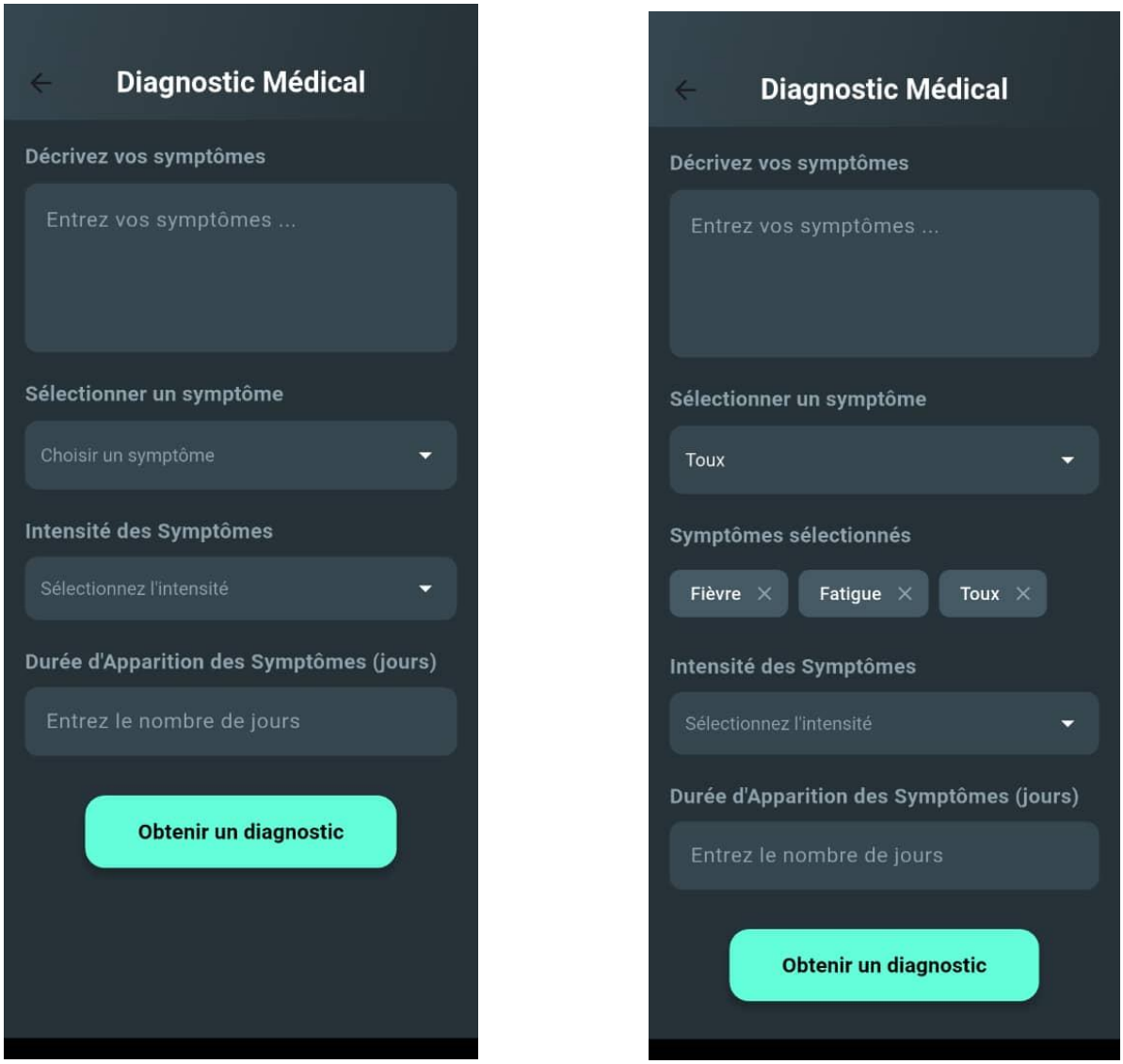


Figure 39 "e" *Diagnostic Medical*

Résultats de diagnostic

Dans cette page « f », les informations ou le réponse provenant de l'API seront reçues et traitées. Ces données seront ensuite organisées et affichées sous forme des trois maladies les plus probables, accompagnées de leurs pourcentages de probabilité. Pour chaque maladie, des conseils médicaux spécifiques seront fournis. Le patient pourra ainsi consulter son diagnostic, comprenant non seulement les maladies les plus susceptibles d'expliquer ses symptômes, mais aussi des recommandations adaptées pour chaque cas, afin de guider le patient dans ses choix de traitement ou de consultation médicale supplémentaire.

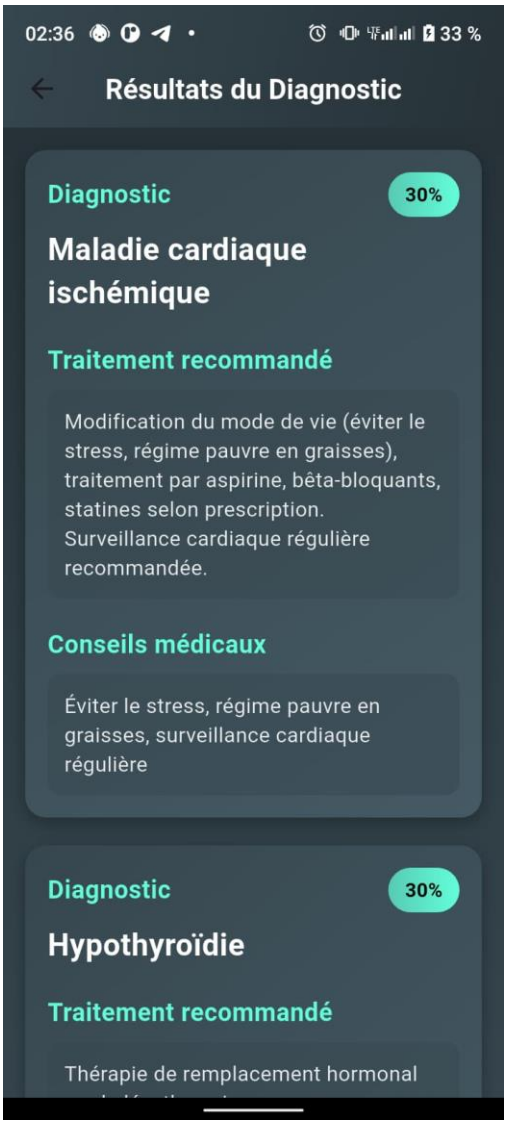


Figure 40 "f" *Résultats du Diagnostic*

III.4.4.2) *Développement de l'application desktop avec API :*

Objectif : L'application desktop est conçue pour les médecins afin de suivre les patients, de les orienter et de les assister dans la détermination du diagnostic approprié.

Technologie : Développée à laide de Flutter pour une interface utilisateur fluide et compatible avec les systèmes d'exploitation desktop.

Intégration : L'application est intégrée à l'API, en particulier avec le point de terminaison **/predict**, permettant aux médecins de soumettre les données des patients (symptômes, âge, poids, etc.) pour obtenir des diagnostics potentiels. De plus, elle permet aux médecins de télécharger des fichiers CSV via le point de terminaison **/upload** pour mettre à jour les données, réentraîner le modèle et l'enrichir avec de nouvelles données, améliorant ainsi ses performances.

Accueil

La page d'accueil contient une interface dédiée aux opérations du médecin, offrant un accès facile aux différentes fonctionnalités. Le médecin peut, depuis cette page, ajouter un nouveau patient ou ajouter une visite pour un patient existant. De plus, la page présente les dernières visites des patients récents, permettant au médecin d'accéder rapidement aux informations des patients qu'il a récemment consultés. Cette interface simplifiée permet une gestion fluide et efficace des consultations et des suivis médicaux.

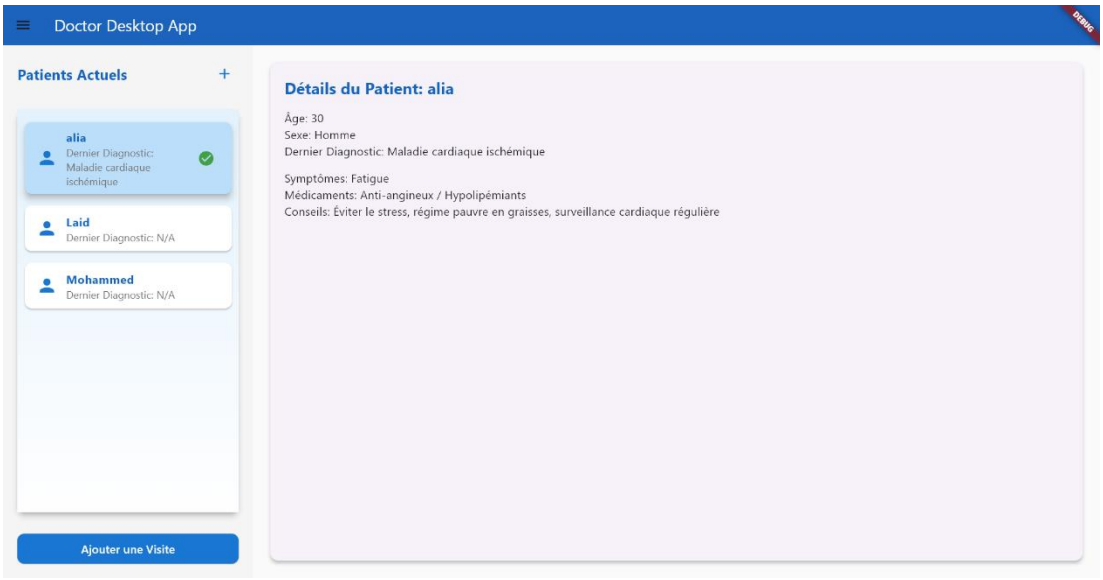


Figure 41 "a" *Accueil*

Ajouter un patient – Informations personnelles et antécédents médicaux
Dans le cas où un nouveau patient doit être ajouté, le médecin pourra accéder à cette fonctionnalité en cliquant sur l'icône « + » située dans le coin supérieur gauche de la page. Il sera alors redirigé vers une page unique où il pourra saisir les informations personnelles du patient, telles que la taille, le poids, et d'autres données pertinentes. Le médecin complétera également les antécédents médicaux du patient ainsi que toute autre information nécessaire. Toutes ces données seront stockées dans Firebase Firestore, permettant un accès sécurisé et distant. Le médecin pourra consulter et modifier ces informations à tout moment, garantissant ainsi un suivi médical précis et personnalisé pour chaque patient.

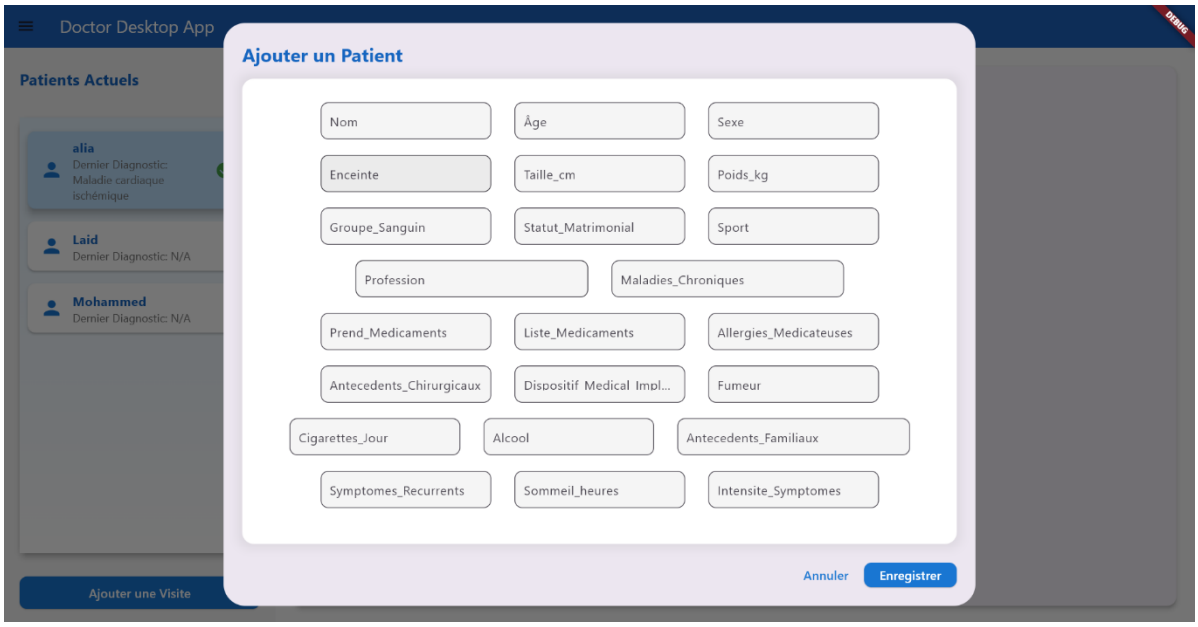


Figure 42 "b" Ajouter un Patient

Ajouter une visite – Saisie des symptômes actuels

Dans la page dédiée à l'ajout d'une visite, le médecin pourra choisir les symptômes actuels du patient à partir d'une liste prédéfinie. Lors de la sélection de chaque symptôme, le médecin devra également indiquer l'intensité des symptômes et le nombre de jours écoulés depuis l'apparition de ces symptômes. Une fois tous les symptômes sélectionnés et les informations complètes, le médecin pourra cliquer sur « Suivant », ce qui l'amènera à l'étape suivante pour finaliser la visite. Cette procédure garantit la collecte d'informations détaillées et pertinentes pour établir un diagnostic précis et personnalisé du patient.

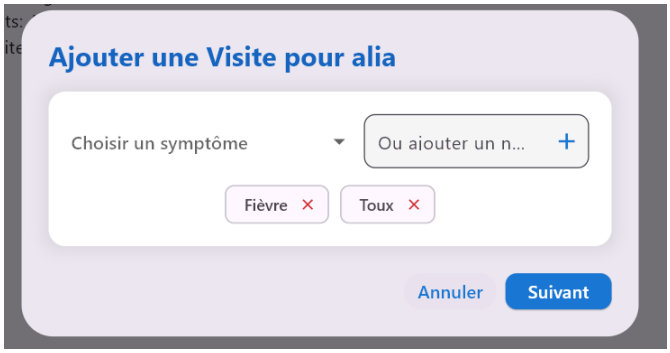


Figure 43 "c" ajoutée une Visite .

Sélection ou ajout d'un diagnostic

Dans la page « f », le médecin verra la liste des trois maladies les plus probables, accompagnées de leurs pourcentages de probabilité, ainsi que des médicaments potentiels à prescrire et des conseils médicaux pour chaque maladie. Le médecin pourra alors sélectionner l'une des maladies proposées ou bien ajouter un autre diagnostic si nécessaire. Une fois le diagnostic choisi ou ajouté, le médecin pourra sauvegarder les informations. Cette fonctionnalité permet au médecin de

personnaliser le diagnostic en fonction de l'état du patient tout en offrant des recommandations adaptées et spécifiques à chaque situation.

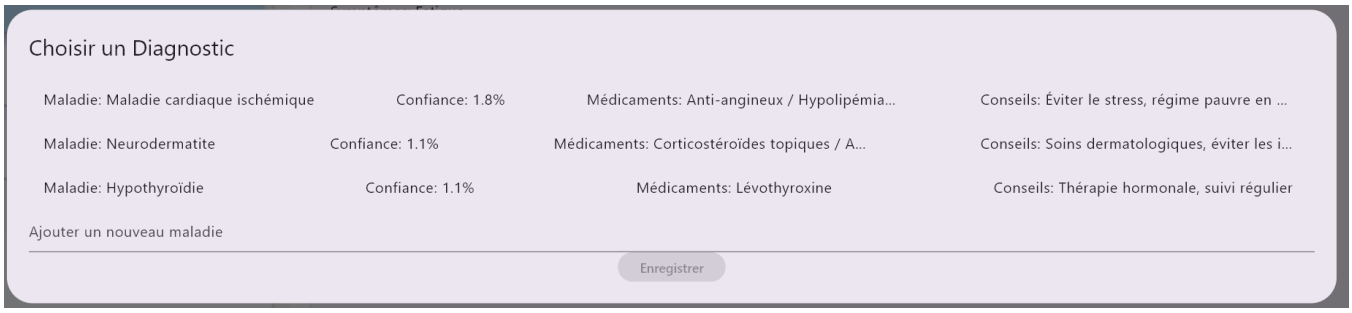


Figure 44 *"d" choisir un Diagnostic*

Téléversement et réentraînement du modèle

Dans la page « Training » « e », le médecin pourra visualiser le nombre de dossiers qu'il a entraînés, ainsi que le nombre de dossiers qui n'ont pas encore été entraînés. Un bouton « Télécharger » permettra au médecin de téléverser de nouveaux fichiers pour l'entraînement du modèle. Lorsque le médecin clique sur ce bouton, une fenêtre apparaîtra « f », offrant la possibilité de sélectionner les fichiers qu'il souhaite télécharger. De plus, il pourra exclure certains fichiers qu'il ne souhaite pas inclure dans le processus de réentraînement. Après avoir confirmé la sélection des fichiers, le médecin pourra procéder à la réinitialisation de l'entraînement en cliquant sur «Confirmer». Ce processus utilisera l'endpoint « upload » pour envoyer les fichiers au modèle et relancer l'entraînement avec les nouvelles données fournies, permettant ainsi une mise à jour continue du modèle pour améliorer sa précision.

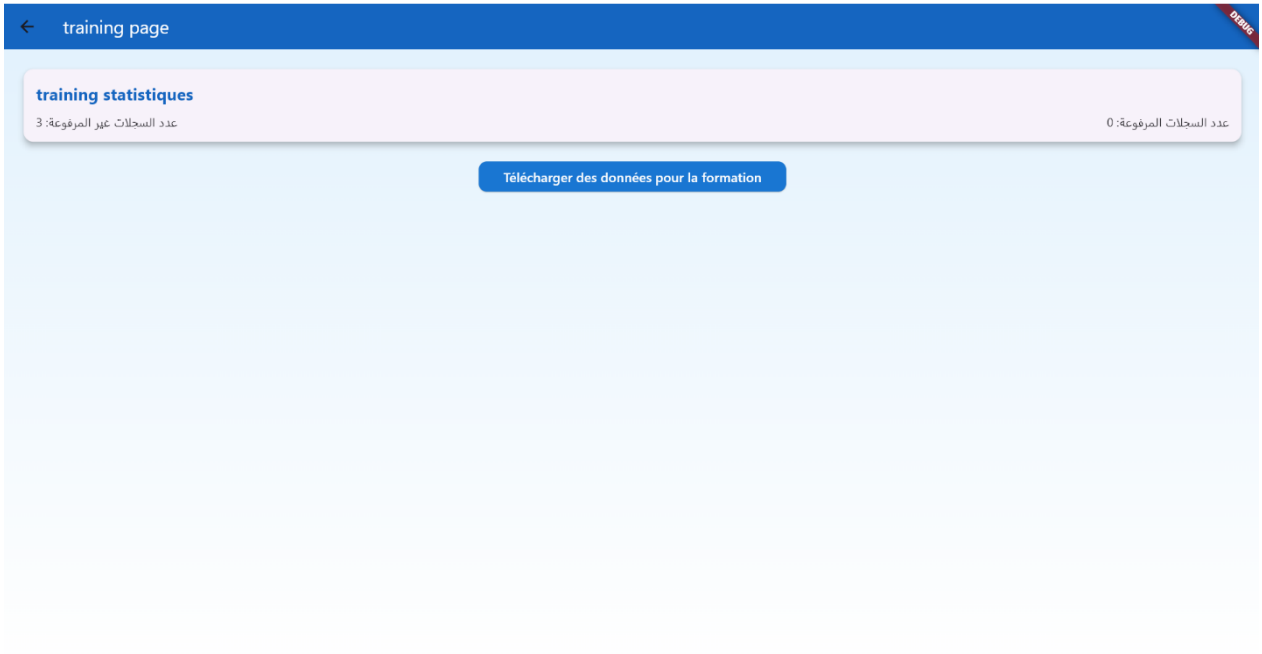


Figure 45 "e" Training page

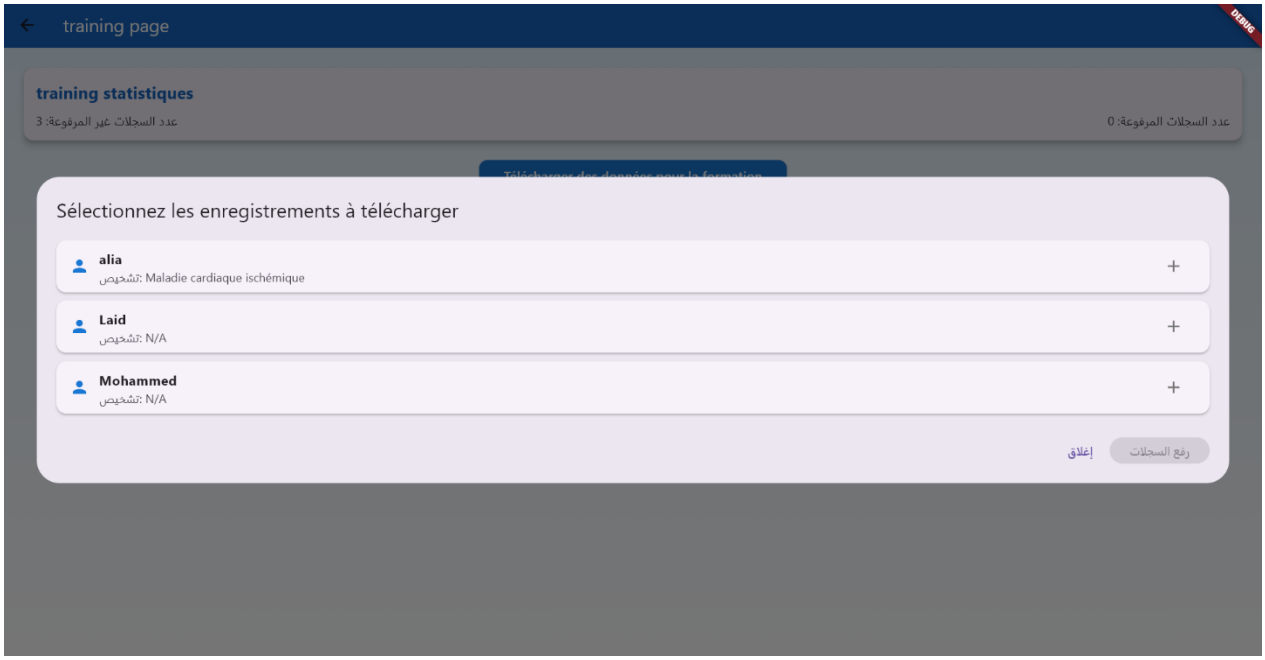


Figure 46 "f" sélectionez les enregistrements

III.5) Conclusion

Ce chapitre a exploré l'entraînement et l'évaluation des modèles XGBoost, Random Forest, et SVM sur des bases de données médicales. Les résultats ont montré que, pour la classification binaire, le modèle SVM a offert des performances exceptionnelles, tandis que pour la classification multi-classes, Random Forest et XGBoost ont montré une grande robustesse. Notamment, XGBoost a non seulement atteint des résultats de haute précision, mais il s'est également distingué par sa rapidité d'entraînement, ce qui en fait un choix optimal pour des applications nécessitant le traitement de grandes quantités de données dans des environnements dynamiques.

Les résultats de l'évaluation initiale, avec une précision parfaite de 100 % sur l'ensemble de test, ont confirmé la solidité des modèles. Toutefois, la vérification du surapprentissage reste cruciale pour s'assurer que ces modèles puissent généraliser efficacement sur de nouvelles données. La validation croisée et l'analyse des courbes d'apprentissage ont montré que, même avec des résultats excellents sur les données de test, une attention particulière doit être portée à la gestion des données déséquilibrées et à l'adaptation des modèles au fil du temps.

En conclusion, bien que Random Forest et XGBoost soient tous deux efficaces pour la gestion des données massives, XGBoost se distingue par sa rapidité d'entraînement, ce qui en fait le modèle le plus adapté pour les applications médicales où le temps de réponse est essentiel. Grâce à sa capacité à traiter efficacement des volumes de données importants, XGBoost représente un choix privilégié pour cette étude.

Conclusion générale :

À travers ce projet, nous avons conçu et développé une plateforme intelligente de diagnostic médical basée sur l'apprentissage automatique, dans le but de répondre à des besoins concrets du domaine de la santé. Ce système vise à assister les professionnels dans la prise de décision, à proposer des traitements personnalisés, et à améliorer la qualité des soins tout en réduisant la charge de travail médicale.

La première étape a consisté à constituer une base de données médicale personnalisée à partir de sources variées. Un travail rigoureux de nettoyage, de traitement et de structuration a permis d'obtenir un jeu de données fiable, adapté à l'entraînement des modèles d'intelligence artificielle. Nous avons ensuite appliqué trois méthodes de classification — XGBoost, Random Forest et SVM — d'abord sur des jeux de données standards (Breast Cancer et Heart Disease), puis sur notre propre base, dans le but de comparer leurs performances et identifier la plus efficace.

Des techniques comme SMOTE et l'ajustement des poids ont été utilisées pour corriger les déséquilibres entre classes. Une attention particulière a également été portée à la détection du surapprentissage, afin d'assurer la généralisation des modèles. L'algorithme XGBoost s'est démarqué par sa stabilité et ses hautes performances, notamment en précision et en robustesse face aux données médicales complexes.

Le modèle final a été intégré dans deux types d'interfaces complémentaires : une application mobile destinée aux patients pour un usage autonome, et une application de bureau pour les professionnels de santé afin de faciliter leur prise de décision. Le tout a été développé dans un environnement technique moderne combinant Python, FastAPI pour le back-end, Flutter pour l'interface mobile, et Swagger pour la documentation API.

Malgré les défis rencontrés, notamment en matière de collecte de données médicales fiables, les résultats obtenus sont prometteurs.

Ainsi, ce projet constitue une première étape concrète vers l'intégration effective de l'intelligence artificielle dans le domaine médical algérien, contribuant à améliorer la qualité des soins et à fournir des outils de diagnostic intelligents aux professionnels de santé comme aux patients.

Liste des références

[1] C. Sanjuanita, V. Ortiz, J. P. Mendoza, V. Villanueva-Hernandez, G. Tijerina, et D. Guzmán, *Languages With Artificial Intelligence Applications*. 2024. doi: 10.4018/979-8-3693-1119-6.ch010.

[2] B. Singh, A. Chandra, D. Joshi, N. Semwal, G. Kukreti, et U. R. Saxena, *Application of Artificial Intelligence Techniques in Healthcare*. 2024. doi: 10.2174/9789815256864124010005.

[3] M. Aldergham, A. Alfouri, et R. A. Madat, « Artificial Intelligence in Medicine », *South East. Eur. J. Public Health*, p. 774-790, 2024, doi: 10.70135/seejph.vi.1561.

[4] C.-A. Azencott, *Introduction au Machine Learning*.

[5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd éd. O'Reilly Media, 2019.

[6] A. Ng, « Machine Learning Yearning ». 2018.

[7] I. Goodfellow, Y. Bengio, et A. Courville, *Deep Learning*. MIT Press, 2016.

[8] IBM, « Qu'est-ce que l'apprentissage supervisé? » [En ligne]. Disponible sur: https://www.ibm.com/fr-fr/topics/supervised-learning

[9] Coursera, « Qu'est-ce que l'apprentissage supervisé? » [En ligne]. Disponible sur: https://www.coursera.org/fr-FR/articles/supervised-learning

[10] DataCamp, « Apprentissage supervisé en Machine Learning ». [En ligne]. Disponible sur: https://www.datacamp.com/fr/blog/supervised-machine-learning

[11] J. Han et M. Kamber, *Data Mining: Concepts and Techniques*. 2006.

[12] R. S. Sutton et A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[13] M. Lapan, *Deep Reinforcement Learning Hands-On*. Packt Publishing, 2018.

[14] P. Nand et N. Sharma, *Artificial Intelligence in Healthcare*. 2024.

[15] P. Rajpurkar et others, « Deep Learning for Detection of Lung Diseases in Radiographs », *Nat. Med.*, 2017.

[16] B. H. Menze et others, « Brain Tumor Segmentation Using Deep Learning », *IEEE Trans Med Imaging*, 2015.

[17] Z. Zhang et others, « Machine Learning for Epileptic Seizure Detection Using fMRI », *J. Neurol. Neurosci.*, 2018.

[18] G. E. Batista et M. C. Monard, « An analysis of four missing data treatment methods for supervised learning », *Intell. Data Anal.*, 2003.

[19] J. L. Schafer, *Analysis of Incomplete Multivariate Data*. Chapman & Hall, 1997.

[20] A. Esteva et others, « Dermatologist-level classification of skin cancer with deep neural networks », *Nature*, vol. 542, nº 7639, p. 115-118, 2017, doi: 10.1038/nature21056.

[21] Z. Obermeyer et others, « Dissecting racial bias in an algorithm used to manage the health of populations », *Science*, 2019.

[22] M. Frid-Adar et others, « GAN-based synthetic medical image augmentation », *IEEE Trans Med Imaging*, 2018.

[23] B. McMahan et others, « Communication-efficient learning of deep networks from decentralized data », in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, 2017.

[24] A. Vaswani et others, « Attention Is All You Need », *Adv. Neural Inf. Process. Syst.*, 2017.

[25] N. Cristianini et J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[26] M. Harrison, *Effective XGBoost: Tuning, Understanding, and Deploying Classification Models*. 2017.

[27] R. Mitchell et E. Frank, « Accelerating the XGBoost algorithm using GPUs », *PeerJ Comput Sci*, 2017.

[28] T. Wang et others, « Maintenance prédictive des ouvrages d'art avec des fondations en sites aquatiques », in *Conference/Report*, 2018.

[29] Z. M'Hamedi et others, « RFIViz: Random Forest Interactive Visualisation, un outil simple pour comprendre les modèles », in *Conference Paper*, 2020.

[30] W. Wolberg, O. Mangasarian, N. Street, et W. Street, « Breast Cancer Wisconsin (Diagnostic) ». 1993. [En ligne]. Disponible sur: https://doi.org/10.24432/C5DW2B

[31] A. Janosi et others, « Heart Disease [Dataset] ». 1989. doi: 10.24432/C52P4X.

[32] L. S. Bickley, *Bates' Guide to Physical Examination and History Taking*, 12th éd. Wolters Kluwer, 2017.

[33] N. J. Talley et S. O'Connor, *Clinical Examination: A Systematic Guide to Physical Diagnosis*, 6th éd. Elsevier, 2014.

[34] G. van Rossum et P. D. Team, *Python Tutorial*. 2020.

[35] M. L. Napoli, *Beginning Flutter: A Hands-On Guide to App Development*. Wiley, 2020.

[36] B. Lubanovic, *FastAPI*. 2020.

[37] P. Chougale, V. Yadav, et A. Gaikwad, « FIREBASE – Overview and Usage », 2020.

[38] S. Casas et others, « Uses and applications of the OpenAPI/Swagger specification: a systematic mapping of the literature », in *Conference Paper*, 2020.

Résume

Ce mémoire s'inscrit dans le cadre du développement d'un système intelligent d'aide au diagnostic médical, basé sur les techniques d'apprentissage automatique. L'objectif principal est de concevoir une application capable de proposer un diagnostic accompagné de traitements et de recommandations personnalisées, en se fondant à la fois sur les symptômes actuels et sur l'historique médical complet du patient (antécédents, allergies, traitements, etc.).
Dans ce projet, nous avons constitué une base de données médicale personnalisée à partir de sources variées, puis appliqué trois méthodes de classification supervisée Support Vector Machine (SVM), Random Forest et XGBoost à des cas médicaux réels sur deux jeux de données issus du UCI Machine Learning Repository : Breast Cancer Wisconsin (classification binaire) et Heart Disease (classification multiclasse), en complément de notre propre base construite. L'objectif était d'évaluer la performance des différents classificateurs et d'identifier celui le plus adapté à l'intégration dans notre application.
Un travail rigoureux de prétraitement des données a été réalisé, comprenant le nettoyage, la normalisation et l'optimisation des paramètres.
L'évaluation des modèles, fondée sur plusieurs métriques (précision, rappel, F1-score, exactitude), a permis d'identifier XGBoost comme la méthode la plus performante pour notre application.
Le système a été développé en s'appuyant sur une architecture technique moderne, combinant Python et **FastAPI** pour le backend, et Flutter pour l'interface mobile, assurant ainsi une solution multiplateforme, flexible et rapide à déployer.
Ce travail représente une avancée concrète vers l'intégration de l'intelligence artificielle dans les outils d'aide à la décision médicale, en particulier dans les contextes à ressources limitées.

Mots-clés :

Apprentissage automatique , Classification , Diagnostic médical , XGBoost, Intelligence artificielle, Données médicales.

الملخص

يأتي هذا العمل في إطار تطوير نظام ذكي للمساعدة في التشخيص الطبي، يعتمد على تقنيات التعلم الآلي. يتمثل الهدف الرئيسي في تصميم تطبيق قادر على اقتراح تشخيص مرفق بالعلاجات والتوصيات الشخصية، استنادًا إلى الأعراض الحالية للمريض بالإضافة إلى تاريخه الطبي الكامل (سوابق مرضية، حساسية، علاجات، إلخ).

في هذا المشروع، قمنا بإنشاء قاعدة بيانات طبية مخصصة من مصادر متنوعة، ثم طبقنا ثلاث خوارزميات تصنيف خاضع للإشراف، وهي: آلة الدعم الناقل(SVM) ، وغابة القرار العشوائي(Random Forest) ، وخوارزميةXGBoost ، على حالات طبية حقيقية. كما تم استخدام مجموعتي بيانات من مكتبة التعلم الآلي **:** UCIسرطان الثدي (تصنيف ثنائي) وأمراض القلب (تصنيف متعدد)، كمكمل لقاعدتنا الخاصة. وكان الهدف هو تقييم أداء المصنفات المختلفة وتحديد الأنسب لدمجه في تطبيقنا.

وقد تم تنفيذ عمل دقيق لمعالجة البيانات المسبقة، شمل التنظيف، والتطبيع، وتحسين المعلمات. أظهرت نتائج التقييم – استنادًا إلى عدة مقاييس مثل الدقة، والاسترجاع، وF1-score، والنسبة الإجمالية للصحة – أن خوارزمية XGBoostهي الأكثر كفاءة وملاءمة لتطبيقنا.

تم تطوير النظام بناءً على بنية تقنية حديثة، باستخدام Python و FastAPI للواجهة الخلفية، و Flutter للواجهة الأمامية على الهاتف المحمول، مما يضمن حلاً متعدد المنصات، مرنًا وسريع النشر. ويمثل هذا العمل خطوة عملية نحو دمج الذكاء الاصطناعي في أدوات دعم القرار الطبي، خاصة في البيئات ذات الموارد المحدودة.

الكلمات المفتاحية:

التعلم الآلي، التصنيف، التشخيص الطبي، XGBoost، الذكاء الاصطناعي، البيانات الطبية.

Abstract

This thesis is part of the development of an intelligent medical diagnosis support system based on machine learning techniques. The main objective is to design an application capable of providing diagnostic suggestions along with personalized treatments and recommendations, based on both current symptoms and the patient's complete medical history (including medical background, allergies, treatments, etc.).

In this project, we created a custom medical database from various sources and applied three supervised classification methods—Support Vector Machine (SVM), Random Forest, and XGBoost — to real-world medical cases. Additionally, two datasets from the UCI Machine Learning Repository were used: Breast Cancer Wisconsin (binary classification) and Heart Disease (multiclass classification), to complement our own constructed dataset. The goal was to evaluate the performance of each classifier and identify the most suitable one for integration into our application.

A rigorous preprocessing workflow was conducted, involving data cleaning, normalization, and parameter optimization. Model evaluation, based on multiple metrics (precision, recall, F1-score, and accuracy), revealed that XGBoost was the most effective method for our application.

The system was developed using a modern technical architecture, combining Python and FastAPI for the backend, and Flutter for the mobile interface, ensuring a cross-platform, flexible, and fast-to-deploy solution. This work represents a concrete step towards the integration of artificial intelligence in medical decision-support tools, especially in resource-limited contexts.

Keywords:

Machine Learning, Classification, Medical Diagnosis, XGBoost, Artificial Intelligence, Medical Data.

Annexes

Demande de facilitation de la mission d'obtention d'informations médicales soumise par l'école

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

République Algérienne Démocratique Populaire

Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Ecole Normale Supérieure de l'enseignement Technologique de Skikda

Département Mathématiques et informatiques

المدرسة العليا لأساتذة

التعليم التكنولوجي بسكيكدة

قسم الرياضيات

الموضوع : طلب تسهيل مهمة

أرجو من سيادتكم الترخيص للطالب (ة) : عاصم علية المسجل في السنة الرابعة أستاذ تعليم متوسط بالمدرسة العليا لأساتذة التعليم التكنولوجي بـ -سكيكدة (للحصول على بيانات طبية لأغراض البحث العلمي) على مستوى عيادتكم .

تقبلوا مني فائق الإحترام والتقدير.

رئيس قسم الرياضيات

رئيس القسم

Échantillon de données médicales collectées auprès des médecins (avec SQL)

	Âge	Sexe	Enceinte	Taille (cm)	Poids (kg)	IMC	Groupe Sang...	Statut Matri...	Sport
1	39	Homme	Non	175	87.2	28.5	AB+	Marié	Non
2	75	Femme	Non	175.8	60.9	19.7	O-	Divorcé	Non
3	70	Homme	Non	170.2	76.9	26.5	AB+	Veuf	Oui
4	32	Homme	Non	184.4	89.7	26.4	AB-	Célibataire	Non
5	12	Homme	Non	150.5	44.8	19.8	B-	Célibataire	Non
6	27	Homme	Non	175.6	70.1	22.7	B+	Divorcé	Non
7	33	Homme	Non	180.1	71.1	21.9	O+	Célibataire	Non
8	25	Femme	Non	173.2	98.5	32.8	B-	Divorcé	Oui
9	29	Homme	Non	155.3	94.9	39.3	AB+	Divorcé	Oui
10	16	Homme	Non	174.4	85.4	28.1	A+	Veuf	Non
11	10	Homme	Non	119.3	32.3	22.7	O+	Célibataire	Non
12	54	Femme	Non	179.6	82.5	25.6	AB-	Célibataire	Oui
13	38	Homme	Non	171.6	88.9	30.2	A-	Veuf	Non
14	12	Femme	Non	103	22.8	21.5	AB+	Célibataire	Non
15	74	Femme	Non	165.9	76.2	27.7	AB-	Veuf	Non
16	47	Femme	Non	175.2	106.2	34.6	O+	Célibataire	Non
17	23	Femme	Non	162.9	96.5	36.4	B+	Veuf	Non
18	69	Homme	Non	177.7	66.5	21.1	AB+	Veuf	Oui
19	18	Homme	Non	161.3	53.4	20.5	AB-	Célibataire	Non
20	66	Homme	Non	178.2	65	20.5	AB-	Divorcé	Non
21	42	Femme	Non	155.7	89.7	37	B-	Célibataire	Oui

.

Diagnostic	Traitement Pr...	Classe Médicaments	Conseils Médicaux	Méthode de traitement
Mononucléose	Aucun	Traitement symptomatique	Repos, hydratation, surveillance	Repos, traitement symptomatique. Cas particuliers: - Surveillance d
Diabète de type 2	Inhibiteur SGLT2	Antidiabétiques oraux / Insuline	Suivi glycémique régulier, régime pauvre en ...	Modification du mode de vie (régime, activité physique), utilisation
Maladie d'Alzheimer	Analgésiques, Antib...	Médicaments contre la perte de mémoire	Accompagnement spécialisé, thérapies adap...	Traitement symptomatique, médicaments contre la perte de mémc
Dermatite séborrhé...	Shampooings antif...	Shampooings antifongiques / Corticostéroïdes topi...	Soins dermatologiques, éviter les irritants	Shampooings antifongiques, corticostéroïdes topiques. Cas particu
Anémie ferriprive	Aucun	Suppléments de fer	Suppléments de fer, alimentation équilibrée	Suppléments de fer, régime alimentaire riche en fer, traitement des
Pancréatite kystique	Gestion de la douleur	Enzymes pancréatiques / Gestion de la douleur	Suivi endocrinien, gestion de la douleur	Gestion de la douleur, enzyme pancréatique, nutrition. Cas particul
Cystite diagnostiqu...	Antibiotiques	Antibiotiques / Analgésiques	Hydratation, bonne hygiène, traitement anti...	Antibiotiques, hydratation, traitement de la douleur. Cas particulier
Vascularite des vais...	Immunosuppresseu...	Immunosuppresseurs / Traitement symptomatique	Suivi médical régulier, rééducation	Immunosuppresseurs, traitement symptomatique. Cas particuliers:
Immunodéficience	Immunoglobulines, ...	Immunoglobulines / Antibiotiques prophylactiques	Prévention des infections, suivi immunologiq...	Traitement des infections, prophylaxie, immunoglobulines. Cas part
Immunodéficience	Antibiotiques prop...	Immunoglobulines / Antibiotiques prophylactiques	Prévention des infections, suivi immunologiq...	Traitement des infections, prophylaxie, immunoglobulines. Cas part
Pharyngite	Aucun	Antibiotiques / Analgésiques	Repos vocal, analgésiques, traitement antibi...	Antibiotiques si bactérienne, analgésiques, repos vocal. Cas particu
Lymphadénite	Antibiotiques	Antibiotiques	Traitement antibiotique complet	Antibiotiques, traitement symptomatique. Cas particuliers: - Draina
Diabète de type 1	Insuline	Insuline	Surveillance glycémique stricte, adaptation d...	Insulinothérapie, surveillance glycémique stricte, adaptation du rég
Dacryocystite	Aucun	Antibiotiques topiques / Anti-inflammatoires	Drainage si nécessaire, hygiène oculaire	Antibiotiques topiques, anti-inflammatoires. Cas particuliers: - Drai
Stéatose hépatique	Modificateurs du m...	Modificateurs du métabolisme lipidique	Modification du mode de vie, suivi régulier	Modification du mode de vie (perte de poids, régime alimentaire),
Stéatose hépatique ...	Modification du mo...	Modification du mode de vie / Contrôle métabolique	Régime alimentaire équilibré, exercice	Modification du mode de vie, régime alimentaire, contrôle métabo
Brucellose	Aucun	Antibiotiques prolongés	Suivi prolongé, traitement antibiotique	Antibiotiques prolongés. Cas particuliers: - Suivi des infections chrc
Dacryocystite	Anti-inflammatoires...	Antibiotiques topiques / Anti-inflammatoires	Drainage si nécessaire, hygiène oculaire	Antibiotiques topiques, anti-inflammatoires. Cas particuliers: - Drai
Thalassémie	Aucun	Transfusions sanguines / Supplémentation en fer	Surveillance des complications, traitement ré...	Transfusions sanguines régulières, supplémentation en fer, suivi hé
Hépatite C	Antiviraux	Antiviraux	Suivi hépatique, traitement antiviral	Antiviraux, suivi hépatique, traitement symptomatique. Cas particul
Épilepsie	Antiépileptiques	Antiépileptiques	Suivi neurologique régulier, respect du traite...	Médicaments antiépileptiques, suivi neurologique régulier. Cas par

Alcool	Antécédents ...	Symptômes ...	Sommeil (he...	Symptômes ...	Intensité Sym...	Durée Appari...
Non	Brucellose	Adénopathie	8	Fatigue, Fièvre	Légère	20
Non	Aucun	Fatigue, Douleurs c...	8	Douleur thoracique,...	Modérée	6
Non	Otite moyenne, Alz...	Difficultés de pensée	9	Difficultés de pensé...	Légère	13
Non	Dermatite séborrhé...	Démangeaisons, Ro...	5	Rougeur	Légère	8
Non	Aucun	Dyspnée, Faiblesse	4	Fatigue, Palpitations	Légère	5
Oui	Pancréatite kystiqu...	Diarrhée, Perte de p...	6	Douleur abdominal...	Légère	21
Non	Cystite, AVC	Fréquence urinaire	5	Douleur abdominal...	Sévère	23
Non	Mononucléose infe...	Fièvre, Éruption cut...	7	Douleur musculaire,...	Légère	19
Non	Aucun	Sueurs nocturnes, P...	5	Infections répétées	Légère	15
Oui	Hépatite auto-imm...	Sueurs nocturnes, P...	9	Perte de poids, Sue...	Légère	22
Non	Sinusite	Douleur et démang...	5	Fièvre, Voix rauque	Modérée	9
Oui	Pancréatite aiguë, A...	Œdème ganglionna...	5	Œdème ganglionna...	Sévère	19
Oui	Hypothyroïdie, Hyp...	Fatigue	7	Perte de poids rapide	Sévère	28
Non	Aucun	Écoulements	5	Écoulements	Sévère	28
Non	Stéatose hépatique	Ballonnement abdo...	9	Douleur légère au c...	Sévère	24
Non	Stéatose hépatique ...	Gain de poids, Doul...	4	Fatigue	Modérée	3
Oui	Mononucléose infe...	Douleurs musculair...	7	Sueurs nocturnes, ...	Légère	17
Non	Mononucléose infe...	Douleur, Écouleme...	4	Rougeur, Écouleme...	Modérée	26
Oui	Otite externe, Canc...	Pâleur, Fatigue sévè...	9	Fatigue sévère	Sévère	25
Non	Rougeole	Nausées, Perte d'ap...	8	Ictère, Perte d'appé...	Modérée	17
Non	Épilepsie	Perte d'équilibre po...	8	Mouvements involo...	Légère	26

Profession	Maladies Chr...	Prend Médic...	Liste Médica...	Allergies Mé...	Antécédents ...	Dispositif Mé...	Fumeur	Cigarettes/Jo...
Fonctionnaire	Mononucléose infe...	Non	Aucun	Aucune	Appendicectomie	Pompe à insuline	Non	0
Aucune	Diabète - Type 2	Oui	Inhibiteur SGLT2	Sulfonylurées, Ibupr...	Appendicectomie	Pacemaker	Non	0
Aucune	Otite moyenne, Alz...	Oui	Analgésiques, Antib...	Aucune	Aucune	Pompe à insuline	Non	0
Cadre	Dermatite séborrhé...	Oui	Shampooings antif...	Pénicilline	Appendicectomie	Pacemaker	Oui	5
Enfant	Anémie ferriprive	Non	Aucun	Aucune	Aucune	Pompe à insuline	Non	0
Chômeur	Pancréatite kystique	Oui	Gestion de la douleur	Sulfonylurées, AINS	Aucune	Pompe à insuline	Non	0
Chômeur	Cystite	Oui	Antibiotiques	Ibuprofène	Césarienne	Pompe à insuline	Non	0
Fonctionnaire	Vascularite des moy...	Oui	Immunosuppresseu...	Aucune	Aucune	Pacemaker	Oui	30
Chômeur	Immunodéficience	Oui	Immunoglobulines, ...	Ibuprofène, Sulfony...	Appendicectomie	Pacemaker	Non	0
Étudiant	Immunodéficience	Oui	Antibiotiques prop...	AINS, Pénicilline	Pontage coronarien	Aucun	Oui	3
Enfant	Pharyngite, Maladie...	Non	Aucun	Aucune	Césarienne	Pompe à insuline	Non	0
Commerçant	Lymphadénite	Oui	Antibiotiques	Pénicilline	Césarienne	Pompe à insuline	Oui	12
Ouvrier	Diabète de type 1	Oui	Insuline	AINS	Aucune	Pompe à insuline	Oui	29
Enfant	Dacryocystite	Non	Aucun	AINS, Pénicilline	Aucune	Pompe à insuline	Non	0
Commerçant	Stéatose hépatique	Oui	Modificateurs du m...	AINS	Pontage coronarien	Pacemaker	Non	0
Ouvrier	Stéatose hépatique ...	Oui	Modification du mo...	Aucune	Pontage coronarien	Pompe à insuline	Oui	2
Fonctionnaire	Brucellose	Non	Aucun	Sulfonylurées, Ibupr...	Aucune	Aucun	Oui	14
Aucune	Dacryocystite	Oui	Anti-inflammatoires...	Sulfonylurées	Césarienne	Pompe à insuline	Non	0
Étudiant	Thalassémie	Non	Aucun	Aucune	Aucune	Pacemaker	Non	0
Commerçant	Hépatite C	Oui	Antiviraux	Aucune	Pontage coronarien	Aucun	Non	0
Ouvrier	Épilepsie	Oui	Antiépileptiques	AINS	Appendicectomie	Aucun	Oui	6

Échantillon de données médicales collectées auprès des médecins (avec CSV)

ÂGE	SEXE	ENCEINTE	TAILLE (CM)	POIDS (KG)	IMC	GROUPE SANGUIN	STATUT MATRIMONIAL	SPORT	PROFESSION	MALADIES CHRONIQUES
66	Homme	Non	163.0	71.7	27.0	O-	Célibataire	Non	Commerçant	Arthrose
54	Homme	Non	167.3	78.9	28.2	O-	Célibataire	Oui	Fonctionnaire	Diabète - Type 2
9	Homme	Non	143.7	40.2	19.5	AB-	Célibataire	Non	Enfant	Anémie ferriprive
8	Homme	Non	153.0	34.5	14.7	O-	Célibataire	Non	Enfant	Sinusite, Maladies Pulmonair
37	Femme	Non	157.0	98.1	39.8	O-	Marié	Non	Commerçant	Asthme allergique
48	Homme	Non	183.3	103.3	30.7	AB+	Divorcé	Oui	Ouvrier	Hypertension Artérielle
19	Homme	Non	189.2	72.2	20.2	O+	Célibataire	Oui	Étudiant	Rage

PREND MÉDICAMENTS	LISTE MÉDICAMENTS	ALLERGIES MÉDICAMENTEUSES	ANTÉCÉDENTS CHIRURGICAUX	DISPOSITIF MÉDICAL IMPLANTÉ
Non	Aucun	Aucune	Césarienne	Pacemaker
Oui	Inhibiteur SGLT2, Metformine, Insuline, Sulfonylurées	Aucune	Appendicectomie	Pacemaker
Oui	Suppléments de fer	AINS	Césarienne	Pacemaker
Oui	Décongestionnants, Antibiotiques	Ibuprofène, AINS	Césarienne	Aucun
Non	Aucun	Sulfonylurées, AINS	Appendicectomie	Aucun
Oui	Bêta-bloquants, Diurétiques, Calcium antagonistes	Pénicilline, Ibuprofène	Aucune	Pompe à insu
Non	Aucun	Pénicilline, AINS	Aucune	Pacemaker

FUMEUR	CIGARETTES/JOUR	ALCOOL	ANTÉCÉDENTS FAMILIAUX	SYMPTÔMES RÉCURRENTS
Oui	24	Oui	Arthrose, Asthme allergique, Migraine	Légère tuméfaction
Oui	10	Non	Hypertension	Douleurs chroniques
Non	0	Non	Anémie ferriprive	Fatigue, Dyspnée, Faiblesse
Oui	23	Non	Pancréatite, Néphrite, Pharyngite	Mauvaise haleine, Écoulements nasaux épais
Non	0	Oui	Arthrose, Borréliose	Oppression thoracique
Oui	1	Non	Aucun	Céphalée, Fatigue
Non	0	Non	Aucun	Hypersensibilité à l'eau, Fièvre, Convulsions, Anxiété

SOMMEIL (HEURES)	SYMPTÔMES ACTUELS	INTENSITÉ SYMPTÔMES	DURÉE APPARITION SYMPTÔMES (JOURS)
5	Légère tuméfaction, Raideur matinale courte	Légère	10
9	Mictions fréquentes, Transpiration froide, Essoufflement	Modérée	2
9	Dyspnée, Palpitations, Faiblesse	Légère	15
4	Mauvaise haleine, Écoulements nasaux épais, Toux nocturne	Sévère	5
5	Toux sèche, Éternuements fréquents, Démangeaisons des yeux et du nez	Sévère	4
5	Dyspnée	Légère	2
5	Fièvre	Modérée	13

	INTENSITÉ SYMPTÔMES	DURÉE APPARITION SYMPTÔMES (JOURS)	DIAGNOSTIC	TRAITEMENT PRESCRIT
	Modérée	14	Dermatite de contact	Corticostéroïdes topiques
	Légère	4	Pharyngite	Antibiotiques
	Modérée	16	Maladie de Parkinson	Médicaments symptomatiques
	Sévère	13	Troubles anxieux diagnostiqués	Anxiolytiques
	Modérée	4	Cancer du poumon	Chimiothérapie
	Sévère	12	Sinusite	Décongestionnants, Analgésiques, Antibiotiques