République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي والبحث العلمي

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Université A/Mira de Bejaia
Faculté des sciences exactes
Département Informatique

Mémoire de fin d'étude

En vue de l'obtention d'un diplôme de Master en informatique
Spécialité : Système d'information Avancée

Thème

Comparaison des algorithmes de classification et l'optimisation du meilleur algorithme

Encadré par :

Mme Khaled Hayette

Présenté par :

Mr ATTAB Agheslane

Mr DEBICHE Imad

Devant le jury d'examen suivant :

Mme Nassima Bouadem	**Présidente**
Mme Souhila Ghanem	**Examinatrice**

Bejaia, 2023.

Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude à Dieu qui nous a donné le courage et la détermination nécessaires pour mener à bien ce travail de recherche.

Nous adressons également nos sincères remerciements à notre encadreur, Khaled Hayette, pour ses conseils avisés, son soutien constant et son expertise précieuse qui ont grandement contribué à la réalisation de ce mémoire.

Un merci tout particulier est également adressé aux membres du jury qui ont accepté de consacrer leur temps et leur expertise pour évaluer ce projet.

Enfin, nous souhaitons exprimer notre reconnaissance envers le département d'informatique et l'ensemble des enseignants qui nous ont transmis leurs connaissances tout au long de notre parcours universitaire.

Nos remerciements vont également à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

Merci encore à tous pour votre précieuse aide et votre soutien tout au long de ce projet

Mr. DEBICHE IMAD Mr. ATTAB AGHESLANE

Dédicaces

Je dédie ce modeste travail :

A mes chers parents Vous êtes mes piliers, mes guides et mes plus grands soutiens tout au long de ma vie

À ma mère, rayon de ma vie, ma source d'inspiration, et je t'aime plus que tout au monde,

À mon père, homme remarquable et modèle de force et de persévérance,

Je suis honoré d'être votre enfant. Votre amour inconditionnel, vos conseils avisés et votre présence constante ont façonné mon parcours. Je vous exprime ma gratitude infinie pour tout ce que vous avez fait et continuez à faire pour moi

À mes frère HAKIM, OUSSAMA mes compagnons de route, vous avez partagé chaque instant de ma vie. Votre soutien inconditionnel, vos encouragements et votre présence ont été précieux pour moi. Vous êtes les meilleurs alliés que j'aurais pu demander et je suis honoré d'avoir grandi à vos côtés.

À mes chers amis : AGHESLANE, RAHIM, ZAKZOUK, DR. RAMZI, MOUAD, OUSSAMA, ISSAM, FOUAD, SHEMSOU, NAAIM

À tous ceux qui me sont chers et tous ceux qui m'ont aidé dans mon travail.

DEBICHE IMAD

Dédicaces

Je dédie ce travail

A ma très chère mère qui m'a soutenu et encouragé durant ces années d'études. Qu'elle trouve ici le témoignage de ma profonde reconnaissance.

A mon très cher père qui a toujours été à mes côtés pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection

A mon frère ANIS, mes sœurs KAHINA et LYLIA qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

A tous mes amis IMAD, RAHIM, ABDERAZAK, YOUCEF, YOUNES, ABDOU, RAFIK, SALIM, AYMEN, ILYES ZOO, SHEMSOU, SLIMANE, ISSAM, LYES BESBASA, WALID, KHEYRO.

A tous ceux que j'aime.

Merci !

ATTAB AGHESLANE

Table des matières

Liste des Figures

Liste des Tableaux

Notations et symboles

KDD: Knowledge Discovery in Databases

KNN: K-Nearest Neighbors

SVM: Support Vector Machines

WEB: World Wide Web

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

OPTICS: Ordering Points to Identify the Clustering Structure

ACP : analyse en composantes principales

VN : vrais négatifs

FP : faux positifs

FN : faux négatifs

VP : vrais positifs

IQR : l'écart interquartile

ANOVA: Analysis of Variance

ROC: Receiver Operating Characteristic

XGBoost : boosting extrême de gradient

KCDC : Korea Centers for Disease Control & Prevention

Introduction générale

Le COVID-19, causé par le virus SARS-CoV-2, a eu un impact majeur sur la santé mondiale et a entraîné des conséquences socio-économiques considérables. Depuis son apparition, la communauté scientifique a travaillé sans relâche pour comprendre le virus et trouver des moyens efficaces de lutter contre sa propagation. Dans ce contexte, la prédiction de l'infection au COVID-19 à l'aide de données cliniques est devenue une priorité pour identifier les individus à risque et prendre des mesures de prévention appropriées.

Contexte général du COVID-19 :

Le COVID-19 est une maladie respiratoire aiguë qui peut entraîner des symptômes allant de légers à graves, voire mortels. Depuis son apparition en décembre 2019, la pandémie de COVID-19 a touché des millions de personnes dans le monde entier, mettant à rude épreuve les systèmes de santé, les économies et les sociétés. La recherche sur le COVID-19 s'est concentrée sur divers aspects, notamment la transmission du virus, les facteurs de risque, les symptômes, les traitements et les vaccins.

Importance de la prédiction de l'infection au COVID-19 :

La prédiction de l'infection au COVID-19 revêt une importance cruciale pour plusieurs raisons. Tout d'abord, la détection précoce des personnes infectées peut contribuer à la prévention de la propagation du virus en leur offrant un traitement rapide et en prenant des mesures de quarantaine appropriées. De plus, la prédiction de l'infection permet d'identifier les individus à risque élevé, tels que les personnes âgées ou celles souffrant de problèmes de santé sous-jacents, afin de leur fournir une attention médicale ciblée.

La prédiction de l'infection au COVID-19 repose sur l'utilisation de données cliniques, telles que les symptômes, les résultats des tests de dépistage, les antécédents médicaux et d'autres caractéristiques pertinentes. L'analyse de ces données permet d'identifier les facteurs qui contribuent à la prédiction de l'infection et de développer des modèles prédictifs fiables.

Problématique :

La problématique principale abordée dans cette étude comparative des méthodes de classification pour la détection de COVID-19 est de déterminer et d’optimiser la méthode de

classification la plus performante pour la détection précise de cette infection virale. Cette problématique soulève plusieurs questions de recherche pertinentes, telles que :

- Quelles sont les méthodes de classification couramment utilisées pour la détection du COVID-19 ?
- Quels sont les critères d'évaluation appropriés pour comparer les performances des méthodes de classification ?
- Quelles sont les performances relatives des différentes méthodes de classification pour la détection du COVID-19 ?
- Comment optimiser la méthode de classification la plus performante pour une détection précise du COVID-19 ?

Organisation du mémoire :

Chapitre 01 : Généralités

Ce chapitre fournit une introduction complète aux généralités du Data Mining, en expliquant le processus KDD, les concepts de base, les techniques courantes, le processus de Data Mining et le prétraitement des données. Il jette les bases nécessaires pour comprendre et appliquer le Data Mining dans des contextes variés.

Chapitre 02 : Etat de l'art

Ce chapitre constitue une étude approfondie sur les travaux antérieurs concernant la prédiction de l'infection au COVID-19 à partir de données cliniques. En analysant les articles sélectionnés, en établissant un tableau comparatif, notre objectif est de fournir un aperçu complet de ce domaine de recherche. Nous nous concentrerons également sur l'identification des algorithmes les plus utilisés ainsi que les métriques d'évaluation prédominantes.

Chapitre 03 : Méthodologie

Ce chapitre présente une méthodologie détaillée pour l'analyse des données du coronavirus, couvrant le prétraitement et le choix des algorithmes. Les résultats obtenus fournissent une base solide pour la poursuite de l'étude et l'interprétation des données.

Chapitre 04 : Réalisation et Résultats

Ce chapitre présente une analyse détaillée des performances d'algorithmes de classification utilisés, ainsi que l'optimisation du meilleur pour améliorer ses performances. Nous comparerons les résultats obtenus avant et après l'optimisation afin de mettre en évidence les gains obtenus. Enfin, nous procéderons à une interprétation des résultats et discuterons de leurs implications pratiques.

Chapitre 1

Généralité

1. Introduction

L'analyse de données est devenue une discipline cruciale dans le contexte du numérique et de l'accumulation massive de données. Parmi les techniques utilisées pour extraire des connaissances à partir de ces données, le data mining occupe une place prépondérante. Il s'agit d'une discipline qui combine des approches statistiques, mathématiques et informatiques pour découvrir des modèles et des connaissances à partir de données massives, structurées ou non.

Le processus de data mining comprend plusieurs étapes essentielles, notamment l'acquisition de données, le nettoyage et la préparation des données, l'exploration des données, la modélisation et l'interprétation des résultats. Chacune de ces étapes joue un rôle crucial dans la réussite de l'analyse des données.

Dans ce mémoire, nous nous concentrons plus particulièrement sur la technique de classification en data mining. La classification consiste à classer des données en différentes catégories ou classes en fonction de leurs caractéristiques. Cette technique trouve de nombreuses applications pratiques, telles que la détection de spam, la reconnaissance de formes, la prédiction de maladies, etc.

Il existe plusieurs algorithmes de classification couramment utilisés en data mining, tels que K-Nearest Neighbors (KNN), les arbres de décision, Random Forest, Support Vector Machines (SVM), et bien d'autres encore. Chaque algorithme présente ses propres caractéristiques et avantages, et leur choix dépend du contexte spécifique et des types de données sur lesquels ils sont appliqués.

Avant de présenter en détail notre étude et les techniques de classification, il est essentiel de comprendre les concepts de base du data mining. Cela comprend le processus KDD (Knowledge Discovery in Data bases), qui définit les étapes clés du data mining, ainsi que les

Chapitre 01 : Généralité

Différents types de données rencontrés dans cette discipline, tels que les données numériques, catégorielles, textuelles, etc. Nous aborderons également les techniques de data mining les

Plus couramment utilisées, telles que l'apprentissage supervisé, l'apprentissage non supervisé et la détection d'anomalies.

Enfin, nous discuterons des critères d'évaluation des modèles de classification en data mining, notamment les métriques d'évaluation et les méthodes de validation croisée. Ces critères permettent d'évaluer la performance des modèles de classification et de s'assurer de leur généralisation aux données non observées.

Cette introduction situe notre travail dans le contexte plus large du data mining, met en évidence l'importance de la classification en data mining et annonce les différentes parties de notre mémoire. Nous espérons que cette étude contribuera à une meilleure compréhension et utilisation des techniques de classification en data mining, et qu'elle ouvrira de nouvelles perspectives pour l'exploitation des connaissances cachées dans les données massives.

2. Processus KDD

Le processus KDD (Knowledge Discovery in data bases), également appelé processus de découverte de connaissances, est une approche systématique pour réaliser le Data Mining de manière efficace et méthodique. Il comprend plusieurs étapes interconnectées qui permettent de passer des données brutes à la découverte de connaissances exploitables. Voici une présentation des différentes étapes du processus KDD [1] :

a. Sélection des données : Cette étape consiste à identifier les sources de données pertinentes pour l'analyse. Il s'agit de déterminer quels ensembles de données sont nécessaires en fonction des objectifs du projet de Data Mining. Les données peuvent provenir de différentes sources telles que des bases de données, des fichiers plats, des flux de données en temps réel, etc.

b. Prétraitement des données : Avant de pouvoir appliquer des techniques de Data Mining, il est souvent nécessaire de prétraiter les données. Cela implique des opérations telles que le nettoyage des données, où les valeurs manquantes ou aberrantes sont traitées, la transformation des données, où les variables peuvent être normalisées ou discrétisées, et la réduction de dimension, où les caractéristiques les plus pertinentes sont sélectionnées.

c. Transformation des données : Cette étape vise à transformer les données prétraitées en une forme appropriée pour le Data Mining. Cela peut impliquer des opérations telles que la réduction de la dimensionnalité, où les caractéristiques sont réduites à un sous-ensemble pertinent, la sélection des attributs, où les attributs les plus informatifs sont choisis, ou encore l'agrégation de données, où les données sont regroupées à un niveau plus élevé de granularité.

d. Data Mining : L'étape centrale du processus KDD consiste à appliquer des algorithmes et des techniques de Data Mining pour extraire des modèles et des connaissances des données transformées. Il existe une variété d'approches et d'algorithmes de Data Mining, tels que l'apprentissage supervisé, l'apprentissage non supervisé, le clustering, la classification, la régression, etc. Ces techniques permettent de découvrir des relations, des tendances et des motifs cachés dans les données.

e. Interprétation et évaluation des résultats : Une fois que les modèles ont été extraits, il est essentiel de les interpréter et de les évaluer pour en tirer des conclusions significatives. Cette étape implique l'analyse des résultats du Data Mining, la compréhension des modèles découverts et l'évaluation de leur qualité et de leur pertinence par rapport aux objectifs du projet. Il est également important de communiquer les résultats de manière claire et compréhensible aux parties prenantes.

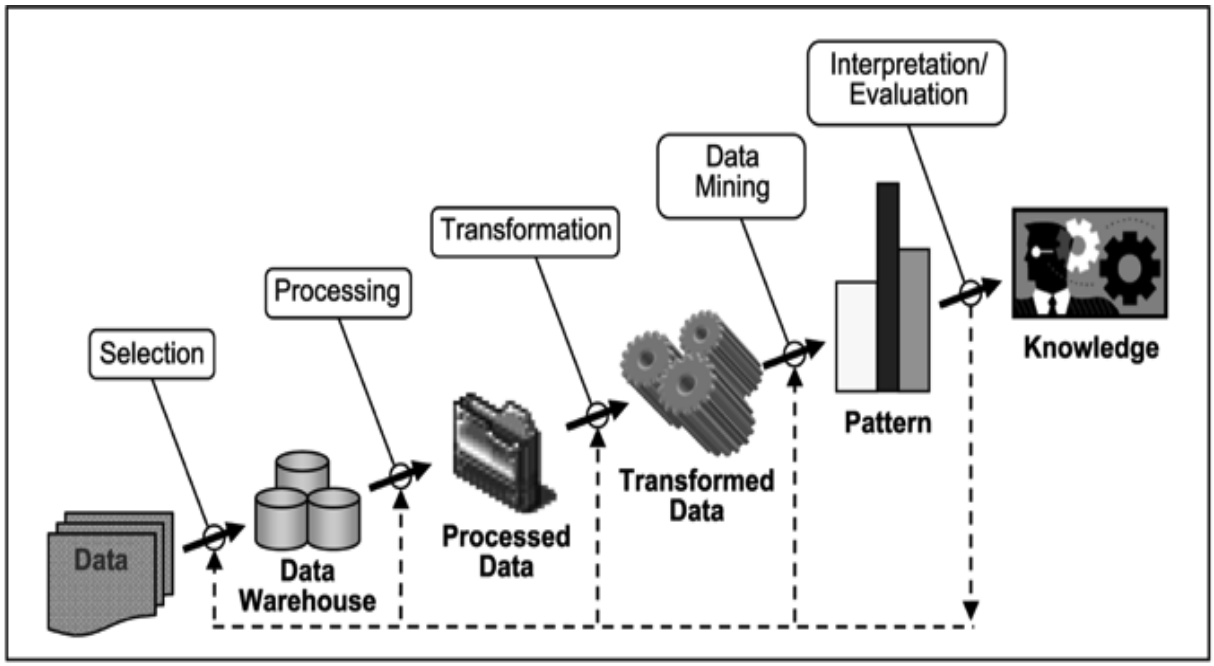

Figure 01 : Processus KDD [2]

Le processus KDD est itératif et cyclique, ce qui signifie que les résultats obtenus peuvent influencer les étapes précédentes. Des ajustements et des itérations peuvent être nécessaires pour améliorer les résultats ou explorer de nouvelles pistes. Cette approche itérative permet une amélioration continue de la qualité des modèles et des connaissances extraites.

3. Définition Datamining

En général, le Data Mining ou la Fouille de données désigne un ensemble de techniques qui permettent d'explorer des données pour en extraire des connaissances sous forme de modèles descriptifs. Ces modèles sont utilisés pour décrire le comportement présent des données et/ou pour prédire leur comportement futur dans le système représenté par ces données.

Le Data Mining se positionne à l'intersection de plusieurs domaines, notamment les bases de données, l'intelligence artificielle, la statistique et l'analyse de données [3] :

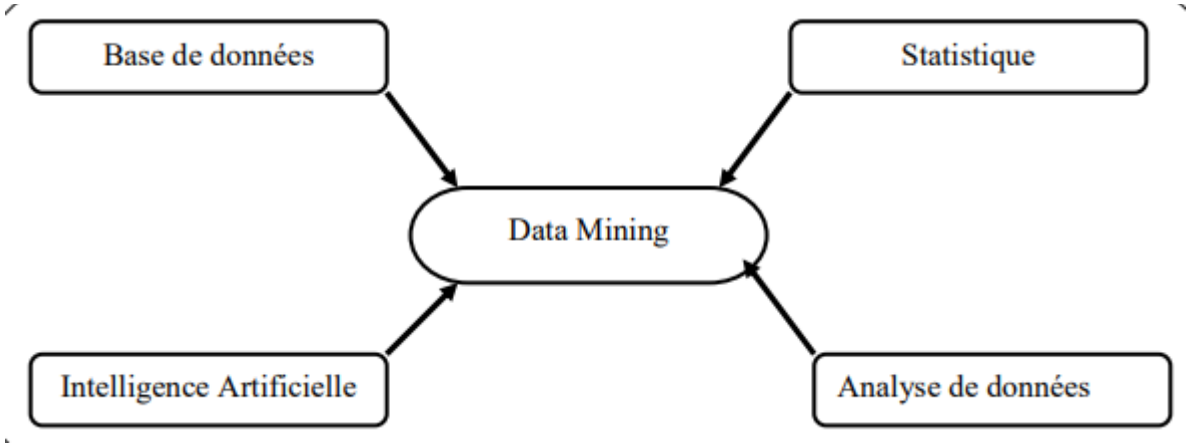

Figure 02 : Techniques Datamining [4]

Le Data Mining requiert une utilisation opérationnelle rapide des résultats d'analyse obtenus, souvent dans des délais très courts. Le processus d'analyse doit permettre à l'organisation de réagir rapidement. Les données traitées proviennent de différents systèmes de stockage en place dans l'organisation, ce qui les rend hétérogènes, multiples et plus ou moins structurées. Le Data Mining vise à extraire des connaissances à partir de vastes volumes de données stockées de manière diverse, notamment dans des bases de données ou dans un (ou plusieurs) entrepôts de données (data Warehouse). Ces données peuvent également être récupérées à partir de sources riches et plus ou moins structurées telles qu'Internet (Web) ou en temps réel (retrait d'argent dans un distributeur de billets...).

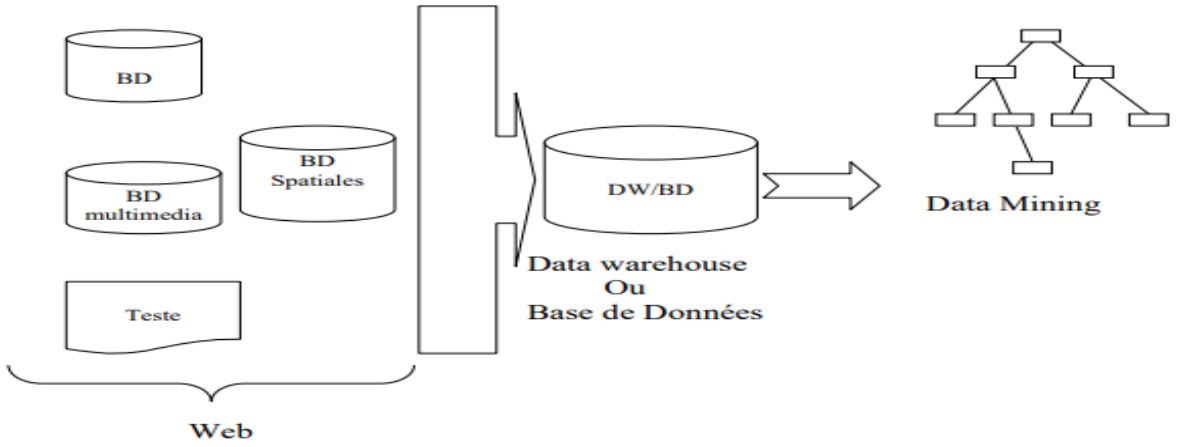

Figure 03 : Architecture Du Datamining [5]

4. Concepts de base du Data Mining

4.1. Caractéristiques des données utilisées en Data Mining

Il existe différents types de données en Data Mining, chacun avec ses propres caractéristiques. Voici une liste des principaux types de données [1] [6] :

1. **Données numériques ou quantitatives :** ce sont des données exprimées sous forme de chiffres, de valeurs continues ou discrètes. Par exemple, la température, la taille, le nombre de ventes, etc.
2. **Données catégorielles :** également connues sous le nom de données nominales, ces données représentent des attributs qui ont des valeurs limitées et ne peuvent pas être mesurées numériquement. Par exemple, le sexe, la couleur des yeux, le type de produit, etc.
3. **Données ordinales :** ces données sont similaires aux données catégorielles, mais les valeurs peuvent être ordonnées ou classées selon une échelle de mesure. Par exemple, l'éducation (niveau primaire, secondaire, universitaire), la taille des entreprises (petite, moyenne, grande), etc.
4. **Données temporelles :** il s'agit de données qui ont une dimension temporelle. Elles sont utilisées pour analyser les tendances, les cycles et les modèles dans le temps. Par exemple, les ventes quotidiennes, les prix des actions, les données climatiques, etc.

5. **Données textuelles :** ce sont des données sous forme de texte brut, telles que les commentaires des clients, les articles de presse, les descriptions de produits, etc. Elles nécessitent un traitement spécial pour être utilisées en Data Mining.

En général, les données en Data Mining peuvent être divisées en deux catégories : les données structurées et les données non structurées. Les données structurées sont stockées dans des bases de données ou des tableurs, tandis que les données non structurées sont stockées sous forme de texte, de fichiers audio, d'images, etc.

4.2. Types de problèmes résolus par le Data Mining

Le Data Mining est utilisé pour résoudre différents types de problèmes en fonction des objectifs et des données disponibles. Voici les principales catégories de problèmes traités en Data Mining [1] [6] :

a. Classification : La classification consiste à attribuer des objets ou des instances à des classes préétablies en fonction de leurs caractéristiques. Par exemple, classifier les emails comme "spam" ou "non spam" en fonction de leur contenu.

b. Régression : La régression vise à prédire une variable continue en fonction d'autres variables. Par exemple, prédire le prix d'une maison en fonction de ses caractéristiques telles que la superficie, le nombre de chambres, etc.

c. Clustering : Le clustering, ou regroupement, consiste à regrouper des objets similaires entre eux en fonction de leurs similitudes. Cela permet d'identifier des structures intrinsèques dans les données et de découvrir des groupes homogènes.

d. Détection d'anomalies : La détection d'anomalies consiste à identifier des observations rares ou inhabituelles qui diffèrent significativement du reste des données. Cela permet de détecter des comportements frauduleux, des erreurs ou des événements inattendus.

e. Association de règles : L'association de règles consiste à découvrir des relations de co-occurrence entre des éléments dans les données. Par exemple, identifier les produits qui sont souvent achetés ensemble dans un supermarché.

Ces différents problèmes résolus par le Data Mining ont des approches et des techniques spécifiques associées à chacun d'entre eux. Le choix de la méthode appropriée dépendra des données, des objectifs et des contraintes du problème à résoudre.

En conclusion, le Data Mining est une discipline qui permet d'extraire des connaissances à partir de grandes quantités de données en utilisant des techniques avancées. Comprendre les caractéristiques des données et les différents types de problèmes résolus par le Data Mining est essentiel pour choisir les méthodes appropriées et obtenir des résultats pertinents.

5. Techniques courantes du Data Mining

5.1. Classification supervisée :

La classification supervisée implique d'attribuer automatiquement une catégorie ou une classe à des données dont on ne connaît pas la catégorie. Pour ce faire, un classifieur, qui est un algorithme de machine learning, est entraîné sur des données similaires ou très proches des données que l'on souhaite classer. Le terme "supervisée" dans la classification supervisée vient du fait que les données d'entraînement ont déjà été triées et classées par des humains. En revanche, lorsque les données d'entraînement ne sont pas triées ou classées, on parle de classification non supervisée [7].

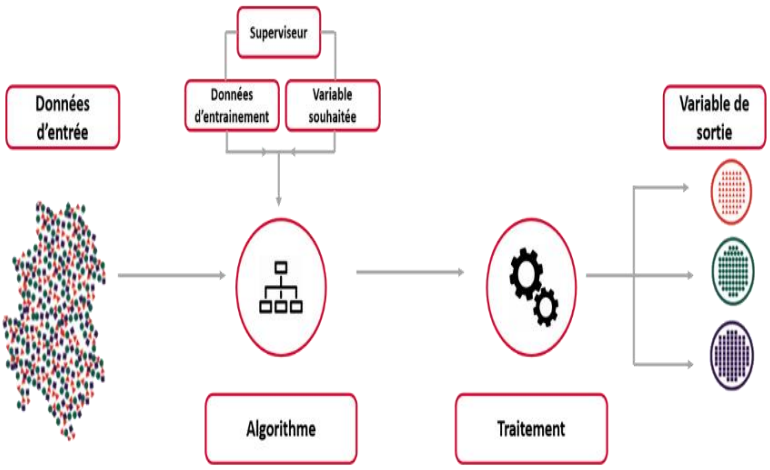

Figure 04 : Classification supervisée [8]

Il existe de nombreuses techniques de classification supervisée, nous pouvons citer [9] :

5.1.1. K plus proches voisins (KNN)

L'algorithme KNN (k-Nearest Neighbors) est un algorithme d'apprentissage automatique supervisé utilisé pour la classification et la régression. Il est basé sur le principe de proximité des exemples d'entraînement pour la prise de décision. L'idée centrale de l'algorithme KNN est

que des exemples similaires sont susceptibles de partager la même étiquette ou valeur cible [10].

Voici les étapes clés de l'algorithme KNN [11] :

- **Représentation des données :** Les exemples d'entraînement sont représentés sous forme de vecteurs dans un espace de caractéristiques, où chaque dimension correspond à une caractéristique ou un attribut de l'exemple.
- **Mesure de similarité :** Pour classer un nouvel exemple, l'algorithme KNN mesure la similarité entre cet exemple et les exemples d'entraînement en utilisant une mesure de distance, généralement la distance euclidienne. La distance euclidienne entre deux exemples x et y est calculée comme suit :

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

  Où n est le nombre de caractéristiques.

- **Sélection des k plus proches voisins :** L'algorithme KNN sélectionne les k exemples d'entraînement les plus proches du nouvel exemple en fonction de leur similarité mesurée. Ces exemples sont appelés les k plus proches voisins.
- **Vote majoritaire :** Une fois les k plus proches voisins identifiés, l'algorithme KNN effectue un vote majoritaire pour déterminer l'étiquette ou la valeur cible du nouvel exemple. Dans le cas de la classification, l'étiquette attribuée sera celle qui est la plus fréquente parmi les k voisins. Dans le cas de la régression, la valeur cible attribuée sera la moyenne des valeurs cibles des k voisins.
- **Classification ou prédiction :** Le nouvel exemple est classé dans la classe correspondant à l'étiquette majoritaire ou est prédit avec la valeur cible attribuée.

Avantages de l'algorithme KNN :

- Simplicité : L'algorithme KNN est simple à comprendre et à implémenter. Il ne nécessite pas d'apprentissage préalable ou de modélisation complexe.
- Adaptabilité : L'algorithme KNN peut s'adapter à des problèmes de classification et de régression, et fonctionne bien pour les ensembles de données avec des frontières de décision complexes.

- Interprétabilité : Les prédictions de l'algorithme KNN peuvent être facilement interprétées, car elles sont basées sur les exemples les plus similaires.

Inconvénients de l'algorithme KNN :

- Sensibilité à la dimensionnalité : L'algorithme KNN peut être sensible à la dimensionnalité des données, car la mesure de distance devient moins discriminante avec un grand nombre de caractéristiques.
- Sensibilité aux valeurs aberrantes : Les valeurs aberrantes peuvent avoir un impact significatif sur la mesure de distance, ce qui peut affecter les prédictions de l'algorithme KNN.
- Coût computationnel élevé : Classer de nouveaux exemples avec l'algorithme KNN peut être coûteux en termes de temps de calcul, en particulier pour de grands ensembles de données.

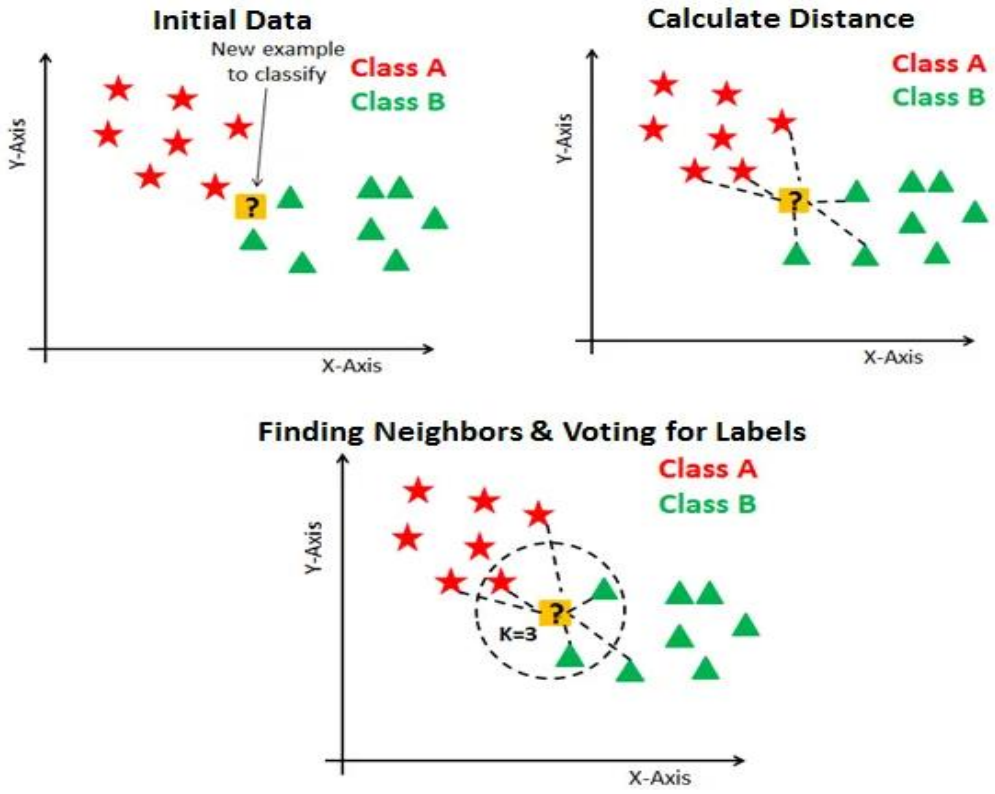

Figure 05 : Méthode K plus proche voisin [12]

L'algorithme KNN est une méthode simple et adaptable pour la classification et la régression. Il utilise la proximité des exemples d'entraînement pour effectuer des prédictions. Cependant, il peut être sensible à la dimensionnalité des données, aux valeurs aberrantes et avoir un coût computationnel élevé.

5.1.2. SVM (Support Vector Machines)

SVM, ou Support Vector Machines, est un algorithme d'apprentissage automatique supervisé utilisé pour la classification et la régression. Il a été développé par Vladimir Vapnik et ses collègues dans les années 1990. L'algorithme SVM est particulièrement réputé pour sa capacité à construire des hyperplans de séparation optimale entre différentes classes, permettant ainsi une classification précise des données [9].

L'idée fondamentale derrière l'algorithme SVM est de trouver un hyperplan dans un espace de grande dimension qui sépare les exemples de différentes classes avec la plus grande marge possible. Voici les étapes clés de l'algorithme [10]:

- **Représentation des données :** Les exemples d'entraînement sont représentés sous forme de vecteurs dans un espace de grande dimension, où chaque dimension correspond à une caractéristique ou un attribut de l'exemple.
- **Sélection de l'hyperplan optimal :** L'objectif de l'algorithme SVM est de trouver l'hyperplan qui sépare les exemples de différentes classes avec la plus grande marge possible. La marge est la distance entre l'hyperplan et les exemples les plus proches de chaque classe, appelés vecteurs de support.
- **Transformation des données :** Si les données ne sont pas linéairement séparables, l'algorithme SVM utilise une technique appelée "kernel trick" pour les projeter dans un espace de dimension supérieure où elles peuvent être séparées linéairement. Les kernels les plus couramment utilisés sont le kernel linéaire, le kernel polynomial et le kernel gaussien.
- **Résolution du problème d'optimisation :** L'algorithme SVM formule le problème de recherche de l'hyperplan optimal comme un problème d'optimisation convexe, où l'objectif est de minimiser une fonction coût tout en maximisant la marge. Des techniques d'optimisation telles que la programmation quadratique sont utilisées pour résoudre ce problème.

- **Classification des nouveaux exemples :** Une fois que l'hyperplan optimal est trouvé, il est utilisé pour classer de nouveaux exemples en les projetant dans l'espace de grande dimension et en les comparant à l'hyperplan de séparation.

Avantages de l'algorithme SVM :

- Grande précision : L'algorithme SVM est connu pour sa précision élevée en classification, en particulier dans les cas où les données sont linéairement ou non linéairement séparables.
- Gestion des données à dimensions élevées : L'algorithme SVM peut gérer des données avec un grand nombre de dimensions sans perdre en performance, grâce à la formulation du problème d'optimisation convexe.
- Robustesse aux données aberrantes : L'utilisation de la marge maximale dans l'algorithme SVM rend l'algorithme plus robuste aux données aberrantes, car il accorde moins d'importance aux exemples qui sont loin de la marge.
- Utilisation de kernels : L'algorithme SVM peut utiliser des kernels pour gérer des problèmes de classification non linéaires, en les projetant dans un espace de dimension supérieure.

Inconvénients de l'algorithme SVM :

- Sensibilité au choix des paramètres : L'algorithme SVM nécessite de sélectionner des paramètres tels que le type de kernel et les paramètres de régularisation. Le choix de ces paramètres peut influencer considérablement les performances de classification.
- Complexité de calcul : La résolution du problème d'optimisation dans l'algorithme SVM peut être coûteuse en termes de calcul, en particulier pour les grands ensembles de données.
- Interprétabilité limitée : L'hyperplan optimal trouvé par l'algorithme SVM peut être difficile à interpréter en termes de signification des caractéristiques ou des attributs.

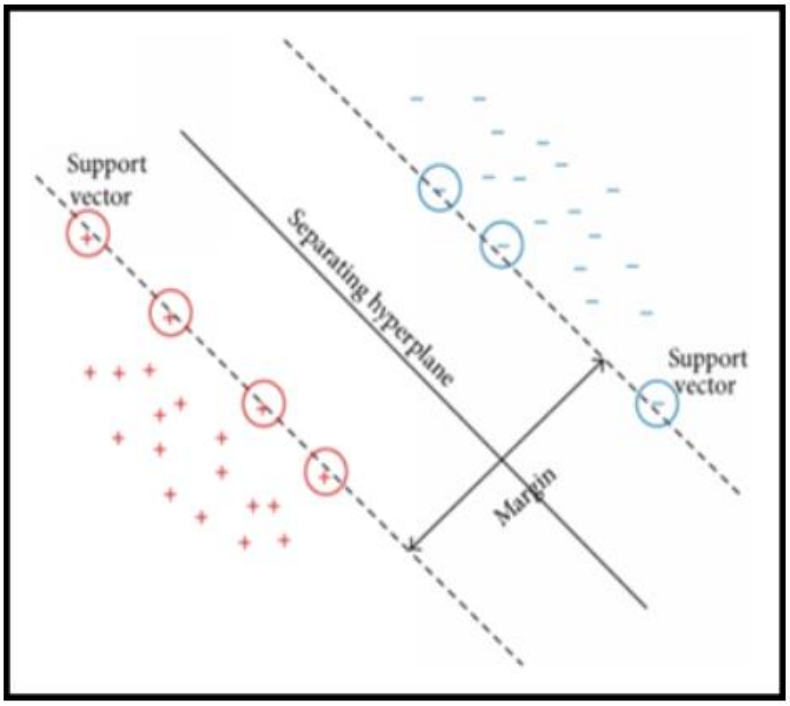

Figure 06 : Méthode SVM (Support Vector Machine) [13]

L'algorithme SVM est une méthode puissante pour la classification et la régression. Il offre une grande précision, une robustesse aux données aberrantes et la possibilité de gérer des problèmes non linéaires grâce à l'utilisation de kernels. Cependant, il peut être sensible au choix des paramètres, avoir une complexité de calcul élevée et une interprétabilité limitée.

5.1.3. ADABOOST

AdaBoost, ou Adaptive Boosting, est un algorithme d'apprentissage automatique supervisé utilisé dans le domaine de la classification. Il a été développé par Yoav Freund et Robert Schapire en 1996. AdaBoost est particulièrement connu pour sa capacité à combiner plusieurs classifieurs faibles pour former un classifieur fort, ce qui améliore les performances de classification.

L'algorithme AdaBoost fonctionne en itérations, en accordant une importance accrue aux exemples mal classés lors de chaque itération. Voici les étapes clés de l'algorithme [14] :

- **Initialisation des poids :** Chaque exemple du data set d'entraînement se voit attribuer un poids initial égal. Ces poids définissent l'importance relative de chaque exemple lors de la classification.

- **Entraînement des classifieurs faibles :** Dans chaque itération, un classifieur faible est entraîné sur le data set d'entraînement. Un classifieur faible est un modèle de classification simple, tel qu'un arbre de décision de faible profondeur ou une règle de décision. Il est important que le classifieur faible soit légèrement meilleur que le hasard.
- **Évaluation du classifieur faible :** Une fois que le classifieur faible est entraîné, il est utilisé pour prédire les étiquettes des exemples du data set d'entraînement. Les exemples mal classés reçoivent un poids plus élevé, tandis que les exemples bien classés reçoivent un poids plus faible.
- **Mise à jour des poids :** Les poids des exemples sont mis à jour en fonction des prédictions du classifieur faible. Les exemples mal classés reçoivent un poids plus élevé, ce qui permet de se concentrer davantage sur ces exemples difficiles à classer. Les exemples bien classés reçoivent un poids plus faible.
- **Combinaison des classifieurs faibles :** Les classifieurs faibles sont combinés pour former un classifieur fort final. Chaque classifieur faible contribue à la décision finale en fonction de sa performance, mesurée par son taux d'erreur pondéré.
- **Répétition des étapes 2 à 5 :** Les étapes d'entraînement des classifieurs faibles, d'évaluation, de mise à jour des poids et de combinaison sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint, tel que le nombre d'itérations prédéfini ou la convergence du taux d'erreur.

Avantages de l'algorithme AdaBoost :

- Grande précision : AdaBoost est capable de fournir des résultats de classification précis, en particulier lorsque les classifieurs faibles sont soigneusement sélectionnés et combinés.
- Gestion des problèmes complexes : L'algorithme AdaBoost peut traiter des problèmes de classification complexes en utilisant des classifieurs faibles relativement simples.
- Résistance au surapprentissage : Grâce à son approche itérative et à la mise à jour des poids, AdaBoost est moins susceptible de surapprendre les données d'entraînement et a une bonne capacité à généraliser sur de nouvelles données.
- Flexibilité : AdaBoost peut être utilisé avec différents types de classifieurs faibles, offrant ainsi une flexibilité dans le choix de l'algorithme de base adapté au problème de classification spécifique.

Inconvénients de l'algorithme AdaBoost :

- Sensibilité aux données aberrantes : Comme AdaBoost accorde une importance accrue aux exemples mal classés, il peut être sensible aux données aberrantes ou aux erreurs d'étiquetage dans le data set d'entraînement.
- Temps d'entraînement plus long : AdaBoost nécessite plusieurs itérations pour entraîner les classifieurs faibles et les combiner. Par conséquent, il peut être plus lent à s'entraîner par rapport à d'autres algorithmes de classification.
- Dépendance à des classifieurs faibles compétents : L'efficacité d'AdaBoost repose sur la sélection de classifieurs faibles légèrement meilleurs que le hasard. Si les classifieurs faibles sont trop faibles, AdaBoost peut ne pas obtenir de bons résultats de classification.

5.1.4. Random Forest

Random Forest (Forêt Aléatoire) est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. Il s'agit d'une méthode d'ensemble qui combine les prédictions de plusieurs arbres de décision individuels pour obtenir une prédiction finale plus robuste et précise.

L'idée principale derrière Random Forest est de construire un grand nombre d'arbres de décision indépendants et diversifiés, appelés "arbres de décision aléatoires". Chaque arbre est formé sur un échantillon aléatoire de données d'entraînement et à chaque étape de construction de l'arbre, une sous-collection aléatoire de variables (caractéristiques) est utilisée pour trouver la meilleure division des données. Cette diversité dans les arbres de décision est obtenue en utilisant des échantillons aléatoires de données et des variables, ce qui réduit le surapprentissage (overfitting) et permet d'obtenir des prédictions plus générales et stables [15].

L'algorithme Random Forest peut être décrit en plusieurs étapes [16] :

- **Construction de l'ensemble de forêts :** Un grand nombre d'arbres de décision aléatoires est créé, généralement des centaines ou même des milliers d'arbres, chacun étant formé sur un échantillon aléatoire de données d'entraînement.
- **Sélection aléatoire des variables :** À chaque nœud de l'arbre, une sous-collection aléatoire de variables est sélectionnée pour la recherche de la meilleure division. Cela garantit la diversité des arbres et réduit la corrélation entre les arbres.

- **Construction des arbres de décision :** Chaque arbre est construit en divisant récursivement les données d'entraînement en sous-groupes homogènes en fonction des valeurs des variables sélectionnées. La division est effectuée en utilisant des critères tels que le gain d'information ou la diminution de l'impureté de Gini.
- **Prédiction :** Une fois que tous les arbres sont construits, la prédiction finale est obtenue en agrégeant les prédictions de chaque arbre. Pour la classification, un vote majoritaire est utilisé pour déterminer la classe prédite, tandis que pour la régression, une moyenne des prédictions est prise.

Les avantages de l'algorithme Random Forest :

- Robustesse aux valeurs aberrantes et aux données manquantes : En utilisant un grand nombre d'arbres, les valeurs aberrantes ou les données manquantes ont moins d'impact sur les prédictions globales.
- Gestion automatique des variables : L'algorithme effectue une sélection aléatoire des variables, ce qui permet d'éviter les problèmes de surapprentissage et de réduire la corrélation entre les arbres.
- Bonnes performances sur des ensembles de données complexes : Random Forest est capable de gérer des ensembles de données de grande dimension avec des interactions complexes entre les variables.

Inconvénients :

- Moins interprétable que les arbres de décision individuels : L'agrégation des prédictions de nombreux arbres rend la compréhension des décisions de l'algorithme plus difficile.
- Sensibilité aux paramètres : Bien que Random Forest soit généralement peu sensible aux paramètres, il est important de régler le nombre d'arbres, la profondeur des arbres et d'autres paramètres pour obtenir de bonnes performances.

Random Forest est un puissant algorithme d'apprentissage ensembliste qui combine les prédictions de multiples arbres de décision pour obtenir des prédictions plus précises et robustes. Il est largement utilisé en raison de sa simplicité d'utilisation, de sa capacité à gérer des ensembles de données complexes et de sa résilience face aux valeurs aberrantes et aux données manquantes. Cependant, il peut être moins interprétable que les arbres de décision individuels et nécessite un réglage approprié des paramètres.

5.1.5. Les réseaux de neurones

Les réseaux de neurones sont des modèles d'apprentissage automatique inspirés du fonctionnement du cerveau humain. Ils sont constitués de neurones artificiels organisés en couches, chacune étant connectée à la précédente. Chaque neurone reçoit des entrées pondérées, effectue un calcul, puis transmet sa sortie à la couche suivante. L'apprentissage consiste à ajuster les poids des connexions entre les neurones pour minimiser une fonction de coût qui mesure l'écart entre les sorties prédites et les sorties attendues. Les réseaux de neurones peuvent être utilisés pour la classification, la régression, la génération de texte, la reconnaissance d'images, la synthèse de la voix et bien d'autres tâches. Ils sont actuellement très populaires en raison de leur grande capacité à modéliser des relations complexes entre les données [3].

5.1.6. Les réseaux bayésiens naïves

La classification bayésienne naïve est une technique de classification probabiliste basée sur le théorème de Bayes avec des hypothèses fortes d'indépendance entre les variables, d'où le terme "naïve". Elle utilise un classifieur bayésien naïf appartenant à la famille des classifieurs linéaires. Ce modèle probabiliste est également appelé "modèle de caractéristiques statistiquement indépendantes". En d'autres termes, ce classifieur suppose que la présence d'une caractéristique pour une classe est indépendante de la présence d'autres caractéristiques. Par exemple, un fruit peut être classé comme une pomme si sa couleur est rouge, sa forme est ronde, et sa taille est d'environ 10 cm, même si ces caractéristiques sont liées dans la réalité. Les classifieurs bayésiens naïfs peuvent être entraînés de manière efficace dans un contexte d'apprentissage supervisé en fonction de la nature de chaque modèle probabiliste.

Le classificateur repose sur le théorème de Bayes, qui permet de calculer les probabilités conditionnelles. Dans un contexte général, ce théorème fournit une méthode de calcul de la probabilité conditionnelle d'une cause sachant la présence d'un effet, à partir de la probabilité conditionnelle de l'effet sachant la présence de la cause, ainsi que des probabilités a priori de la cause et de l'effet [9].

Les avantages de cette méthode sont sa facilité d'implémentation, sa simplicité et sa rapidité, ainsi que ses bons résultats. Toutefois, ses performances sont limitées lorsqu'elle est confrontée à une grande quantité de données à traiter. De plus, le modèle est considéré comme naïf ou simple en raison de l'hypothèse d'indépendance [10].

5.2. Classification non-supervisée

La classification non supervisée, également connue sous le nom de la segmentation (clustering en Anglais), les classes ne sont pas connues a priori. Elle utilise des règles ou des critères de regroupement pour créer des groupes de données qui dépendent des données disponibles à un moment donné. Les classes sont généralement fondées sur la structure des données, ce qui rend plus difficile la détermination de la sémantique associée à chaque classe.

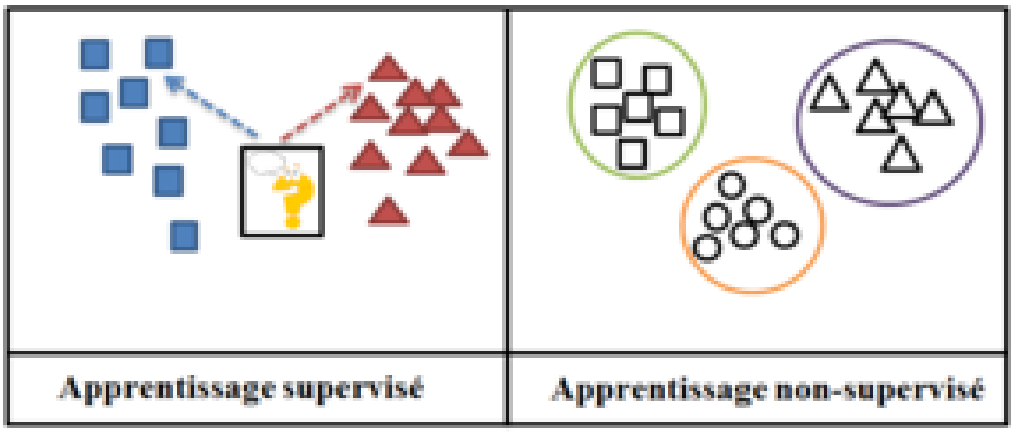

Figure 07 : Apprentissage supervisé et non-supervisé [17]

Il existe aussi de nombreuses techniques pour la classification non-supervisée, nous pouvons citer :

5.2.1. Les méthodes hiérarchiques

Le clustering hiérarchique est une technique d'analyse de données non supervisée qui vise à regrouper des données en clusters (groupes) en fonction de leur similarité. Contrairement à d'autres techniques de clustering, le clustering hiérarchique permet de visualiser les résultats sous forme d'arbre hiérarchique (dendrogramme). Il existe deux types de clustering hiérarchique : le clustering hiérarchique agglomératif (Bottom-up) et le clustering hiérarchique divisif (top-down) [9].

Dans le clustering hiérarchique agglomératif, chaque donnée est initialement considérée comme un cluster. Les clusters les plus proches sont fusionnés pour former un cluster plus grand, et ce processus est répété jusqu'à ce que tous les clusters soient fusionnés en un seul grand cluster. À chaque étape, la distance entre les clusters est calculée en utilisant une mesure de similarité telle que la distance euclidienne ou la corrélation.

Dans le clustering hiérarchique divisif, tous les points sont initialement considérés comme appartenant à un seul cluster. Ce cluster est ensuite divisé en plusieurs sous-clusters en fonction de la distance entre les points, et ce processus est répété jusqu'à ce que chaque cluster ne contienne plus qu'un seul point. Ce type de clustering est généralement moins utilisé car il peut être plus difficile à mettre en œuvre et à interpréter que le clustering hiérarchique agglomératif.

Le clustering hiérarchique est souvent utilisé dans des domaines tels que la biologie, la génétique, l'analyse de marché et l'analyse de données géospatiales. Les avantages de cette méthode incluent sa simplicité, sa facilité d'interprétation et la possibilité de visualiser les résultats sous forme de dendrogramme. Cependant, cette méthode peut être sensible aux choix de mesures de similarité et de seuils de coupure, et peut être coûteuse en termes de temps de calcul pour de grandes quantités de données.

5.2.2. Les méthodes de partitionnement

Les méthodes de partitionnement sont une classe d'algorithmes de clustering qui visent à diviser un ensemble de données en plusieurs groupes ou partitions. Chaque partition est constituée d'un sous-ensemble d'objets de données similaires, tandis que les objets de données de partitions différentes sont considérés comme étant différents [9].

Les méthodes de partitionnement les plus courantes sont l'algorithme de K-means et l'algorithme de clustering hiérarchique. L'algorithme de K-means est un algorithme de partitionnement qui consiste à trouver K clusters en minimisant la distance intra-cluster et en maximisant la distance inter-cluster. K est défini à l'avance par l'utilisateur. L'algorithme commence par choisir K centres de cluster initiaux au hasard et attribue chaque point de données au centre de cluster le plus proche. Ensuite, les centres de cluster sont mis à jour et les points de données sont à nouveau affectés aux centres de cluster les plus proches. Le processus se répète jusqu'à ce qu'il n'y ait plus de changements dans l'affectation des points de données.

Les méthodes de partitionnement sont couramment utilisées dans divers domaines tels que la segmentation de marché, la classification d'image, la reconnaissance de forme, la biologie, l'analyse de données génomiques, etc. Ces méthodes sont relativement simples à implémenter et peuvent fournir des résultats de clustering efficaces et utiles. Cependant, elles peuvent être sensibles aux centres de cluster initiaux et ne conviennent pas à tous les types de données [18].

5.2.3. Les méthodes basées sur la densité

Les méthodes basées sur la densité sont des techniques de clustering qui se fondent sur l'hypothèse que les clusters sont des régions de haute densité de données. Elles identifient les clusters en recherchant des régions dans l'espace de données où la densité de points est élevée et qui sont séparées par des régions de faible densité [9].

La méthode de clustering basée sur la densité la plus connue est DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Cette méthode consiste à identifier des "points centraux" dans des régions de haute densité de données, des "points frontières" qui se trouvent entre les clusters et des "points de bruit" qui ne sont pas associés à des clusters. Elle utilise deux paramètres : un rayon de recherche autour de chaque point et un nombre minimal de points requis dans ce rayon pour considérer un point comme un point central.

Une autre méthode basée sur la densité est OPTICS (Ordering Points To Identify the Clustering Structure). Cette méthode permet de détecter des clusters de densités différentes en construisant une structure de hiérarchie à partir des distances entre les points et en utilisant un paramètre appelé "epsilon" pour définir la distance maximale entre les points d'un même cluster [18].

Ces méthodes ont l'avantage de pouvoir détecter des clusters de formes et de tailles variées, et de pouvoir gérer des données bruyantes ou ayant des densités différentes. Toutefois, elles sont sensibles aux paramètres choisis et peuvent être coûteuses en temps de calcul pour des jeux de données volumineux.

5.2.4. Les méthodes basées sur les grilles

Les méthodes basées sur les grilles, ou grilles de partitionnement, sont des algorithmes de clustering qui découpent l'espace de données en une grille multidimensionnelle régulière, et affectent chaque point de données à la cellule de la grille correspondante. Cette approche peut être utilisée pour des données de toutes dimensions, mais elle est particulièrement utile pour des données à haute dimensionnalité.

Le processus de partitionnement consiste à créer une grille de cellules, qui peut être uniforme ou adaptative, en fonction de la densité des points de données. Chaque point de données est affecté à la cellule de la grille correspondante, en fonction de sa position dans l'espace de données. Les grilles uniformes sont souvent utilisées pour des données à faible dimensionnalité,

tandis que les grilles adaptatives sont plus appropriées pour des données à haute dimensionnalité.

Les méthodes basées sur les grilles sont relativement simples et efficaces pour les grandes bases de données. Cependant, elles peuvent être sensibles à la taille de la grille, qui peut avoir un impact important sur la qualité du clustering. Si la taille de la grille est trop grande, les cellules de la grille peuvent ne pas être suffisamment peuplées, ce qui peut conduire à des erreurs de classification. D'autre part, si la taille de la grille est trop petite, les cellules peuvent être surpeuplées, ce qui peut également affecter négativement la qualité du clustering [9].

6. Processus de Data Mining

Le processus de Data Mining suit généralement plusieurs étapes pour extraire des connaissances à partir des données. Voici un aperçu des étapes clés du processus de Data Mining [19]:

1. **Collecte et préparation des données** : cette étape implique la collecte de données brutes à partir de différentes sources, y compris des fichiers, des bases de données, des flux de données en temps réel, etc. Les données sont préparées pour l'analyse en les nettoyant, en les intégrant, en les transformant et en les prétraitant.
2. **Nettoyage et prétraitement des données :** cette étape implique la correction des erreurs, la suppression des données manquantes, la normalisation des données et la transformation des données brutes en un format adapté pour l'analyse.
3. **Réduction et transformation des données :** cette étape implique la réduction de la dimensionnalité des données pour faciliter l'analyse en supprimant les caractéristiques non pertinentes. Cette étape peut également inclure la transformation des données en utilisant des techniques telles que la réduction de la variance, l'analyse en composantes principales et la transformation en ondelettes.
4. **Construction et évaluation du modèle :** cette étape implique la sélection et la construction d'un modèle de Data Mining approprié pour résoudre le problème d'analyse de données. Les modèles peuvent être construits en utilisant des techniques telles que la classification, la régression, la détection d'anomalies, le clustering et les réseaux neuronaux. Les modèles sont évalués en utilisant des métriques de performance telles que la précision, le rappel et la F-mesure.

5. **Interprétation et déploiement :** cette étape implique l'interprétation des résultats de Data Mining pour extraire des connaissances utiles et les communiquer aux parties prenantes. Les résultats peuvent être déployés en utilisant des outils tels que des tableaux de bord, des rapports et des applications de visualisation de données.

7. Datamining et l'apprentissage automatique :

Le Data Mining et l'apprentissage automatique sont des méthodes utilisées pour comprendre et tirer des informations à partir des données. Le Data Mining cherche à découvrir des modèles et des relations intéressantes dans les données, tandis que l'apprentissage automatique permet aux ordinateurs d'apprendre à partir des données pour faire des prédictions ou prendre des décisions. En d'autres termes, le Data Mining explore les données pour trouver des choses intéressantes, tandis que l'apprentissage automatique apprend à partir des données pour accomplir des tâches spécifiques. Ces deux domaines sont souvent utilisés ensemble pour analyser les données et obtenir des connaissances utiles [43].

8. Prétraitement de données

8.1. Définition et compréhension du problème

La définition et la compréhension du problème sont essentielles dans la plupart des cas, afin de comprendre le sens des données et le domaine à explorer. Sans cette compréhension, aucun algorithme ne peut fournir des résultats fiables. En effet, en comprenant le problème, nous sommes en mesure de préparer les données nécessaires à l'exploration et d'interpréter correctement les résultats obtenus. Le datamining est généralement réalisé dans des domaines spécifiques tels que les banques, la médecine, la biologie, le marketing, où la connaissance et l'expérience dans le domaine jouent un rôle crucial dans la définition du problème, l'orientation de l'exploration et l'explication des résultats. Une bonne compréhension du problème implique l'évaluation des résultats de l'exploration, ainsi que la justification de son coût éventuel. En d'autres termes, il est important d'être en mesure d'évaluer les résultats obtenus et de convaincre l'utilisateur de leur rentabilité.

8.2. Collecte des données

Au cours de cette étape, notre attention se porte sur la manière dont les données sont générées et collectées. En se référant à la définition du problème et aux objectifs du datamining, nous

pouvons avoir une idée des données à utiliser. Cependant, ces données ne sont pas toujours présentées dans le même format ni avec la même structure. Elles peuvent prendre la forme de textes, de bases de données, de pages web, et ainsi de suite. Parfois, il est nécessaire de faire une copie d'un système d'information en cours d'exécution afin de collecter les données à partir de sources potentiellement hétérogènes telles que des fichiers, des bases de données relationnelles, temporelles, etc. [18].

8.3. Préparation et prétraitement des données

Lors de la préparation et du prétraitement des données, l'objectif est de garantir la qualité et la cohérence de l'ensemble des données.

Une étape essentielle consiste à évaluer la qualité des données avant de les traiter. Il est fréquent de trouver des données manquantes, aberrantes ou en double. Les données manquantes peuvent résulter d'erreurs de saisie ou de lacunes dans les informations collectées. Si ces valeurs ne sont pas correctement gérées, elles peuvent entraîner des modèles inexacts et des conclusions erronées. Le traitement des valeurs manquantes et aberrantes représente donc un défi, tant sur le plan méthodologique que computationnel.

Le nettoyage des données vise à éliminer les informations non pertinentes, incorrectes ou susceptibles de causer des erreurs lors de l'analyse et de la modélisation ultérieures.

La phase de préparation des données peut également inclure des manipulations, des modifications ou la création de nouvelles variables à partir des données disponibles.

8.3.1. Valeurs manquantes

Une problématique courante concerne les valeurs manquantes, qui peuvent être observées, perdues ou incohérentes [6].

Les Méthodes d'imputation :

L'imputation des valeurs manquantes est une approche souvent utilisée. Elle consiste à remplacer ces valeurs par des estimations appropriées afin de minimiser les biais. Deux méthodes courantes sont :

- L'imputation par la moyenne ou le mode : pour les variables quantitatives, les valeurs manquantes sont remplacées par la moyenne ou la médiane des valeurs existantes. Pour les variables qualitatives, les valeurs manquantes sont remplacées par le mode.
- L'imputation par le ratio ou la régression : cette méthode utilise des modèles statistiques pour prédire les valeurs manquantes en se basant sur des variables auxiliaires.

8.3.2. Valeurs aberrantes

Un autre aspect important concerne les valeurs aberrantes, qui se distinguent considérablement du reste des observations. Avant de procéder à l'imputation des valeurs manquantes, il est crucial d'identifier et de traiter les valeurs aberrantes de manière appropriée.

Il existe plusieurs approches pour gérer les valeurs aberrantes, notamment [6]:

- Vérification des erreurs de saisie : si une valeur aberrante est due à une erreur de saisie, il est recommandé de corriger la source de l'erreur, soit en se référant à la source originale des données, soit en appliquant des méthodes d'imputation appropriées.
- Conservation des valeurs aberrantes : si la valeur aberrante est valide et ne résulte pas d'une erreur de collecte de données, il est souvent préférable de la conserver pour éviter de fausser les analyses ultérieures.
- Normalisation des variables : une technique courante pour traiter les valeurs aberrantes consiste à effectuer une analyse en composantes principales (ACP) afin de réduire la dimensionnalité des données et de réduire l'impact des valeurs extrêmes.

8.3.3. La Normalisation avec RobustScaler

La normalisation des variables est une étape essentielle du prétraitement des données, car elle permet de mettre toutes les variables à la même échelle. Cela est particulièrement important dans les cas où les variables ont des plages de valeurs très différentes. L'une des techniques couramment utilisées pour la normalisation des variables est le RobustScaler [21].

Le RobustScaler est une méthode de normalisation robuste aux valeurs aberrantes (outliers) dans les données. Il est basé sur la médiane et l'écart interquartile (IQR) plutôt que sur la moyenne et l'écart-type, ce qui le rend plus résistant aux valeurs extrêmes. Voici les étapes du processus de normalisation des variables avec RobustScaler [22] .

1. Calcul de la médiane (M) et de l'écart interquartile (IQR) pour chaque variable. La médiane représente la valeur centrale de la distribution, tandis que l'IQR mesure la dispersion des valeurs autour de la médiane.
2. Soustraction de la médiane à chaque valeur de la variable pour centrer la distribution autour de zéro. Cette étape permet d'éliminer le biais potentiel dans les données.
3. Division de chaque valeur par l'écart interquartile (IQR) pour réduire l'amplitude des valeurs. Cela permet de mettre les variables à la même échelle, en les ramenant à une plage commune.

Avantages :

- **Robustesse aux valeurs aberrantes :** Étant basé sur la médiane et l'écart interquartile, le RobustScaler est plus résistant aux valeurs extrêmes qui pourraient avoir un impact disproportionné sur la normalisation des variables.
- **Conservation des informations :** Contrairement à la normalisation basée sur la moyenne et l'écart-type, le RobustScaler ne modifie pas la distribution des variables. Il conserve donc les informations relatives à la forme et à la dispersion des données.
- **Préservation des relations ordinales :** Si vos variables contiennent des informations ordinales, telles que des classements ou des échelles d'évaluation, le RobustScaler maintient ces relations ordinales intactes.

Inconvénients :

- **Sensibilité à la présence de valeurs aberrantes extrêmes :** Bien que le RobustScaler soit conçu pour être robuste aux valeurs aberrantes, des valeurs extrêmes extrêmement éloignées de la médiane peuvent encore influencer la normalisation des variables.
- **Perte d'interprétabilité :** Lorsque vous utilisez le RobustScaler, les valeurs normalisées n'ont plus la même signification que les valeurs d'origine. Par conséquent, l'interprétation des coefficients ou des résultats basés sur les variables normalisées peut être plus complexe.

RobustScaler est une méthode de normalisation des variables qui est robuste aux valeurs aberrantes et qui permet de mettre les variables à la même échelle. Il offre des avantages tels que la préservation des informations, la robustesse et la conservation des relations ordinales.

Cependant, il peut être sensible aux valeurs aberrantes extrêmes et peut entraîner une perte d'interprétabilité.

8.3.4. Le transformateur PolynomialFeatures

Le transformateur PolynomialFeatures est une technique de prétraitement des données qui permet de générer des combinaisons polynomiales des variables d'entrée. Il s'agit d'une transformation non linéaire qui étend l'espace des fonctionnalités en ajoutant des termes polynomiaux aux variables existantes.

Le processus de transformation des variables avec PolynomialFeatures se déroule en plusieurs étapes [23] :

1. Entrée des variables : Les variables d'entrée sont représentées sous forme d'une matrice X, où chaque ligne correspond à une observation et chaque colonne correspond à une variable.

2. Sélection du degré : Vous devez spécifier le degré du polynôme que vous souhaitez générer. Le degré détermine le nombre maximum de termes polynomiaux à inclure. Par exemple, pour un degré de 2, les combinaisons polynomiales incluront des termes de degré 0, 1 et 2.

3. Génération des termes polynomiaux : Le transformateur PolynomialFeatures génère toutes les combinaisons possibles des variables d'entrée jusqu'au degré spécifié. Par exemple, si vous avez deux variables d'entrée, X1 et X2, et un degré de 2, les termes polynomiaux générés seront X1, X2, X1^2, X1*X2 et X2^2.

4. Ajout des termes polynomiaux : Les termes polynomiaux générés sont ajoutés à la matrice d'entrée X, créant ainsi une nouvelle matrice X_poly. Chaque terme polynômial est ajouté en tant que nouvelle colonne dans X_poly.

Avantages :

- Capture des relations non linéaires : En ajoutant des termes polynomiaux aux variables d'entrée, PolynomialFeatures permet de capturer les relations non linéaires entre les variables. Cela peut être utile lorsque les données présentent des relations complexes qui ne peuvent pas être capturées par une simple régression linéaire.

- Expansion de l'espace des fonctionnalités : La génération de termes polynomiaux étend l'espace des fonctionnalités, ce qui peut permettre de modéliser des phénomènes plus complexes. Cela peut améliorer la capacité du modèle à capturer les motifs sous-jacents dans les données.

Inconvénients :

- Augmentation de la dimensionnalité : L'ajout de termes polynomiaux peut considérablement augmenter la dimensionnalité de l'espace des fonctionnalités, ce qui peut entraîner une augmentation du nombre de variables et de la complexité du modèle. Cela peut rendre le modèle plus susceptible de surajuster les données, en particulier lorsque le nombre d'observations est limité.
- Risque de surajustement : L'ajout de termes polynomiaux peut introduire une complexité supplémentaire dans le modèle, ce qui peut le rendre plus sensible au bruit et aux fluctuations aléatoires des données. Il est donc important de surveiller attentivement les performances du modèle et de régulariser si nécessaire pour éviter le surajustement.

Le transformateur PolynomialFeatures est une technique de prétraitement des données qui génère des termes polynomiaux à partir des variables d'entrée. Il permet de capturer des relations non linéaires et d'élargir l'espace des fonctionnalités. Cependant, il peut augmenter la dimensionnalité et le risque de surajustement, nécessitant une gestion appropriée lors de l'utilisation dans la construction du modèle [24].

8.3.5. SelectKbest

Le SelectKBest est une méthode de sélection de caractéristiques qui vise à identifier les K meilleures caractéristiques (variables) parmi un ensemble plus large. Cette technique est utilisée dans le prétraitement avancé des données pour réduire la dimensionnalité et améliorer la performance des modèles d'apprentissage automatique [25].

Voici une description détaillée du processus de SelectKBest [26]:

- **Calcul de la mesure de qualité des caractéristiques :** Le SelectKBest utilise une mesure de qualité spécifique pour évaluer chaque caractéristique et déterminer son importance. Cette mesure peut être basée sur des statistiques telles que l'analyse de

variance (ANOVA), le score du chi-carré, l'information mutuelle, ou d'autres métriques appropriées en fonction du type de données et de la tâche d'apprentissage.

- **Classement des caractéristiques :** Les caractéristiques sont classées en fonction de leur mesure de qualité, de la plus élevée à la plus faible. Cela permet de déterminer leur ordre d'importance et de sélectionner les meilleures caractéristiques.
- **Sélection des K meilleures caractéristiques :** Les K meilleures caractéristiques sont sélectionnées en fonction de leur classement. Ces caractéristiques sont considérées comme les plus informatives et les plus pertinentes pour la tâche d'apprentissage.

Avantages :

- Réduction de la dimensionnalité : En sélectionnant les meilleures caractéristiques, le SelectKBest permet de réduire le nombre de variables, ce qui peut améliorer l'efficacité du modèle en réduisant les temps de calcul et en évitant les problèmes de surajustement (overfitting) liés à une trop grande dimensionnalité.
- Amélioration de la performance : En sélectionnant les caractéristiques les plus informatives, le SelectKBest permet de concentrer l'attention du modèle sur les aspects les plus pertinents des données, ce qui peut conduire à de meilleures performances de prédiction ou de classification.
- Interprétation simplifiée : En utilisant un sous-ensemble plus restreint de caractéristiques, le modèle devient plus facile à interpréter, car il est plus facile d'identifier les caractéristiques les plus importantes et leur relation avec la variable cible.

Inconvénients :

- Perte d'information : En sélectionnant uniquement un sous-ensemble de caractéristiques, il est possible de perdre des informations potentiellement utiles présentes dans les caractéristiques non sélectionnées. Cela peut être préjudiciable si ces caractéristiques contiennent des informations pertinentes pour la tâche d'apprentissage.
- Dépendance à la méthode de mesure de qualité : Le choix de la mesure de qualité utilisée par le SelectKBest peut influencer les résultats de la sélection des caractéristiques. Il est donc important de choisir une mesure adaptée à la nature des données et à la tâche d'apprentissage.

Le SelectKBest est une méthode de sélection de caractéristiques qui identifie les K meilleures caractéristiques en utilisant une mesure de qualité appropriée. Il offre des avantages tels que la réduction de la dimensionnalité, l'amélioration de la performance et la simplification de l'interprétation. Cependant, il peut entraîner une perte d'information et dépend de la méthode de mesure de qualité choisie.

8.3.6. Feature engineering

Le feature engineering (ingénierie des caractéristiques) est une étape cruciale dans le processus d'analyse de données et de construction de modèles prédictifs. Il consiste à créer de nouvelles caractéristiques à partir des données existantes, en utilisant des connaissances du domaine, des transformations mathématiques ou des techniques statistiques. L'objectif principal du feature engineering est d'améliorer la représentation des données pour aider les algorithmes de machine learning à mieux comprendre les relations sous-jacentes et à améliorer les performances prédictives des modèles.

Voici quelques techniques couramment utilisées en feature engineering [27] :

- **Sélection des caractéristiques :** Il s'agit de sélectionner les caractéristiques les plus informatives et pertinentes pour le problème spécifique que vous essayez de résoudre. Cela peut être fait en utilisant des méthodes statistiques telles que l'analyse de corrélation, les tests de significativité, ou des techniques plus avancées telles que la régression régularisée (Lasso, Ridge) ou les arbres de décision.
- **Extraction de caractéristiques :** Cette technique consiste à extraire de nouvelles caractéristiques à partir des données brutes en utilisant des méthodes mathématiques ou statistiques. Par exemple, vous pouvez extraire des statistiques récapitulatives telles que la moyenne, la variance, le maximum, le minimum, ou des caractéristiques plus complexes telles que les transformées de Fourier, les transformations en ondelettes, les histogrammes, etc.
- **Création de variables indicatrices :** Si vous avez des variables catégorielles, vous pouvez les transformer en variables indicatrices (ou variables "one-hot encoded") pour les rendre exploitables par les algorithmes de machine learning. Cela implique de créer une nouvelle variable binaire pour chaque catégorie, où la valeur sera 1 si l'observation appartient à cette catégorie, et 0 sinon.

- **Transformation des variables :** Cette technique vise à transformer les variables existantes pour les rendre plus conformes aux hypothèses des modèles de machine learning. Par exemple, vous pouvez appliquer une transformation logarithmique ou une transformation de Box-Cox pour stabiliser la variance, ou utiliser des transformations non linéaires pour capturer des relations complexes entre les variables.
- **Normalisation/Standardisation :** Il s'agit de mettre les variables à la même échelle pour éviter que certaines variables ne dominent les autres. La normalisation met les variables dans une plage spécifique, tandis que la standardisation les met à une échelle standard avec une moyenne de 0 et un écart type de 1.

Avantages :

- Amélioration des performances des modèles : En créant de nouvelles caractéristiques ou en transformant les caractéristiques existantes, le feature engineering permet de mieux représenter les relations sous-jacentes des données. Cela peut améliorer les performances des modèles de machine learning en leur permettant de capturer des motifs plus complexes et de prendre des décisions plus précises.
- Réduction de la dimensionnalité : En sélectionnant les caractéristiques les plus informatives, le feature engineering peut réduire la dimensionnalité des données. Cela peut faciliter le processus d'apprentissage en éliminant le bruit et en améliorant l'efficacité des algorithmes.
- Gestion des données manquantes ou aberrantes : Le feature engineering peut inclure des techniques de gestion des données manquantes ou aberrantes, telles que l'imputation des valeurs manquantes ou le traitement des valeurs aberrantes. Cela permet de préparer les données pour l'analyse et de garantir la cohérence des résultats.
- Adaptation aux algorithmes spécifiques : Le feature engineering permet d'adapter les caractéristiques aux exigences spécifiques des algorithmes de machine learning. Par exemple, certains algorithmes peuvent nécessiter des variables normalisées ou des variables indicatrices pour fonctionner correctement. Le feature engineering permet de préparer les données de manière appropriée pour chaque algorithme.

Inconvénients :

- Complexité et coût en temps : Le feature engineering peut être un processus complexe et intensif en temps, en particulier lorsque les données sont volumineuses ou complexes. Il peut nécessiter des connaissances spécialisées et une exploration approfondie des données. Cela peut augmenter la complexité de la modélisation et nécessiter des ressources supplémentaires.
- Risque de surajustement : Lors de la création de nouvelles caractéristiques, il existe un risque de surajustement des modèles aux données d'entraînement. Si les caractéristiques sont trop spécifiques ou ne généralisent pas bien sur de nouvelles données, cela peut entraîner une baisse des performances lors de la prédiction.
- Sensibilité aux erreurs dans le processus de feature engineering : Si des erreurs sont commises lors de la création ou de la transformation des caractéristiques, cela peut entraîner des distorsions ou des biais dans les données. Il est donc important de valider et de vérifier les étapes de feature engineering pour s'assurer de leur qualité.
- Dépendance des connaissances domaines : Le feature engineering peut nécessiter une bonne compréhension du domaine d'application pour prendre des décisions appropriées sur la création et la transformation des caractéristiques. Cela peut limiter son applicabilité à des domaines où les connaissances sont limitées ou difficiles à obtenir.

Le feature engineering est une étape essentielle pour améliorer les performances des modèles de machine learning en adaptant les caractéristiques aux exigences spécifiques des algorithmes et en permettant une meilleure représentation des données. Cependant, il peut également être complexe et nécessiter des ressources supplémentaires, tout en étant sensible aux erreurs et aux risques de surajustement. Une approche prudente et rigoureuse est donc nécessaire pour tirer pleinement parti des avantages du feature engineering tout en atténuant ses inconvénients potentiels [28].

8.3.7. Sélection des caractéristiques

La sélection des caractéristiques, également connue sous le nom de sélection des variables ou de l'attribut, est une étape importante du prétraitement des données qui vise à identifier les variables les plus pertinentes et informatives pour un problème de modélisation donné. L'objectif est de réduire la dimensionnalité des données en éliminant les caractéristiques

redondantes ou peu significatives, ce qui peut améliorer les performances du modèle, réduire les temps de calcul et faciliter l'interprétation des résultats [29].

Il existe plusieurs approches pour la sélection des caractéristiques, voici quelques-unes des méthodes les plus couramment utilisées [30]:

Méthodes basées sur les filtres : Ces méthodes utilisent des mesures statistiques ou des techniques de corrélation pour évaluer l'importance des caractéristiques de manière indépendante de tout algorithme de modélisation. Elles filtrent les caractéristiques en se basant sur des critères tels que la corrélation, l'information mutuelle, l'ANOVA, le gain d'information, etc. Les caractéristiques sont classées selon leur pertinence et un seuil est appliqué pour sélectionner les caractéristiques les plus importantes.

Méthodes basées sur les enveloppes : Ces méthodes utilisent des techniques d'apprentissage automatique pour évaluer l'importance des caractéristiques en fonction de leur capacité à améliorer la performance d'un modèle spécifique. Elles utilisent des algorithmes d'apprentissage itératifs, tels que Forward Sélection, Backward Elimination ou Recursive Feature Elimination, pour sélectionner les caractéristiques qui maximisent la performance du modèle.

Méthodes basées sur l'incorporation : Ces méthodes combinent les étapes de sélection des caractéristiques et d'apprentissage du modèle en un seul processus. Elles utilisent des algorithmes d'apprentissage automatique, tels que les arbres de décision, les SVM ou les réseaux neuronaux, qui sont capables de sélectionner automatiquement les caractéristiques les plus importantes tout en construisant le modèle. Ces méthodes sont souvent plus complexes et nécessitent plus de ressources computationnelles, mais elles peuvent donner de meilleurs résultats lorsque la relation entre les caractéristiques et la variable cible est complexe.

Avantages :

- Réduction de la dimensionnalité des données, ce qui peut améliorer les performances des modèles et réduire les temps de calcul.
- Élimination des caractéristiques redondantes ou peu informatives, ce qui facilite l'interprétation des résultats.

- Réduction du risque de surajustement (overfitting) en éliminant les caractéristiques non pertinentes.

Inconvénients :

- Perte potentielle d'informations si des caractéristiques pertinentes sont éliminées de manière incorrecte.
- Sensibilité aux choix de méthode et de paramètres, ce qui peut entraîner une sélection biaisée ou non optimale.

Augmentation du risque de biais de sélection si la sélection est basée sur les mêmes données utilisées pour l'apprentissage du modèle.

8.4. Estimation du modèle

Dans cette étape, il est essentiel de sélectionner la technique appropriée pour extraire les connaissances (exploration) à partir des données. Des méthodes telles que les arbres de décision, la régression logistique, etc., sont utilisées. En général, l'implémentation repose sur plusieurs de ces techniques, puis le résultat le plus adéquat est choisi [1].

Méthodes d'échantillonnage

À partir d'un jeu de données initial, on crée un échantillon d'entraînement sur lequel le modèle sera construit, ainsi qu'un échantillon de test pour évaluer le modèle. En pratique, il est courant de réserver 80 % des données pour l'échantillon d'entraînement et 20 % pour l'échantillon de test. Cette séparation permet d'expérimenter différents choix de variables et paramètres du modèle sur l'échantillon d'apprentissage afin de déterminer le modèle optimal [1].

8.5. Évaluation et interprétation du modèle

Il est relativement facile de construire un modèle qui donne de bons résultats avec les données utilisées pour son estimation. Cependant, il est plus difficile de s'assurer que le modèle peut généraliser et prédire de manière satisfaisante de nouvelles observations non utilisées lors de son apprentissage. Pour établir un juste équilibre entre l'apprentissage du modèle et sa capacité prédictive, il est essentiel de mettre en place une évaluation globale de la qualité du modèle.

En d'autres termes, les modèles extraits, en particulier par les méthodes d'apprentissage supervisé, ne peuvent pas être directement utilisés avec fiabilité. Ils doivent être évalués en

confrontant leurs prédictions à la réalité et en évaluant leur exactitude. La méthode habituelle consiste à estimer le taux d'erreur du modèle, permettant ainsi à l'utilisateur de décider d'utiliser ou non le modèle de prédiction en connaissance des risques encourus [18].

8.5.1. Matrice de confusion

Une matrice de confusion fournit des informations sur les classifications réelles et prédites effectuées par un système de classification. Les performances de tels systèmes sont généralement évaluées à l'aide des données de la matrice.

Voici un exemple de matrice de confusion pour un classifieur à deux classes [18] :

Tableau 01 : Matrice de confusion :

		Valeur prédite	
		Négative	Positive
Valeur Réel	Négative	VN	FP
	Positive	FN	VP

Les entrées de la matrice de confusion ont les significations suivantes :

- VN est le nombre de prédictions correctes où une instance est négative (vrais négatifs).
- FP est le nombre de prédictions incorrectes où une instance est prédite positive (faux positifs).
- FN est le nombre de prédictions incorrectes où une instance est prédite négative (faux négatifs).
- VP est le nombre de prédictions correctes où une instance est positive (vrais positifs).

Différents termes standard ont été définis pour la matrice de confusion à deux classes :

(a) La précision totale (PT) est la proportion du nombre total de prédictions correctes, calculée avec l'équation :

$$PT = (VN + VP) / (VN + FP + FN + VP)$$

(b) L'erreur est la proportion du nombre total de prédictions incorrectes, calculée avec l'équation

$$\text{Taux d'erreur} = (FP + FN) / (VN + FP + FN + VP)$$

(c) Le rappel, également appelé taux de vrais positifs ou sensibilité, est la proportion de cas positifs correctement identifiés, calculée avec l'équation :

$$\text{Rappel} = VP / (FN + VP)$$

(d) La spécificité, également appelée taux de vrais négatifs, est la proportion de cas négatifs correctement classés, calculée avec l'équation :

$$\text{Spécificité} = VN / (VN + FP)$$

(e) Le taux de faux positifs est la proportion de cas négatifs incorrectement classés comme positifs, calculée avec l'équation :

$$\text{Taux de faux positifs} = FP / (VN + FP)$$

(f) Le taux de faux négatifs est la proportion de cas positifs incorrectement classés comme négatifs, calculée avec l'équation :

$$\text{Taux de faux négatifs} = FN / (FN + VP)$$

(g) Enfin, la précision est la proportion de cas positifs prédits correctement, calculée avec l'équation :

$$\text{Précision} = VP / (FP + VP)$$

8.5.2. Courbe d'apprentissage

La courbe d'apprentissage est un outil utilisé en apprentissage automatique pour évaluer la performance d'un modèle en fonction de la quantité de données d'entraînement utilisée. Elle représente l'erreur du modèle en fonction du nombre d'exemples d'entraînement. Au début de l'apprentissage, le modèle peut avoir une performance faible, mais celle-ci s'améliore généralement à mesure que la quantité de données augmente. La courbe d'apprentissage peut prendre différentes formes, telles que convergence rapide, sous-apprentissage ou sur-apprentissage. Elle permet de diagnostiquer les problèmes du modèle et de prendre des décisions pour améliorer sa performance [43].

8.5.3. La validation croisée

La validation croisée consiste à diviser l'échantillon original en k échantillons, puis à sélectionner à tour de rôle chacun des k échantillons comme ensemble de validation, tandis que les k-1 autres échantillons sont utilisés pour l'apprentissage du modèle. On calcule l'erreur quadratique moyenne pour chaque échantillon de validation et on répète l'opération k fois jusqu'à ce que chaque sous-échantillon ait été utilisé exactement une fois comme ensemble de validation. Enfin, la moyenne des k erreurs quadratiques moyennes est calculée pour estimer l'erreur de prédiction [20].

8.5.4. La courbe ROC

La courbe ROC est un outil qui permet d'examiner les performances des classificateurs. Elle est représentée par un graphique qui met en relation le taux de faux positifs sur l'axe des x et le taux de vrais positifs sur l'axe des y. Le point (0,1) correspond à un classificateur parfait, qui classe correctement tous les cas positifs et négatifs, tandis que le point (0,0) représente un classificateur qui prédit que tous les cas sont négatifs. Le point (1,1) correspond à un classificateur qui prédit que chaque cas est positif, et le point (1,0) est le classificateur incorrect pour toutes les classifications [18] [1].

9. Conclusion

Dans ce chapitre, nous avons abordé les concepts fondamentaux du data mining, en mettant l'accent sur la classification en tant que technique clé de cette discipline. Nous avons exploré le processus KDD (Knowledge Discovery in Databases) qui guide les différentes étapes du data mining, de l'acquisition des données à l'interprétation des résultats.

Nous avons également examiné les différents types de données utilisées en data mining, tels que les données numériques, catégorielles et textuelles, ainsi que les problèmes couramment résolus par le data mining, tels que la classification, la régression, le clustering et la détection d'anomalies.

En termes de techniques courantes du data mining, nous avons présenté l'apprentissage supervisé et non supervisé, ainsi que des méthodes de réduction de dimension et d'association de règles. Nous avons également passé en revue des algorithmes de classification populaires

tels que K-Nearest Neighbors (KNN), les arbres de décision, Random Forest et Support Vector Machines (SVM).

Nous avons souligné l'importance de la préparation des données et du prétraitement dans le processus de data mining, en mettant en évidence des techniques telles que la gestion des valeurs manquantes et des valeurs aberrantes, ainsi que l'estimation et l'évaluation des modèles.

Enfin, nous avons souligné que le data mining offre de nombreux avantages, notamment l'identification de modèles cachés, l'amélioration des performances des modèles prédictifs et la prise de décisions éclairées basées sur les connaissances extraites des données. Cependant, nous avons également noté qu'il existe des défis à relever, tels que la sélection appropriée des techniques et des algorithmes, ainsi que la gestion de problèmes tels que le surapprentissage (overfitting) et le sous-apprentissage (underfitting).

Ce chapitre constitue une base solide pour la suite de notre mémoire, où nous allons approfondir l'étude de la classification en data mining en nous concentrant sur des cas d'utilisation spécifiques et en expérimentant différentes approches et algorithmes. Nous espérons que cette exploration approfondie des concepts de base du data mining a permis de mieux comprendre l'importance et les possibilités offertes par cette discipline passionnante.

Chapitre 2

Etat de L’art

1. Etat de l’art

Ci-dessous une étude sur les travaux antérieures qui caractérise et vise à comparer et à optimiser les algorithmes de classification pour la prédiction de l'infection au COVID-19 à partir de données cliniques.

1.1. Travaux connexes

Article 1

[Yuri Kravchenko, Nataliia Dakhno, Olga Leshchenko, Anastasiia Tolstokorova]

Cet article se concentre sur l'utilisation d'algorithmes d'apprentissage automatique pour prédire les résultats de l'infection par le COVID-19. Les auteurs ont analysé des données provenant de patients atteints du COVID-19 dans le monde entier en prenant en compte différentes variables cliniques telles que l'âge, les maladies chroniques, les symptômes, etc. Ils ont développé un classificateur binaire pour prédire si un patient est susceptible de mourir ou non de la maladie. Les résultats obtenus ont montré que différents algorithmes de classification, tels que la régression logistique, l'algorithme des k-plus proches voisins, les arbres de décision, la méthode des vecteurs de référence et le classifieur bayésien naïf, peuvent être utilisés avec succès pour prédire les résultats de l'infection au COVID-19 [31].

Article 2

[Khadijeh Moulaei, Mostafa Shanbehzadeh, Zahra Mohammadi-Taghiabad, Hadi Kazemi-Arpanahi]

Cet article met l'accent sur l'identification des prédicteurs les plus pertinents de la mortalité liée au COVID-19 parmi les patients hospitalisés. Les chercheurs ont identifié des caractéristiques cliniques pertinentes à partir d'une revue de littérature et d'un processus Delphi impliquant des experts. Ils ont utilisé des données cliniques de patients hospitalisés pour COVID-19, en éliminant les valeurs manquantes et aberrantes, et ont appliqué une technique de sur-échantillonnage pour équilibrer les classes de résultats. En utilisant Sept algorithmes

d'apprentissage automatique, tels que l'arbre de décision J48, la forêt aléatoire, le k-plus proches voisins, le perceptron multicouche, Naïve Bayes, le boosting extrême de gradient et la régression logistique, ils ont développé des modèles de prédiction de la mortalité. Les résultats ont montré que l'algorithme de boosting extrême de gradient (XGBoost) a obtenu les meilleures performances en termes d'exactitude, de sensibilité et de courbe ROC pour prédire la mortalité liée au COVID-19 [32].

Article 3

[L. J. Muhammad · Md. Milon Islam · Sani Sharif Usman · Safal Islam Ayon]

Cet article se concentre sur l'utilisation de techniques de data mining pour prédire la récupération des patients atteints de COVID-19. Les données épidémiologiques des patients en Corée du Sud ont été utilisées pour développer des modèles prédictifs en utilisant des algorithmes tels que l'arbre de décision, la machine à vecteurs de support, le naïf de Bayes, la régression logistique, la forêt aléatoire et le plus proche voisin. Les résultats ont montré que le modèle développé avec l'algorithme de l'arbre de décision était le plus efficace, avec une précision globale de 99,85%. Ce modèle a pu prédire le nombre de jours nécessaires à la récupération des patients, ainsi que les groupes d'âge présentant un risque élevé de ne pas se rétablir ou de se rétablir rapidement. L'article souligne également l'importance de l'évaluation de la précision des modèles de data mining, et conclut que le modèle basé sur l'algorithme de l'arbre de décision est capable de prédire efficacement la possibilité de récupération des patients infectés par la pandémie de COVID-19 [33].

1.2. Tableau Comparatif

Nous avons établi un tableau comparatif des différentes méthodes utilisées dans chacun des travaux lus. Ce tableau résume les résultats obtenus des approches proposées ainsi que le principal travail menant à l’implémentation des composants des systèmes de recommandations. La Table (01) figurée ci-dessous est constituée de 9 lignes comportant les caractéristiques suivantes :

- 1ere ligne : “Article“ représente le numéro de l’article
- 2eme ligne : “catégorie“ représente le type de prédiction pour laquelle appartiennent ces approches.
- 3eme ligne : ”Approche” représente l’approche elle-même (l’auteur ou les auteurs avec l’année d’´Edition).
- 4eme ligne : ”Titre” c’est le titre de l’article en question.
- 5eme ligne ” Data source” indique les données en entrée.
- 6eme ligne : ” Méthodes utilisées” : explicite toutes les méthodes de classification utilisées.
- 7eme ligne : ”Métriques d’évaluation” : indique les métriques d’évaluation utilisé
- 8eme ligne : ”output” indique les/le résultat de l’approche.
- 9eme ligne ” Outils Utilisée” : indique le nom de l’outil/logiciel utilisé dans le cas où l’approche a été implémentée.

Chapitre 02 : Etat de l'art

Tableau 01 : Tableau Comparatif

Article	Article 01	Article 02	Article 03
Catégorie	Prédiction de l'infection par le COVID-19	Prédiction de la mortalité liée au COVID-19 parmi les patients hospitalisés	Prédire la récupération des patients atteints de COVID-19
Approche	Yuri Kravchenko, Nataliia Dakhno, Olga Leshchenko, Anastasiia Tolstokorova	Khadijeh Moulaei, Mostafa Shanbehzadeh, Zahra Mohammadi Taghiabad, Hadi Kazemi Arpanahi. 04/01/2022	L. J. Muhammad, Md. Milon Islam, Sani Sharif Usman, Safal Islam Ayon 21/06/2020
Titre	Machine Learning Algorithms for Predicting the Results of COVID-19 Coronavirus Infection	Comparing machine learning algorithms for predicting COVID-19 mortality	Predictive Data Mining Models for Novel Coronavirus (COVID 19) Infected Patients' Recovery.
Data source	Ensemble de données des patients hospitalisés confirmés en laboratoire atteints de la COVID-19	Un ensemble de données de patients hospitalisés confirmés en laboratoire atteints de la COVID-19.	Ensemble de données du Korea Centers for Disease Control & Prevention (KCDC)
Méthode utilisée	Régression logistique, k-plus proches voisins, arbres de décision, vecteurs de référence, classifieur bayésien naïf	L'arbre de décision, Random Forest, k-plus proches voisins, perceptron multicouche, Naïve Bayes, le boosting extrême de gradient et la régression logistique	L'arbre de décision, la machine à vecteurs de support, le naïf de Bayes, la régression logistique, la forêt aléatoire et le plus proche voisin
Métrique d'évaluation	Matrice de confusion, Exactitude, Précision, Rappel, la mesure F	Courbe ROC, Exactitude, Précision, Sensibilité, Spécificité	Exactitude
Output	L'arbre de décision est le meilleur pour prédire l'infection	XGBoost est le meilleure pour prédire la mortalité	L'arbre de décision est le meilleur pour la récupération des patients infectés
Outils utilisé	Python, Jupiter	Python, Weka (v3.9.2), The SPSS,	Python

1.3. Discussion et comparaison

L’article 1 de Yuri Kravchenko, Nataliia Dakhno, Olga Leshchenko, Anastasiia Tolstokorova met l'accent sur l'utilisation d'algorithmes d'apprentissage automatique pour prédire les résultats de l'infection par le COVID-19, Cela pourrait être étendue à d'autres maladies pour aider le système de soins de santé à réagir de manière plus efficace à une épidémie ou une pandémie.

L’article 2 de Khadijeh Moulaei, Mostafa Shanbehzadeh, Zahra Mohammadi Taghiabad, Hadi Kazemi Arpanahi met l’accent sur l'identification des prédicteurs les plus pertinents de la mortalité liée au COVID-19 parmi les patients hospitalisés, le modèle proposé est efficace pour prédire le risque de mortalité chez les patients hospitalisés atteints de COVID-19 et optimiser l'utilisation des ressources limitées des hôpitaux. Ce modèle a la capacité d'identifier automatiquement les patients à haut risque dès leur admission ou pendant leur séjour à l'hôpital.

L’article 3 de L. J. Muhammad, Md. Milon Islam, Sani Sharif Usman, Safal Islam Ayon met l’accent sur l'utilisation de techniques de data mining pour prédire la récupération des patients atteints de COVID-19. Les modèles développés seraient très utiles dans le domaine de la santé pour lutter contre le COVID-19.

Les jeux de données utilisées dans ces recherches sont différés :

- L’article 1 a utilisé un ensemble de données des patients infecté par le COVID-19
- L’article 2 a utilisé un ensemble de données des patients hospitalisés confirmés en laboratoire atteints de COVID-19
- L’article 3 a utilisé un ensemble de données du Korea Centers for Disease Control & Prevention (KCDC)

Les métriques d'évaluation utilisées sont différentes :

- L’article 1 a utilisé : Matrice de confusion, Exactitude, Précision, Rappel, la mesure F La mesure F= (2 précision * rappel) / (précision +rappel)
- L’article 2 a utilisé : Courbe ROC, Exactitude, Précision, Sensibilité, Spécificité.
- L’article 3 s'est limité à l'exactitude.

Ces recherches démontrent l'efficacité des modèles de data mining et d'apprentissage automatique pour prédire l’infection par le COVID-19. Les algorithmes tels que les arbres de décision, les machines à vecteurs de support, les naïves Bayes, la régression logistique, les forêts aléatoires et les k-plus proches voisins ont été appliqués sur les ensembles de données, et les résultats ont montré de bonne précisons. Cette étude ouvre également la voie à l'application de l'apprentissage automatique dans d'autres domaines de la santé pour aider à la gestion des épidémies ou des pandémies, ces modèles peuvent fournir une classification précise

1.4. Conclusion

Les modèles de datamining et d'apprentissage automatique ont démontré leur utilité et leur efficacité dans la prédiction et la prise de décision en matière de santé, en particulier lorsqu'il s'agit de la gestion de la pandémie de COVID-19. Ces modèles ont été capables de traiter de grandes quantités de données, d'extraire des schémas cachés et de fournir des prédictions précises sur la récupération des patients, la mortalité, l’infection et d'autres facteurs liés à la maladie.

Chapitre 3

Méthodologie

1. Introduction

Le chapitre de la méthodologie joue un rôle crucial dans notre étude, car il décrit en détail les différentes étapes et procédures utilisées pour atteindre nos objectifs de recherche. Dans ce chapitre, nous présenterons la description du data set de coronavirus que nous avons utilisé, ainsi que les étapes de prétraitement des données que nous avons suivies. De plus, nous expliquerons en détail le choix des algorithmes de classification et leur mise en œuvre sur les données prétraitées. Nous discuterons également l'évaluation des performances des algorithmes et de l'exploration des techniques d'optimisation pour ajuster les hyperparamètres du meilleur algorithme sélectionné, à savoir SVC.

2. Description du data set de corona virus utilisé

Le jeu de données "COVID-19 Diagnosis and Clinical Spectrum" contient des informations cliniques et diagnostiques sur les patients atteints du COVID-19. Il a été recueilli dans le cadre d'études et de recherches visant à mieux comprendre la maladie, son évolution et ses caractéristiques cliniques.

Le jeu de données comprend un total de plusieurs colonnes, chacune fournissant une mesure ou une caractéristique spécifique liée au COVID-19 et à son spectre clinique. Voici une description détaillée des principales colonnes [34] :

Chapitre 03 : Méthodologie

Tableau 01 : description détaillée des principales colonnes

COLONNE	DESCRIPTION
Patient ID	Identifiant unique du patient
Patient Age quantile	Quantile d'âge du patient
SARS-Cov-2 exam result	Résultat de l'examen de dépistage du virus SARS-CoV-2 (positif ou négatif)
Patient admitted to regular ward (1=yes, 0=no)	Indique si le patient a été admis dans une unité de soins régulière (1=oui, 0=non)
Patient admitted to semi-intensive unit (1=yes, 0=no)	Indique si le patient a été admis dans une unité semi-intensive (1=oui, 0=non)
Patient admitted to intensive care unit (1=yes, 0=no)	Indique si le patient a été admis en unité de soins intensifs (1=oui, 0=non)
Hematocrit	Mesure de l'hématocrite, qui est le volume de globules rouges dans le sang
Hemoglobin	Mesure de l'hémoglobine, qui transporte l'oxygène dans le sang
Platelets	Nombre de plaquettes sanguines
Meanplatelet volume	Volume moyen des plaquettes sanguines
Red bloodCells	Nombre de globules rouges
Lymphocytes	Nombre de lymphocytes, un type de globules blancs
Meancorpuscularhemoglobin concentration (MCHC)	Concentration moyenne d'hémoglobine corpusculaire dans les globules rouges
Leukocytes	Nombre total de globules blancs
Basophils	Nombre de basophiles, un type de globules blancs
Meancorpuscularhemoglobin (MCH)	Quantité moyenne d'hémoglobine dans les globules rouges
Eosinophils	Nombre d'éosinophiles, un type de globules blancs
Meancorpuscular volume (MCV)	Volume moyen des globules rouges
Monocytes	Nombre de monocytes, un type de globules blancs

COLONNE	DESCRIPTION
Red bloodcell distribution width (RDW)	Indice de distribution de la largeur des globules rouges
Serum Glucose	Niveau de glucose dans le sérum sanguin
Respiratory Syncytial Virus	Présence ou absence du virus respiratoire syncytial
Influenza A	Présence ou absence du virus de la grippe A
Influenza B	Présence ou absence du virus de la grippe B
Parainfluenza 1	Présence ou absence du parainfluenza 1
Coronavirus NL63	Présence ou absence du coronavirus NL63
Rhinovirus/Entérovirus	Présence ou absence du rhinovirus/entérovirus
Mycoplasma pneumoniae	Présence ou absence de Mycoplasma pneumoniae, une bactérie responsable de la pneumonie
Coronavirus HKU1	Présence ou absence du coronavirus HKU1
Parainfluenza 3	Présence ou absence du parainfluenza 3
Chlamydophila pneumoniae	Présence ou absence de Chlamydophila pneumoniae, une bactérie responsable de la pneumonie
Adenovirus	Présence ou absence de l'adénovirus
Parainfluenza 4	Présence ou absence du parainfluenza 4
Coronavirus 229^{E}	Présence ou absence du coronavirus 229E
Coronavirus OC43	Présence ou absence du coronavirus OC43
Inf A H1N1 2009	Présence ou absence du virus de la grippe A H1N1 de 2009
Bordetella pertussis	Présence ou absence de Bordetella pertussis, la bactérie responsable de la coqueluche
Metapneumovirus	Présence ou absence du metapneumovirus
Parainfluenza 2	Présence ou absence du parainfluenza 2
Neutrophils	Nombre de neutrophiles, un type de globules blancs
Urea	Niveau d'urée dans le sang, un indicateur de la fonction rénale
Proteina C reativa mg/dL	Proteina C reativa mg/dL
Creatinine	Niveau de créatinine dans le sang, un indicateur de la fonction rénale
Potassium	Niveau de potassium dans le sang

COLONNE	DESCRIPTION
Sodium	Niveau de sodium dans le sang
Alanine transaminase	Niveau d'alanine aminotransférase (ALAT), une enzyme présente dans le foie
Aspartate transaminase	Niveau d'aspartate aminotransférase (ASAT), une enzyme présente dans le foie
Gamma-glutamyltransferase	Niveau de gamma-glutamyltransférase (GGT), une enzyme présente dans le foie
Total Bilirubin	Bilirubine totale dans le sang
Direct Bilirubin	Bilirubine directe dans le sang
Indirect Bilirubin	Bilirubine indirecte dans le sang
Alkaline phosphatase	Phosphatase alcaline dans le sang, une enzyme présente dans le foie et les os
Ionized calcium	Calcium ionisé dans le sang
Strepto A	Présence ou absence du streptocoque A
Magnesium	Niveau de magnésium dans le sang
pCO2 (venousbloodgasanalysis)	Pression partielle de dioxyde de carbone (CO2) dans le sang veineux
Hb saturation (venousbloodgasanalysis)	Saturation de l'hémoglobine en oxygène dans le sang veineux
Base excess (venousbloodgasanalysis)	Excès de base dans l'analyse du gaz sanguin veineux
pO2 (venousbloodgasanalysis)	Pression partielle d'oxygène (O2) dans le sang veineux
Fio2 (venousbloodgasanalysis)	Fraction inspirée d'oxygène (FiO2) dans l'analyse du gaz sanguin veineux
Rods #	Nombre de bâtonnets dans un frottis sanguin
Total CO2 (venousbloodgasanalysis)	CO2 total dans l'analyse du gaz sanguin veineux
PH (venousbloodgasanalysis)	PH dans l'analyse du gaz sanguin veineux

Chapitre 03 : Méthodologie

COLONNE	DESCRIPTION
HCO3 (venousbloodgasanalysis)	Bicarbonate (HCO3) dans l'analyse du gaz sanguin veineux
Segmented	Nombre de neutrophiles segmentés dans un frottis sanguin
Promyelocytes	Nombre de promyélocytes dans un frottis sanguin
Metamyelocytes	Nombre de métamyélocytes dans un frottis sanguin
Myelocytes	Nombre de myélocytes dans un frottis sanguin
Myeloblasts	Nombre de myéloblastes dans un frottis sanguin
Urine – Esterase	Présence ou absence d'estérase dans l'urine
Urine – Aspect	Aspect de l'urine (clair, trouble, etc.)
Urine – pH	pH de l'urine
Urine – Hemoglobin	Présence ou absence d'hémoglobine dans l'urine
Urine - Bile pigments	Présence ou absence de pigments biliaires dans l'urine
Urine - Ketone Bodies	Présence ou absence de corps cétoniques dans l'urine
Urine – Nitrite	Présence ou absence de nitrites dans l'urine
Urine – Density	Densité de l'urine
Urine – Urobilinogen	Présence ou absence d'urobilinogène dans l'urine
Urine – Protein	Présence ou absence de protéines dans l'urine
Urine – Sugar	Présence ou absence de sucre dans l'urine
Urine – Leukocytes	Présence ou absence de globules blancs dans l'urine
Urine – Crystals	Présence ou absence de cristaux dans l'urine
Urine - Red bloodcells	Présence ou absence de globules rouges dans l'urine
Urine - Hyaline cylinders	Présence ou absence de cylindres hyalins dans l'urine
Urine – Granularcylinders	Présence ou absence de cylindres granulaires dans l'urine
Urine – Yeasts	Présence ou absence de levures dans l'urine
Urine – Color	Couleur de l'urine
Partial thromboplastin time (PTT)	Temps de céphaline activée (TCA), un test de coagulation sanguine
Relationship (Patient/Normal)	Relation entre le patient et un individu témoin
International normalized ratio (INR)	Rapport international normalisé (INR), un indicateur de la coagulation sanguine

COLONNE	DESCRIPTION
LacticDehydrogenase	Lactate déshydrogénase (LDH), une enzyme présente dans divers tissus
Prothrombin time (PT) Activity	Activité du temps de prothrombine (TP), un test de coagulation sanguine
Vitamin B12	Niveau de vitamine B12 dans le sang
Creatine phosphokinase (CPK)	Créatine phosphokinase (CPK), une enzyme présente dans les muscles
Ferritin	Ferritine, une protéine de stockage du fer
ArterialLactic Acid	Acide lactique artériel, un indicateur du métabolisme anaérobie
Lipase dosage	Dosage de la lipase, une enzyme digestive
D-Dimer	D-dimères, un produit de dégradation de la fibrine dans le sang
Albumin	Niveau d'albumine dans le sang, une protéine produite par le foie
Hb saturation (arterialbloodgases)	Saturation de l'hémoglobine en oxygène dans le sang artériel
pCO2 (arterialbloodgasanalysis)	Pression partielle de dioxyde de carbone (CO2) dans le sang artériel
Base excess (arterialbloodgasanalysis)	Excès de base dans l'analyse du gaz sanguin artériel
pH (arterialbloodgasanalysis)	pH dans l'analyse du gaz sanguin artériel
Total CO2 (arterialbloodgasanalysis)	CO2 total dans l'analyse du gaz sanguin artériel
HCO3 (arterialbloodgasanalysis)	Bicarbonate (HCO3) dans l'analyse du gaz sanguin artériel
pO2 (arterialbloodgasanalysis)	Pression partielle d'oxygène (O2) dans le sang artériel
Arterial Fio2	Fraction inspirée d'oxygène (FiO2) dans l'analyse du gaz sanguin artériel

COLONNE	DESCRIPTION
Phosphor	Niveau de phosphore dans le sang
ctO2 (arterialbloodgasanalysis)	Contenu en oxygène total (ctO2) dans l'analyse du gaz sanguin artériel

Ce jeu de données offre une vaste gamme d'informations cliniques et diagnostiques sur les patients atteints du COVID-19. Il peut être utilisé pour analyser les caractéristiques cliniques, prédire les résultats ou identifier les facteurs de risque associés à la maladie. Cependant, il est important de noter que chaque colonne peut avoir des valeurs manquantes ou des variations dans les méthodes de collecte des données, il est donc essentiel de prendre en compte ces aspects lors de l'analyse des résultats.

3. Prétraitement des données

3.1. Sélectionner les colonnes appropriées à inclure dans notre étude

Pour garantir la qualité et la pertinence des données utilisées dans notre étude, une étape cruciale a été de sélectionner les colonnes appropriées à inclure dans notre analyse. Nous avons effectué cette sélection en nous basant sur le taux de valeurs manquantes dans chaque colonne

Chapitre 03 : Méthodologie

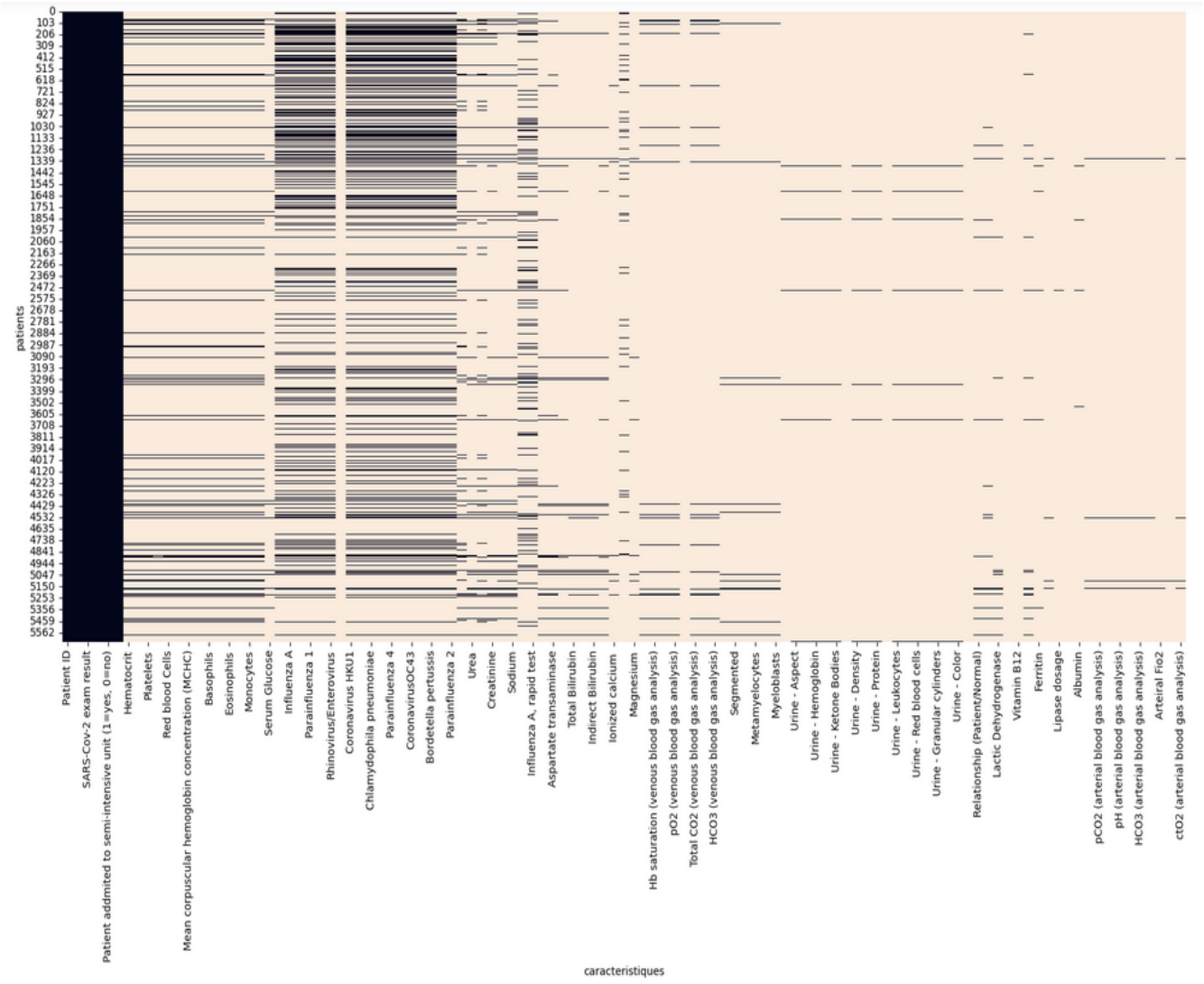

Figure 01 : Heatmap des valeurs manquantes de data set COVID-19

Chaque cellule du heatmap représente une valeur dans le Data Frame, où les cellules blanches indiquent des valeurs manquantes et les cellules non-blanches indiquent des valeurs présentes

En utilisant la formule **missing_rate = df. isna(). sum()/df.shape[0]** [44], nous avons calculé le taux de valeurs manquantes pour chaque colonne du jeu de données. Nous avons ensuite appliqué des critères pour choisir les colonnes à inclure dans notre analyse.

Plus précisément, nous avons sélectionné les colonnes satisfaisant les critères suivants :

- Pour les colonnes de données sanguines, nous avons inclus celles dont le taux de valeurs manquantes était inférieur à 90% (missing_rate < 0.9) et supérieur à 88% (missing_rate >

0.88), Nous avons ignoré les colonnes dont le taux de valeurs manquantes est supérieur à 90 %, car elles en contiennent trop.

- Pour les colonnes de données virales, nous avons inclus celles dont le taux de valeurs manquantes était inférieur à 80% (missing_rate < 0.8) et supérieur à 75% (missing_rate > 0.75).

En utilisant ces critères, nous avons pu sélectionner les colonnes de données sanguines et virales les plus informatives et les moins sujettes aux valeurs manquantes excessives.

En plus de ces colonnes, nous avons également inclus les colonnes clés telles que "Patient, âge quantile" et "SARS-Cov-2 exam result". Ces colonnes sont essentielles pour notre analyse, car elles fournissent des informations cruciales sur l'âge du patient et les résultats des tests de dépistage du COVID-19.

Cette sélection de colonnes nous permettra de réaliser une analyse plus précise et pertinente, en se concentrant sur les attributs les plus informatifs et les moins impactés par les valeurs manquantes.

3.2. Identification des variables catégorielles et encodage

L'encodage des variables catégorielles est une étape essentielle dans notre processus d'analyse des données. Les variables catégorielles représentent des informations qualitatives ou des étiquettes dans notre ensemble de données. Cependant, de nombreux algorithmes de classification nécessitent que les données d'entrée soient numériques pour effectuer les calculs et les prédictions.

```
SARS-Cov-2 exam result---------------------------- ['negative' 'positive']
Respiratory Syncytial Virus----------------------- [nan 'not_detected' 'detected']
Influenza A--------------------------------------- [nan 'not_detected' 'detected']
Influenza B--------------------------------------- [nan 'not_detected' 'detected']
Parainfluenza 1----------------------------------- [nan 'not_detected' 'detected']
CoronavirusNL63----------------------------------- [nan 'not_detected' 'detected']
Rhinovirus/Enterovirus---------------------------- [nan 'detected' 'not_detected']
Coronavirus HKU1---------------------------------- [nan 'not_detected' 'detected']
Parainfluenza 3----------------------------------- [nan 'not_detected' 'detected']
Chlamydophila pneumoniae-------------------------- [nan 'not_detected' 'detected']
Adenovirus---------------------------------------- [nan 'not_detected' 'detected']
Parainfluenza 4----------------------------------- [nan 'not_detected' 'detected']
Coronavirus229E----------------------------------- [nan 'not_detected' 'detected']
CoronavirusOC43----------------------------------- [nan 'not_detected' 'detected']
Inf A H1N1 2009----------------------------------- [nan 'not_detected' 'detected']
Bordetella pertussis------------------------------ [nan 'not_detected' 'detected']
Metapneumovirus----------------------------------- [nan 'not_detected' 'detected']
Parainfluenza 2----------------------------------- [nan 'not_detected']
Influenza B, rapid test--------------------------- [nan 'negative' 'positive']
Influenza A, rapid test--------------------------- [nan 'negative' 'positive']
```

Figure 02 : Identification des variables catégorielles

Dans cette étape, nous avons identifié les variables catégorielles (figure02), qui nous permet de sélectionner les colonnes contenant des données de type "Object". Ensuite, nous avons appliqué un encodage pour convertir ces variables catégorielles en valeurs numériques.

En transformant les variables catégorielles en valeurs numériques, nous nous assurons que toutes les variables de notre ensemble de données sont cohérentes et compatibles avec les algorithmes de classification que nous utilisons. Cela nous permet d'exploiter pleinement les informations contenues dans les variables catégorielles lors de la construction de nos modèles de prédiction.

L'étape d'encodage des variables catégorielles revêt une grande importance dans notre processus d'analyse des données, car elle nous permet de convertir des informations qualitatives en valeurs numériques, facilitant ainsi l'application des algorithmes de classification et l'obtention de résultats fiables et interprétables.

3.3. Créer des nouvelles variables à partir des variables existantes

L'étape de "feature engineering" joue un rôle essentiel dans notre analyse en fournissant de nouvelles informations pertinentes à partir des données existantes. Dans notre recherche, nous avons utilisé cette technique pour créer une nouvelle variable appelée "est malade".

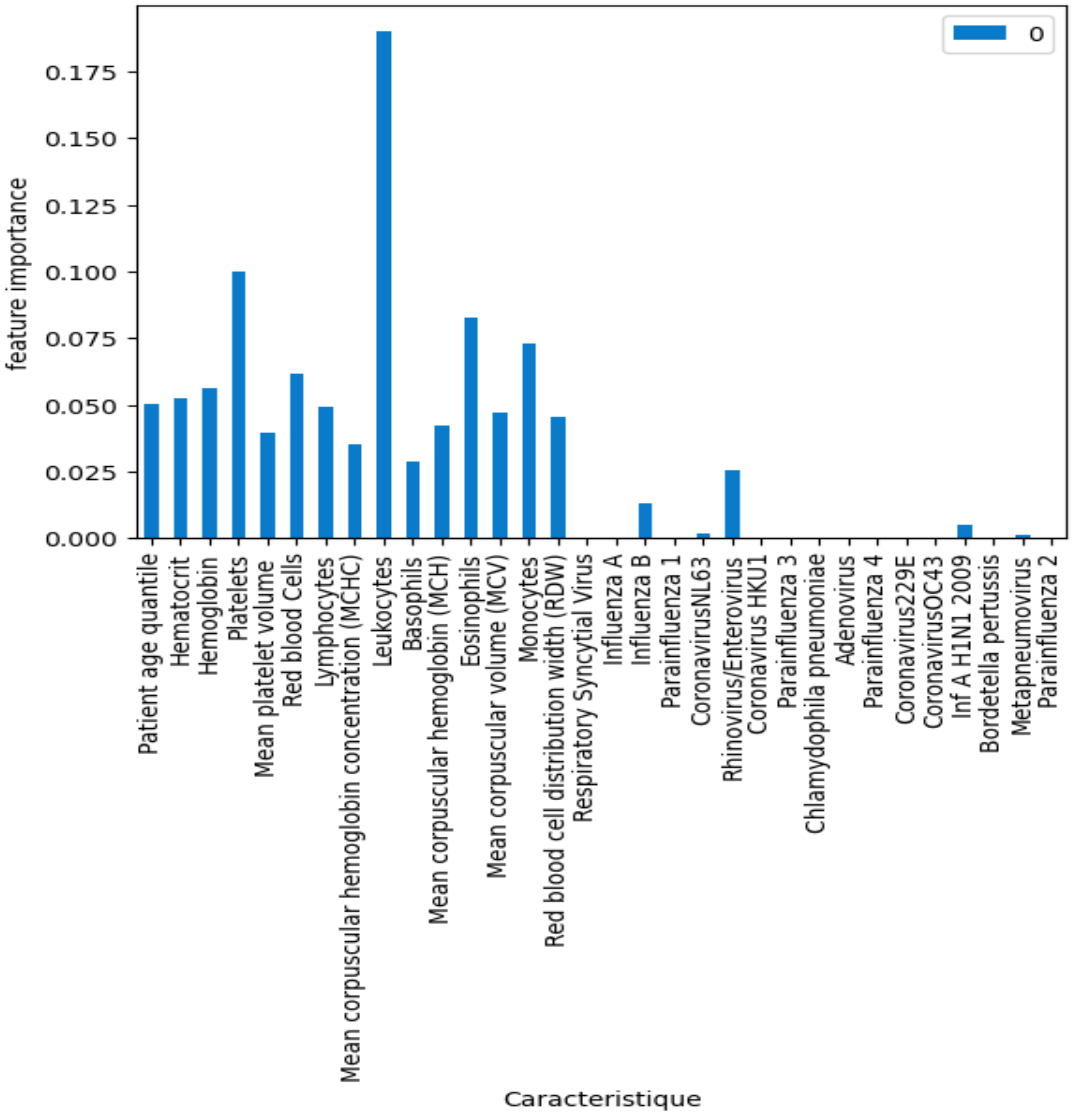

Figure 03 : Les variables importantes

En observant le graphique qui montre les variables les plus importantes (figure 03 ci-dessus), on remarque que les leucocytes exposent une corrélation significative, suivis par les plaquettes. Ensuite, on trouve les globules rouges et les autres variables sont intermédiaires. Cependant, il est important de noter que la majorité des variables sans importance sont les variables virales. Par conséquent, toutes ces variables sont combinées avec une seule variable booléenne appelée "est malade". Ensuite, nous éliminons toutes les variables de type virale.

Les avantages de cette étape sont les suivants : lorsque nous revisitons le graphique des valeurs manquantes que nous avions tracé précédemment, nous constatons que les valeurs manquantes ne sont pas bien alignées entre les données de type sanguines et les données

de type viral. Ainsi, si nous utilisons la fonction (dropna) qui nous permet de supprime les valeurs manquantes sur l'ensemble du jeu de données, nous perdrons un grand nombre de valeurs. En revanche, si nous utilisons (dropna) uniquement sur les variables de type sanguin, nous conserverons plus de données. Et comme plus de données signifie un potentiel d'augmentation du score de validation, cette approche est préférable.

L'importance de cette étape réside dans la capacité à extraire des informations pertinentes à partir des données existantes et à les représenter de manière plus concise et informative. Cela peut améliorer la performance des modèles en réduisant le bruit et en mettant l'accent sur les caractéristiques les plus discriminantes pour la prédiction de l'infection au COVID-19.

3.4. Suppression des valeurs manquantes

L'étape d'imputation des données est essentielle dans notre processus de prétraitement pour garantir la qualité et l'intégrité de notre jeu de données. Dans notre approche, nous avons choisi d'utiliser une méthode de suppression des lignes contenant des valeurs manquantes.

Chapitre 03 : Méthodologie

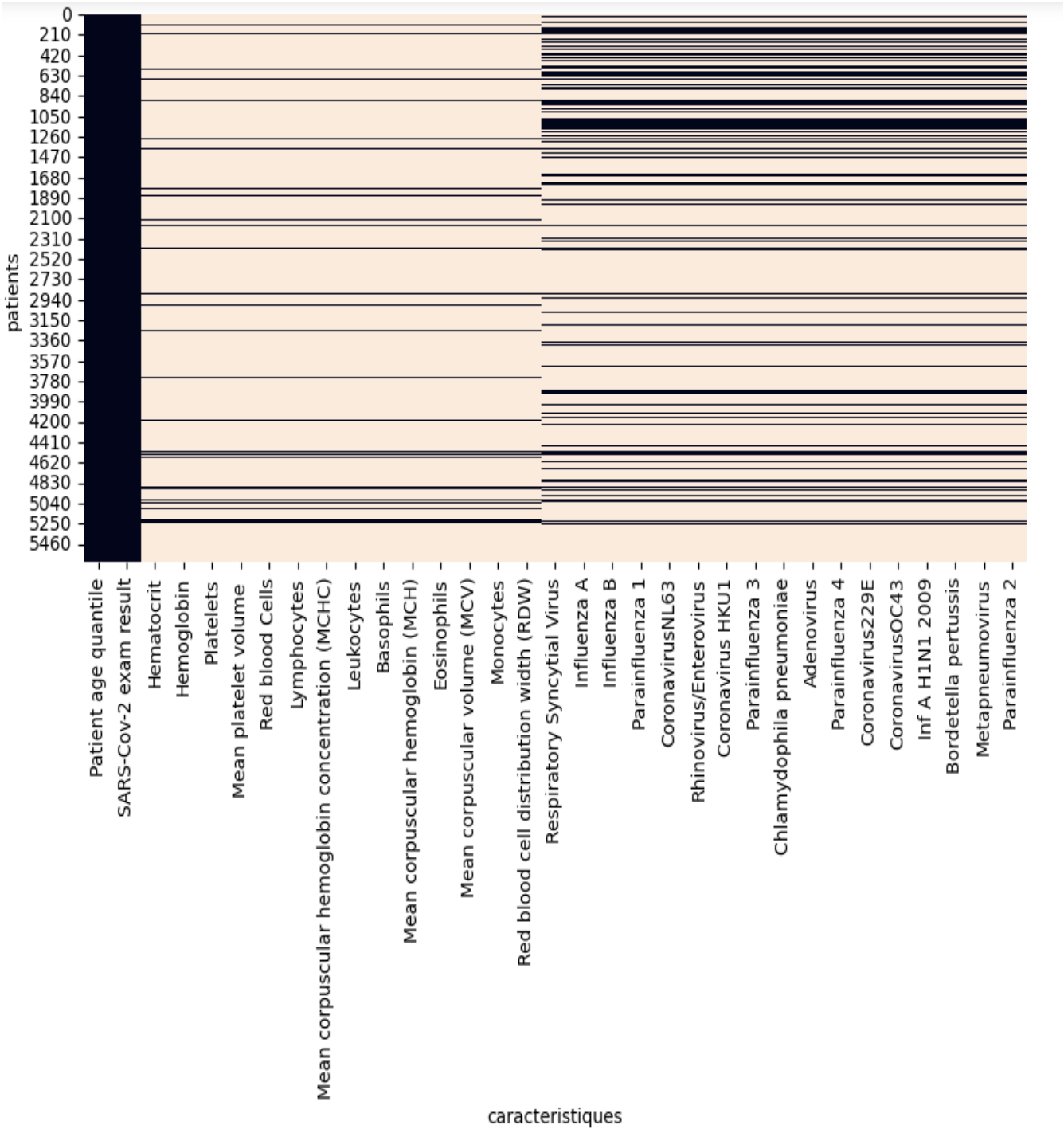

Figure 04 : Heatmap avant la suppression des valeurs manquantes

Chapitre 03 : Méthodologie

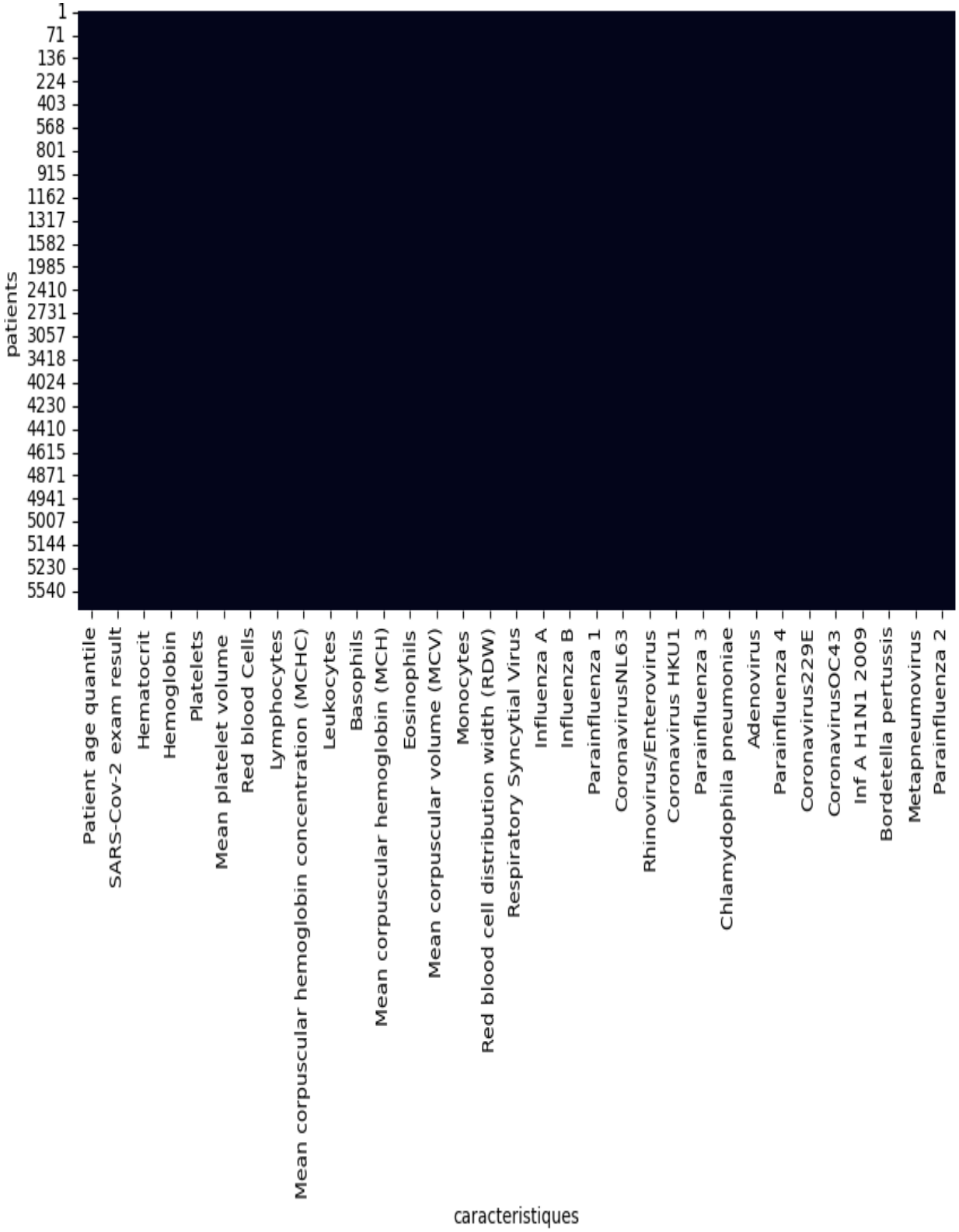

Figure 05 : Heatmap après la suppression des valeurs manquantes

La (figure04) représente un heatmap qui met en évidence les valeurs manquantes dans le data set avant la suppression des valeurs manquantes. Chaque cellule du heatmap correspond à une colonne du data set, et la présence d'une barre colorée (généralement en blanc ou en jaune) indique la présence d'une valeur manquante dans cette colonne. Plus la couleur est intense, plus le nombre de valeurs manquantes est élevé.

La (figure 05) représente un heatmap similaire, mais cette fois-ci après la suppression des valeurs manquantes. Le heatmap montre donc le data set nettoyé, où toutes les valeurs manquantes ont été éliminées. Idéalement, toutes les cellules de couleur noire, indiquant l'absence de valeurs manquantes.

En comparant les deux figures, on peut observer visuellement le changement dans la répartition des valeurs manquantes avant et après la suppression de celles-ci. Cela permet d'évaluer l'impact de la suppression des valeurs manquantes sur la qualité et la complétude des données dans le data set.

3.5. Le transformateur PolynomialFeatures

L'utilisation de caractéristiques polynomiales de degré 2 est motivée par la nature non linéaire des relations entre les variables cliniques et la probabilité d'infection au COVID-19. Les relations linéaires traditionnelles ne sont souvent pas suffisantes pour représenter ces interactions complexes. En créant des caractéristiques polynomiales de degré 2, le modèle est en mesure d'explorer des relations quadratiques entre les caractéristiques, ce qui permet de modéliser des effets non linéaires et des interactions de second ordre [25].

Cette approche est essentielle car certaines caractéristiques cliniques peuvent avoir des effets quadratiques sur la probabilité d'infection. Tel que, l'âge combiné à d'autres variables cliniques peut influencer de manière non linéaire la probabilité d'infection. En incluant ces interactions quadratiques, nous sommes en mesure de distinguer des relations plus complexes et réalistes entre les caractéristiques cliniques et l'infection au COVID-19.

3.6. Le transformateur SelectKBest

En utilisant le test F de l'analyse de variance (f_classif), le transformateur SelectKBest attribue un score à chaque caractéristique, reflétant son degré d'association avec la variable cible. L'objectif est de choisir les caractéristiques les plus importantes parmi toutes celles disponibles.

Dans notre étude, on a utilisé le paramètre k=10 pour sélectionner les 10 meilleures caractéristiques les plus pertinentes. Cette sélection se base sur la corrélation avec la variable cible, ce qui permet de réduire la dimensionnalité du problème en ne conservant que les caractéristiques les plus informatives.

La sélection des caractéristiques les plus pertinentes avec le transformateur SelectKBest est importante car elle permet de se concentrer sur les aspects les plus influents de notre data set. Cela peut contribuer à améliorer les performances des modèles de prédiction de l'infection au COVID-19 en utilisant des caractéristiques significatives et en réduisant le bruit potentiel causé par des caractéristiques moins importantes [27].

3.7. Normalisation des variables

La normalisation des variables est une étape importante dans le traitement des données pour la prédiction de l'infection au COVID-19 dans notre data set. Pour cette tâche, nous allons utiliser le transformateur RobustScaler.

Le RobustScaler est une méthode de normalisation robuste qui est utilisée pour mettre à l'échelle les variables en présence de valeurs aberrantes ou d'écarts importants entre les données. Contrairement à d'autres méthodes de normalisation telles que le MinMaxScaler, le RobustScaler utilise la médiane et l'écart interquartile plutôt que la moyenne et l'écart-type pour normaliser les données [22] .

La médiane et l'écart interquartile sont des mesures résistantes aux valeurs aberrantes, ce qui signifie qu'ils ne sont pas influencés de manière significative par les valeurs extrêmes ou les valeurs manquantes. Par conséquent, le RobustScaler est plus adapté à des données présentant des valeurs aberrantes ou une distribution non normale [22] .

En appliquant le RobustScaler à notre data set, nous allons pouvoir ramener toutes les variables à une échelle commune. Cela est important car les algorithmes de classification tels que le SVC et le KNeighborsClassifier que nous avons utilisés sont sensibles à l'échelle des variables. La normalisation permet de garantir que toutes les variables ont une influence équilibrée dans la prédiction de l'infection au COVID-19.

Chapitre 03 : Méthodologie

4. Choix des algorithmes de classification

Pour prédire l'infection au COVID-19 à partir de données cliniques, il est essentiel de sélectionner des algorithmes de classification appropriés. Nous avons choisi les algorithmes suivants en raison de leur popularité, de leur performance dans des tâches de classification similaires et de leur capacité à gérer différents types de données :

- **AdaBoostClassifier** : L'algorithme AdaBoost est un algorithme de boosting qui combine plusieurs modèles d'apprentissage faibles pour former un modèle fort. Il est particulièrement efficace pour résoudre les problèmes de classification binaires et peut être utilisé avec diverses fonctions de base (classificateurs faibles). [14].
- **RandomForestClassifier :** RandomForestClassifier est un algorithme de type ensemble qui construit plusieurs arbres de décision aléatoires et combine leurs prédictions pour obtenir une prédiction finale. Il est adapté aux ensembles de données de grande dimension et est capable de gérer des relations non linéaires entre les caractéristiques et la variable cible. Nous avons choisi RandomForestClassifier car il peut fournir une bonne performance prédictive tout en évitant les problèmes de sur ajustement [15].
- **Support Vector Classifier (SVC)** : SVC est un algorithme de classification basé sur les machines à vecteurs de support. Il est particulièrement efficace pour résoudre des problèmes de classification binaires, même lorsque les données sont linéairement non séparables dans l'espace d'origine. En utilisant des fonctions de noyau, SVC peut également gérer des relations non linéaires entre les caractéristiques et la variable cible. [9].
- **KNeighborsClassifier :** KNeighborsClassifier est un algorithme basé sur les k plus proches voisins. Il attribue une étiquette de classe à un nouvel exemple en fonction de la majorité des étiquettes de ses k voisins les plus proches dans l'espace des caractéristiques. Cet algorithme est adapté aux ensembles de données avec des frontières de décision complexes et peut gérer des relations non linéaires. Nous avons inclus KNeighborsClassifier dans notre étude en raison de sa simplicité et de sa capacité à capturer des relations locales entre les caractéristiques et la variable cible [10].

En sélectionnant ces algorithmes de classification, nous espérons couvrir une variété d'approches et de capacités pour prédire l'infection au COVID-19 à partir des données cliniques.

Cela nous permettra de comparer leurs performances et de déterminer quel algorithme est le plus approprié pour notre tâche de prédiction.

5. Mise en œuvre des algorithmes de classification sur les données prétraitées

Dans le cadre de notre étude, nous allons mettre en œuvre les algorithmes de classification sélectionnés (AdaBoostClassifier, RandomForestClassifier, SVC et KNeighborsClassifier) sur les données prétraitées. Voici comment nous allons réaliser cette étape :

Après le prétraitement des données, nous allons mettre en œuvre les algorithmes de classification sur les données préparées. Pour chaque algorithme, nous allons utiliser une implémentation disponible dans une bibliothèque de machine Learning, telle que Scikit-learn.

Pour chaque algorithme, Nous allons diviser les données prétraitées en ensembles d'entraînement et de test, en utilisant une proportion de 80% pour les données d'entraînement et 20% pour les données de test.

Ensuite, nous allons entraîner chaque algorithme de classification sur l'ensemble d'entraînement et évaluer ses performances sur l'ensemble de test. Nous allons enregistrer les métriques de performance pertinentes, telles que la matrice de confusion, la précision, le rappel, le score F1 et la courbe d’apprentissage pour chaque algorithme.

En mettant en œuvre les algorithmes de classification sur les données prétraitées, nous allons réaliser une comparaison des performances des différents algorithmes de classification (AdaBoostClassifier, RandomForestClassifier, SVC, KNeighborsClassifier) pour la prédiction de l'infection au COVID-19 à partir de données cliniques. Nous allons également effectuer une analyse des facteurs contribuant à la prédiction de l'infection et identifier l'algorithme de classification le plus performant. Ensuite, nous allons optimiser l'algorithme sélectionné en explorant différentes techniques d'optimisation telles que l'ajustement des hyperparamètres la sélection des caractéristiques et l'optimisation des seuils de décision. Nous allons évaluer les performances après l'optimisation et discuter des améliorations obtenues.

Chapitre 03 : Méthodologie

6. Conclusion

Dans ce chapitre, nous avons consacré une attention particulière à décrire en détail le processus que nous avons suivi pour mener notre étude sur le coronavirus, en mettant l'accent sur les étapes de prétraitement des données que nous allons entreprendre. Ces étapes jouent un rôle important dans la garantie de la qualité et de la fiabilité de notre analyse, après avoir effectué les étapes de prétraitement des données, nous avons soigneusement choisi les algorithmes de classification les plus appropriés. Ce choix méthodique nous permettra de maximiser la qualité, la fiabilité et la pertinence de notre analyse et nous permettra d'obtenir des résultats pertinents et significatifs par la suite.

Chapitre 4

Réalisation et Résultats

1. Introduction

L'objectif de ce rapport est d'analyser les performances des différents algorithmes de classification et d'explorer les techniques d'optimisation afin d'identifier l'algorithme le plus performant pour une tâche donnée. Dans le cadre de cette étude, nous avons utilisé divers outils de développement, tels que Anaconda et Jupiter, ainsi que différentes bibliothèques couramment utilisées en apprentissage automatique pour la réalisation de notre travail.

2. Outils de développement

2.1. Anaconda

Anaconda est une plateforme complète et puissante dédiée à la science des données et à l'apprentissage automatique. Elle a été créée en 2012 pour faciliter l'utilisation de Python dans l'analyse des données commerciales en pleine évolution. Depuis lors, Anaconda est devenue un outil essentiel pour les étudiants, les professionnels et les entreprises du monde entier. Elle offre une vaste collection de bibliothèques, d'outils et d'environnements de développement prêts à l'emploi, permettant aux utilisateurs de travailler de manière efficace sur des projets de données.

Avec Anaconda, les utilisateurs ont accès à une installation simplifiée de Python, ainsi qu'à une variété d'outils supplémentaires tels que Jupiter Notebook pour l'analyse interactive, NumPy et pandas pour la manipulation des données, et Scikit-learn pour l'apprentissage automatique. La plateforme est conçue pour faciliter le développement, la collaboration et le déploiement de projets en fournissant un environnement cohérent et fiable.

En tant que solution open-source, Anaconda est soutenue par une communauté active et bénéficie de mises à jour régulières, assurant la disponibilité des dernières fonctionnalités et améliorations. Que ce soit pour l'apprentissage, la recherche ou les applications professionnelles, Anaconda offre un ensemble d'outils et de ressources essentiels pour exploiter le potentiel de la science des données et de l'apprentissage automatique. [36].

2.2. JUPITER

Jupiter est une plateforme polyvalente qui permet aux utilisateurs de travailler de manière interactive avec du code, des visualisations et des explications. Il favorise la collaboration et facilite la science des données et le calcul scientifique dans divers langages de programmation. [37].

2.3. Bibliothèques utilisées

Les bibliothèques sont des ensembles de modules ou de packages qui contiennent des fonctionnalités et des outils préconstruits pour faciliter le développement de logiciels dans des domaines spécifiques. Dans le contexte de la programmation en Python, il existe de nombreuses bibliothèques populaires qui offrent une large gamme de fonctionnalités pour diverses tâches.

- **NumPy :** NumPy est une bibliothèque fondamentale pour le calcul scientifique en Python. Elle fournit des structures de données performantes pour les tableaux multidimensionnels, ainsi que des fonctions mathématiques pour manipuler et analyser ces tableaux. NumPy est largement utilisé pour la manipulation de données et le calcul numérique dans l'apprentissage automatique [38].
- **Pandas :** Pandas est une bibliothèque qui offre des structures de données et des outils d'analyse de données faciles à utiliser. Elle permet de manipuler et de traiter efficacement de grandes quantités de données, en offrant des fonctionnalités telles que la fusion de données, le filtrage, le tri, l'agrégation et la gestion des valeurs manquantes. Pandas est largement utilisé pour l'exploration de données et la préparation des données dans l'apprentissage automatique [39].
- **Matplotlib :** Matplotlib est une bibliothèque de traçage de données qui permet de créer des graphiques et des visualisations de haute qualité. Elle offre une grande flexibilité pour créer des graphiques en 2D et en 3D, des diagrammes, des histogrammes, des graphiques à barres, etc. Matplotlib est couramment utilisé pour visualiser les données, les résultats de modèles et les analyses statistiques [40].
- **Scikit-learn :** Scikit-learn est une bibliothèque d'apprentissage automatique conviviale et complète en Python. Elle propose une large gamme d'algorithmes d'apprentissage automatique supervisé et non supervisé, ainsi que des outils pour la préparation des données, la validation croisée, la sélection de modèles, l'évaluation des performances et

plus encore. Scikit-learn est une bibliothèque incontournable pour la plupart des tâches d'apprentissage automatique [41].

- **Seaborn :** Seaborn est une bibliothèque Python pour la visualisation de données statistiques. Elle offre des graphiques esthétiquement améliorés, des fonctions simplifiées pour les graphiques statistiques, et facilite la visualisation des données catégorielles. Seaborn s'intègre à Pandas pour visualiser des données tabulaires, ce qui en fait un outil populaire dans l'analyse de données et la science des données [42].

2.4. Caractéristiques de la machine utilisée

Les caractéristiques de la machine sur laquelle nous avons mis en œuvre notre application
Et évalué nos jeux de données sont les suivantes :

- **Machine 01 :**
  – Type de processeur Intel (R) Core (TM) i3-6006U CPU 2.00GHz, 1.99 MHz,
  – Fabricant Intel
  – Version IntelrCore™ i3-6006U CPU 2.00GH, 8.00 Go de RAM
  – Voltage 0.75 V
  – Capacité de disque dur 237 Go
- **Machine 02 :**
  – Type de processeur Intel (R) Core (TM) i3-1101U CPU 2.00GHz, 1.99 MHz,
  – Fabricant Intel
  – Version IntelrCore™ i3-1101U CPU 2.00GH, 8.00 Go de RAM
  – Voltage 0.75 V
  – Capacité de disque dur 237 Go

Chapitre 04 : Réalisation et Résultats

3. Analyse des performances des différents algorithmes de classification

3.1. Analyse des matrices de confusion

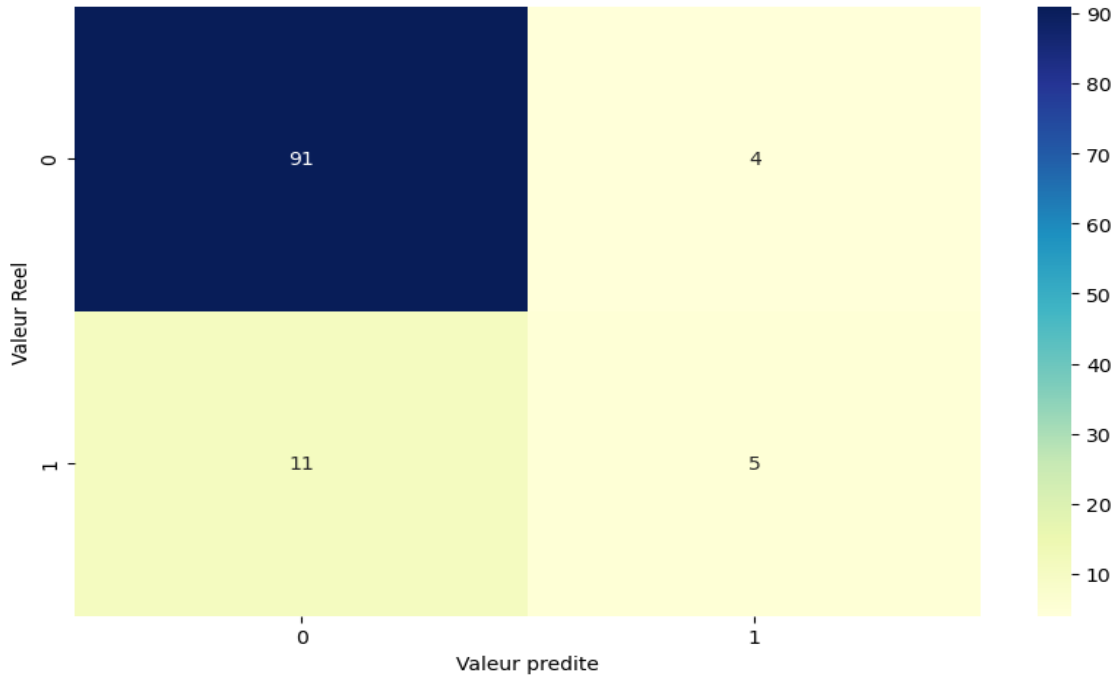

Figure 01 : Matrice de Confusion (RandomForest)

Cette matrice de confusion montre les résultats de la classification effectuée par le modèle RandomForest (Figure 01). Le modèle a correctement prédit 91 cas négatifs (vrais négatifs) et 5 cas positifs (vrais positifs). Cependant, il a également fait 4 erreurs en prédisant des cas négatifs comme positifs (faux négatifs) et 11 erreurs en prédisant des cas positifs comme négatifs (faux positifs).

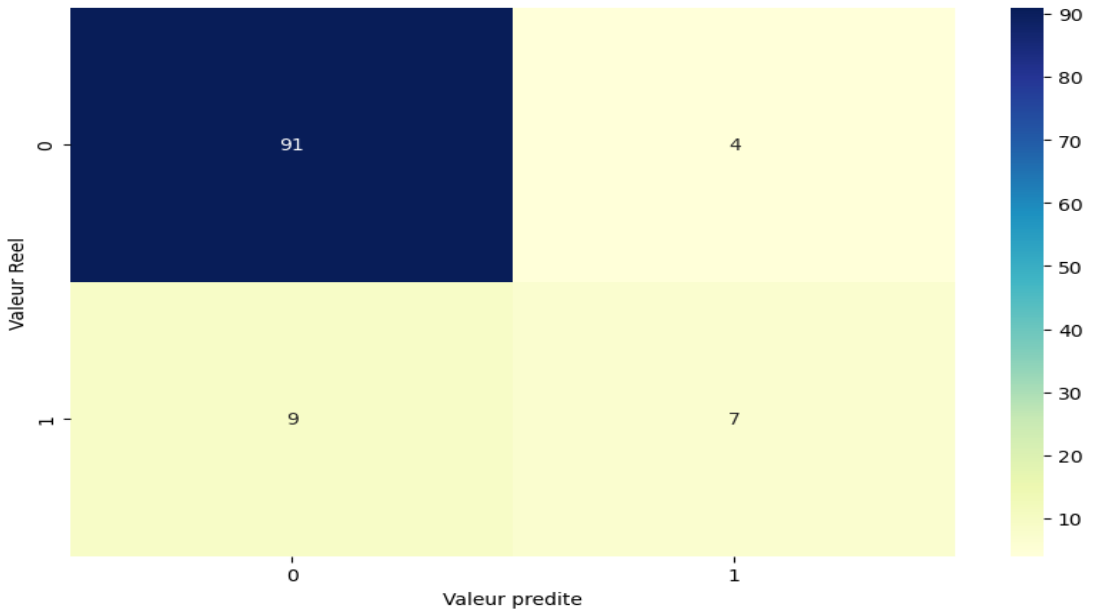

Figure 02 : Matrice de Confusion (AdaBoost)

Dans cette matrice de confusion (Figure 02), le modèle AdaBoost a correctement prédit 91 cas négatifs et 7 cas positifs. Il a commis 4 erreurs en prédisant des cas négatifs comme positifs et 9 erreurs en prédisant des cas positifs comme négatifs.

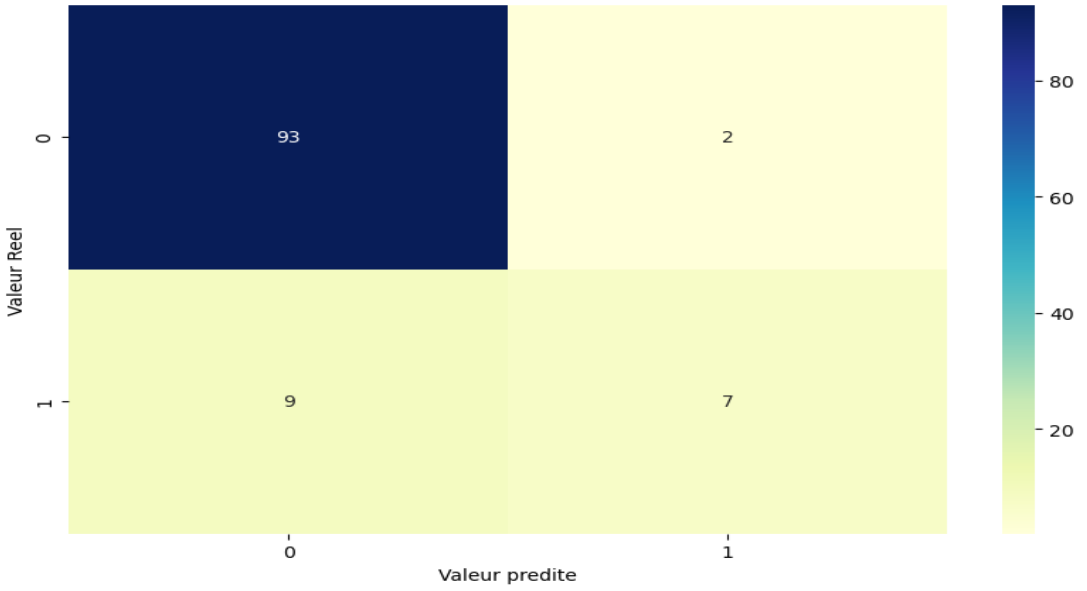

Figure 03 : Matrice de Confusion (SVM)

Chapitre 04 : Réalisation et Résultats

La matrice de confusion pour le modèle SVM (Figure 03), montre qu'il a correctement prédit 93 cas négatifs et 7 cas positifs. Il a fait 2 erreurs en prédisant des cas négatifs comme positifs et 9 erreurs en prédisant des cas positifs comme négatifs.

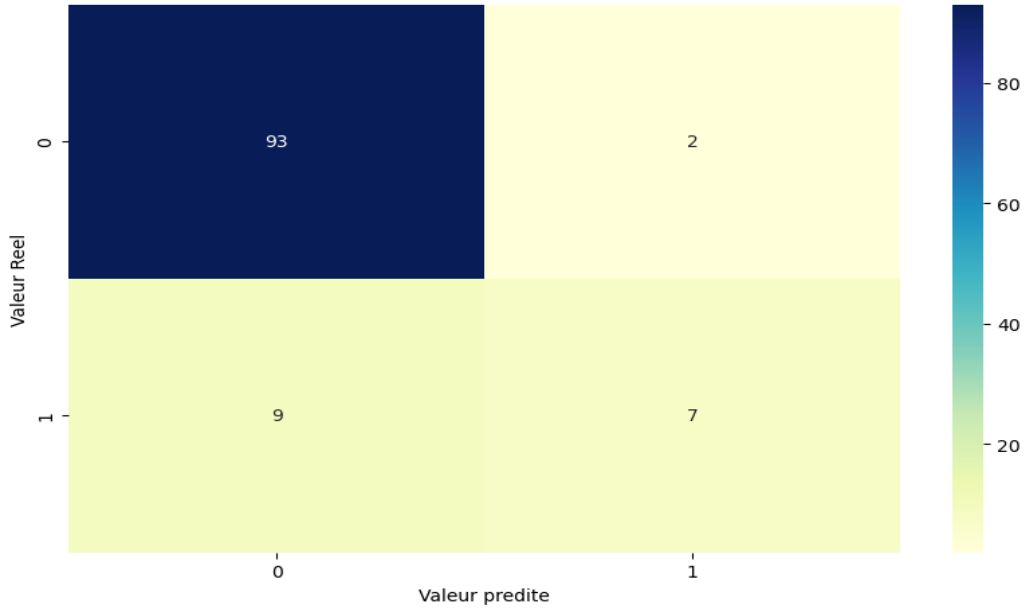

Figure 04 : Matrice de Confusion (KNN)

Dans cette matrice de confusion (Figure 04), le modèle KNN a correctement prédit 93 cas négatifs et 7 cas positifs. Il a également fait 2 erreurs en prédisant des cas négatifs comme positifs et 9 erreurs en prédisant des cas positifs comme négatifs.

En comparant ces matrices de confusion, nous pouvons noter les performances suivantes :

- Random Forest a un nombre relativement élevé de faux négatifs (FN) par rapport aux autres modèles.
- AdaBoost a un nombre équilibré de vrais positifs (TP) et de faux positifs (FP).
- SVM et KNN présentent des performances similaires avec un nombre équilibré de vrais positifs (TP), de vrais négatifs (TN) et de faux négatifs (FN), ainsi qu'un faible nombre de faux positifs (FP).

Il est important de noter que la comparaison des performances ne peut être basée uniquement sur les matrices de confusion. D'autres mesures d'évaluation telles que la précision, le rappel, la F1-score et les courbes d'apprentissage peuvent fournir une analyse plus complète et précise des performances des modèles.

3.2. Analyse rapports de classification

Tableau 01 : Rapports de Classification

	Précision		Recall		F1-score		Support	
	Classe 0	Classe 1	Classe 0	Classe 1	Classe 0	Classe 1	Classe 0	Classe1
Random Forest	0.89	0.56	0.96	0.31	0.92	0 .40	95	16
AdaBoost	0.91	0.64	0.96	0.44	0.93	0.52		
SVM	0.91	0.78	0.98	0.44	0.94	0.56		
KNN	0.91	0.78	0.98	0.44	0.94	0.56		

En analysant les résultats obtenus, nous pouvons observer les performances de chaque algorithme :

Nous pouvons constater que RandomForest a la précision la plus faible pour la classe positive (0.56), tandis que SVM et KNN ont la plus haute précision pour cette classe (0.78). En termes de rappel et de score F1 pour la classe positive, SVM et KNN ont également de meilleurs résultats.

Cependant, pour la classe négative (non-infection), tous les algorithmes obtiennent de très bonnes performances avec des valeurs élevées de précision, de rappel et de score F1.

En comparant les performances de ces différents algorithmes de classification, nous pouvons conclure que SVM et KNN ont obtenu les meilleurs résultats en termes de précision, de rappel et de score F1 pour la prédiction de l'infection au COVID-19. Ces deux algorithmes ont démontré une capacité supérieure à identifier avec précision les individus infectés par le virus.

3.3. Analyse des courbes d'apprentissage

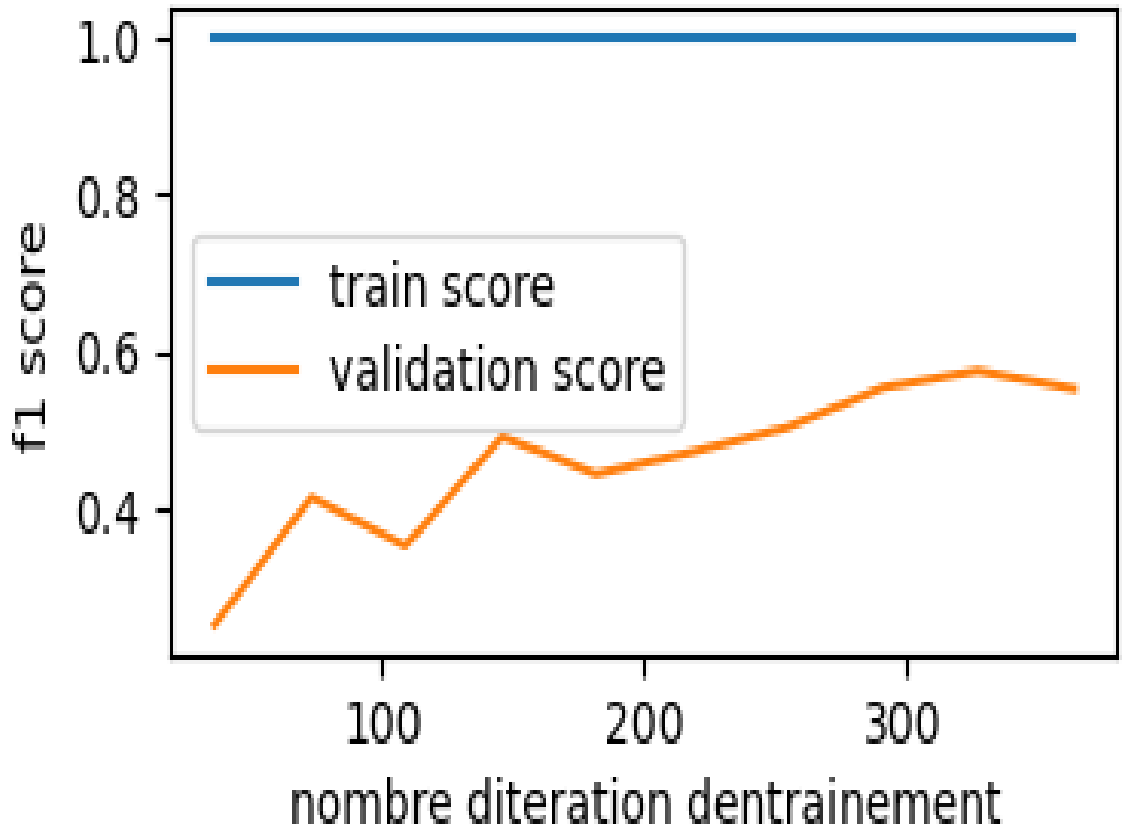

Figure 05 : Courbe d'apprentissage du modèle RandomForestClassifier

Notre modèle (figure 05) présente un cas de surajustement (overfitting) car il obtient un score de 100% sur les données de l'ensemble d'entraînement. Cela indique que le modèle a appris de manière parfaite à partir de ces données spécifiques. Cependant, lorsqu'il est confronté à de nouvelles données du jeu de validation, c'est-à-dire des données qu'il n'a pas rencontrées lors de son entraînement, sa performance en termes de score f1 est considérablement réduite. Plus précisément, nous obtenons un score de 58%, ce qui est encore acceptable mais souligne clairement l'écart important entre les performances sur l'ensemble d'entraînement et l'ensemble de validation, indiquant ainsi que notre modèle est en situation de surajustement.

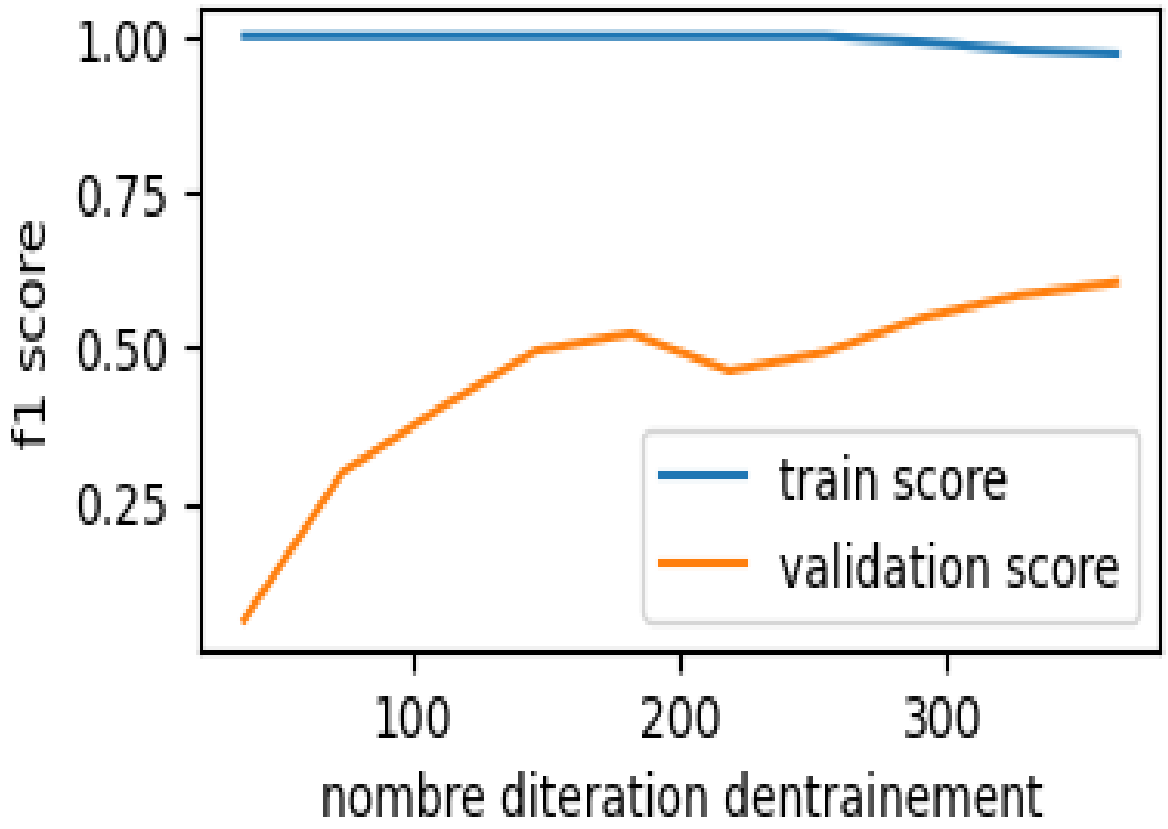

Figure 06 : Courbe d'apprentissage du modèle AdaBoostClassifier

Notre modèle (figure 06) obtient un score presque parfait de près de 100% sur les données de l'ensemble d'entraînement, ce qui indique qu'il a appris de manière parfaite à partir de ces données spécifiques. Cependant, lorsqu'il est confronté à de nouvelles données du jeu de validation, c'est-à-dire des données qu'il n'a jamais vues pendant son entraînement, sa performance en termes de score f1 est considérablement réduite. Plus précisément, nous obtenons un score de 62%, ce qui est déjà assez satisfaisant. Cependant, malgré cette amélioration, l'écart important entre les performances sur l'ensemble d'entraînement et l'ensemble de validation nous indique que notre modèle souffre de surajustement (overfitting). Il est intéressant de noter que le score de validation pour AdaBoostClassifier est plus élevé que celui de RandomForestClassifier.

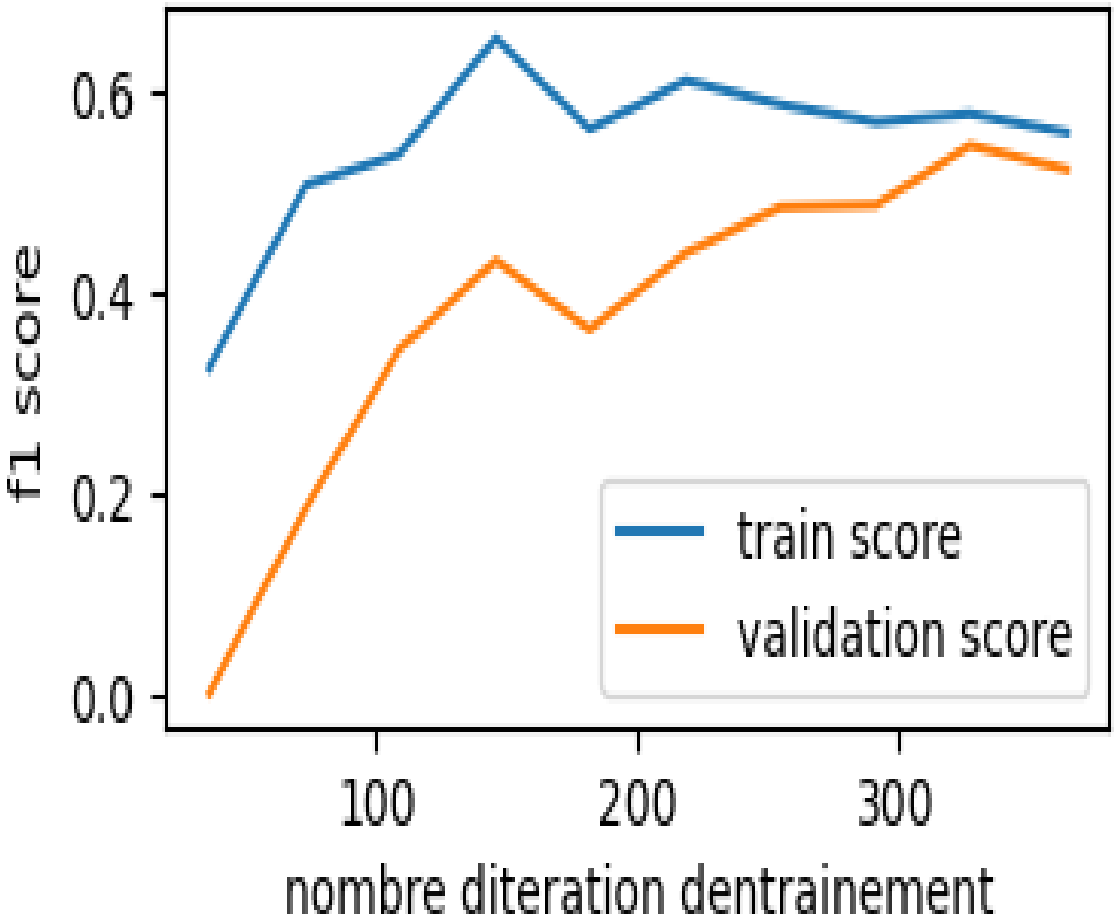

Figure 07 : Courbe d'apprentissage du modèle KNN

Le KNN semble obtenir de bons scores, malgré une légère diminution du score d'entraînement (figure 07). Cependant, ce qui est important, c'est d'avoir un écart réduit entre l'ensemble d'entraînement et l'ensemble de validation, car cela indique que le modèle a bien appris et est capable de généraliser, évitant ainsi le surajustement (overfiting). Le KNN pourrait donc être un choix intéressant. Cependant, nous préfèrons le mettre de côté car il ne semble pas être le choix le plus adapté. Le KNN est un modèle basé sur les instances, ce qui signifie qu'il se concentre uniquement sur les données fournies. Par conséquent, nous préférons nous concentrer sur le modèle SVM qui semble prometteur.

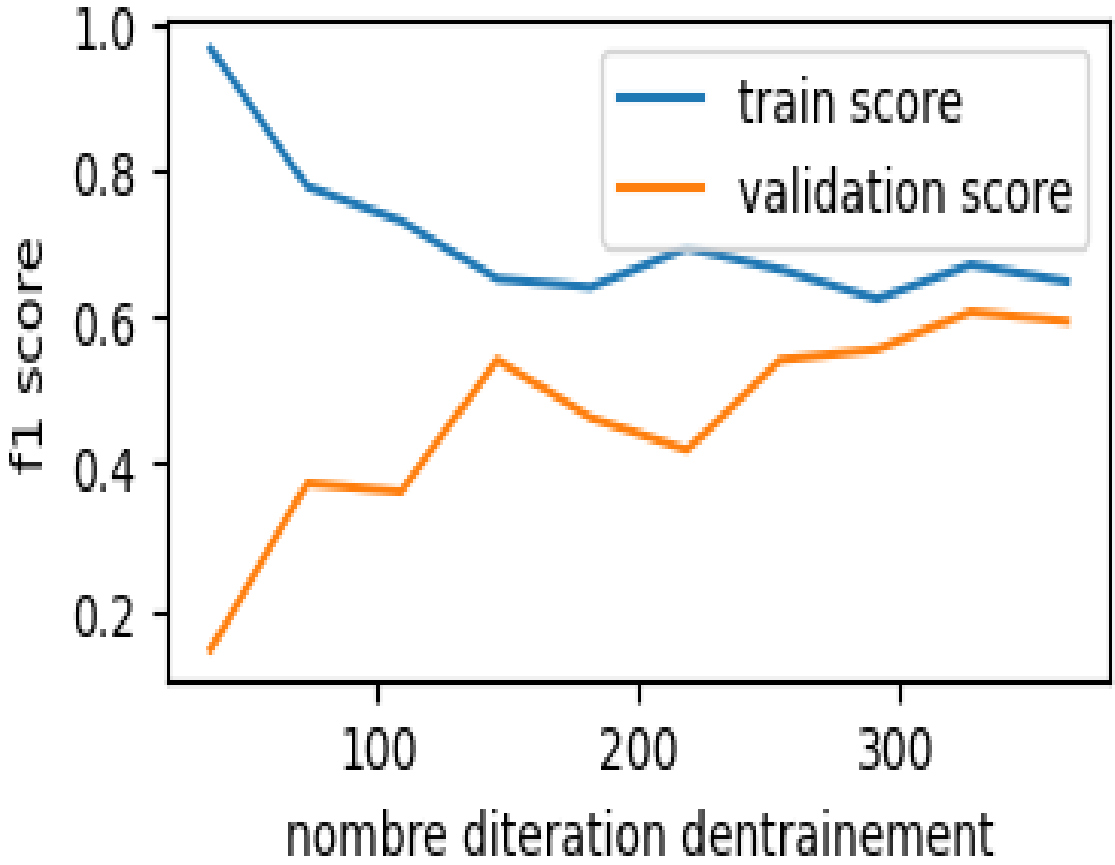

Figure 08 : Courbe d'apprentissage du modèle SVM

Le SVM est le plus intéressant car il nous indique qu'il ne souffre pas de surajustement(figure08). Son score sur l'ensemble d'entraînement a diminué, mais il se rapproche considérablement de l'ensemble de validation. Dans cette situation, nous ne pouvons pas parler de surajustement. Le SVM devrait être le bon choix parmi les modèles que nous avons examinés précédemment.

4. Identification de l'algorithme de classification le plus performant

Apres avoir analysé les résultats des différents algorithmes de classification, l'algorithme SVM semble être le plus performant. Il a démontré une capacité supérieure à identifier avec précision les individus infectés par le virus, et il a également montré une meilleure capacité à généraliser et à éviter le surajustement par rapport aux autres modèles.

Chapitre 04 : Réalisation et Résultats

5. Exploration des techniques d'optimisation

L’optimisation des hyper paramètres a été réalisée pour améliorer les performances de l'algorithme de classification SVM (Support Vector Machine) dans la prédiction de l'infection au COVID-19 à partir des données cliniques fournies.

Les hyper paramètres sont des paramètres définis avant l'apprentissage du modèle et qui influencent ses performances. L'ajustement des hyper paramètres consiste à rechercher les meilleures combinaisons de valeurs pour ces paramètres afin d'optimiser les performances du modèle.

Les hyper paramètres suivants ont été sélectionnés pour l'optimisation :

- Svc__gamma : Ce paramètre contrôle l'influence des exemples d'entraînement sur la frontière de décision. Différentes valeurs ont été testées, notamment 1e-3, 1e-4 et 0.0005.
- Svc__C : Il s'agit du paramètre de régularisation qui contrôle la marge d'erreur tolérée par le modèle. Différentes valeurs ont été testées, telles que 1, 10, 100, 1000 et 3000.
- Pipeline__polynomialfeatures__degree : Ce paramètre est spécifique à la transformation des caractéristiques en utilisant PolynomialFeatures dans le pipeline. Différents degrés ont été testés, notamment 2 et 3, pour capturer les relations polynomiales entre les caractéristiques.
- Pipeline__selectkbest__k : Ce paramètre est utilisé pour sélectionner les meilleures caractéristiques à inclure dans le modèle. Différentes valeurs ont été testées, notamment de 45 à 60.

La recherche des meilleures combinaisons d'hyper paramètres a été effectuée à l'aide de la méthode RandomizedSearchCV. Cette méthode effectue une recherche aléatoire des combinaisons d'hyper paramètres dans l'espace défini, tout en utilisant une validation croisée pour évaluer les performances des modèles. Un total de 40 itérations a été réalisées pour notre recherche.

Après l'optimisation, les meilleurs paramètres ont été obtenus en utilisant grid. Best_params_, et ces paramètres ont été utilisés pour prédire les étiquettes sur l'ensemble de test.

Chapitre 04 : Réalisation et Résultats

L'optimisation des hyper paramètres permet d'améliorer les performances du modèle en ajustant les paramètres pour mieux s'adapter aux données fournies. Cela peut conduire à une prédiction plus précise de l'infection au COVID-19 à partir des données cliniques, ce qui est essentiel pour la gestion de la pandémie et la prise de décisions en matière de santé publique.

6. Évaluation des performances après l'optimisation

6.1. Analyse de la matrice de confusion optimisé :

Après l'optimisation du modèle, la matrice de confusion résultante est la suivante :

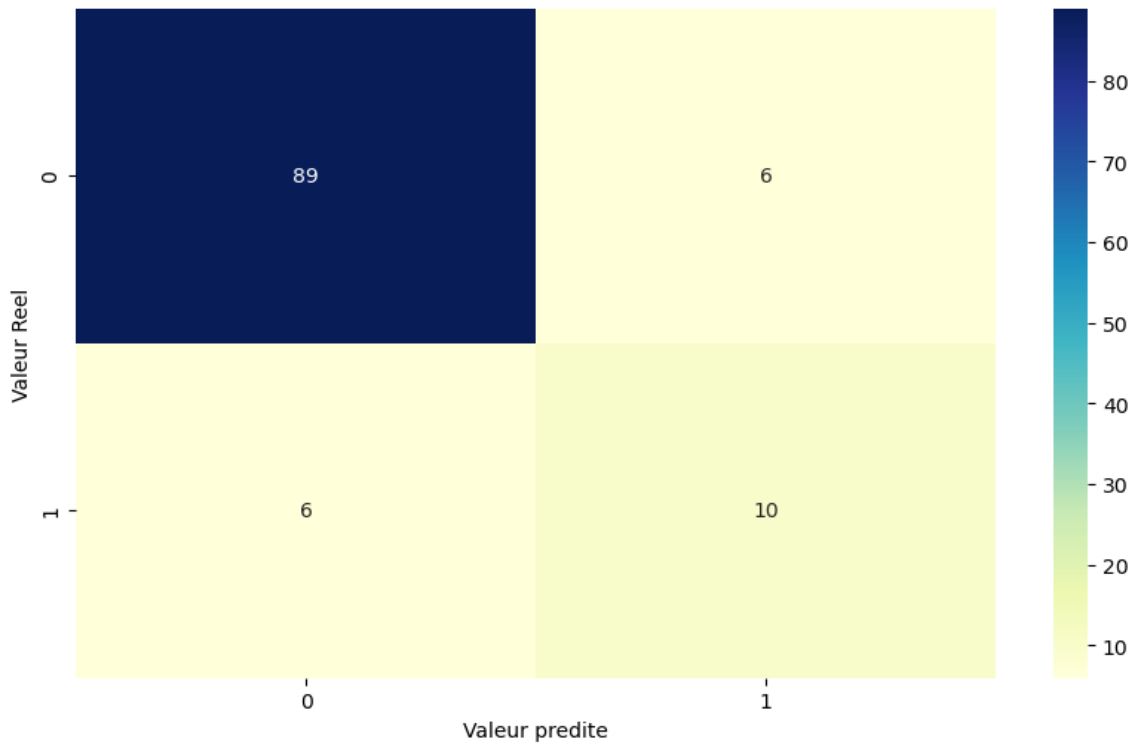

Figure 09 : Matrice de confusion de modèle SVM optimisé

Les éléments en diagonale représentent les prédictions correctes, tandis que les éléments hors diagonale représentent les erreurs de prédiction.

Pour la classe 0 (non infectée) :

- 89 échantillons ont été correctement prédits comme non infectés (vrais négatifs).
- 6 échantillons ont été prédits à tort comme infectés (faux positifs).

Pour la classe 1 (infectée) :

- 6 échantillons ont été prédits à tort comme non infectés (faux négatifs).
- 10 échantillons ont été correctement prédits comme infectés (vrais positifs).

6.2. Analyse du rapport de classification optimisé

Après l'optimisation du modèle SVM, le rapport de classification présente les résultats suivants

	precision	recall	f1-score	support
0	0.94	0.94	0.94	95
1	0.62	0.62	0.62	16
accuracy			0.89	111
macro avg	0.78	0.78	0.78	111
weighted avg	0.89	0.89	0.89	111

Figure 10 : Rapports de Classification Apres L'optimisation

Pour la classe 0 (non infectée) : La précision est de 0,94, ce qui signifie que le modèle prédit correctement 94 % des individus non infectés parmi tous les individus prédits comme non infectés. Le rappel (taux de vrais positifs) est également de 0,94, indiquant que le modèle identifie correctement 94 % des individus non infectés parmi tous les individus réellement non infectés. Le score F1 de 0,94 est une mesure harmonique de la précision et du rappel. Cela montre que le modèle a une performance élevée pour prédire les cas négatifs.

Pour la classe 1 (infectée) : La précision est de 0,62, ce qui indique que le modèle prédit correctement 62 % des individus infectés parmi tous les individus prédits comme infectés. Le rappel (taux de vrais positifs) est également de 0,62, ce qui signifie que le modèle identifie correctement 62 % des individus infectés parmi tous les individus réellement infectés. Le score F1 de 0,62 est une mesure harmonique de la précision et du rappel. Cela montre que le modèle a une performance modérée pour prédire les cas positifs.

L'exactitude globale (accuracy) du modèle est de 0,89, ce qui représente la proportion d'individus correctement prédits parmi tous les individus. Cependant, l'exactitude seule peut

être trompeuse lorsque les classes sont déséquilibrées, car elle ne tient pas compte des faux positifs et des faux négatifs.

6.3. Discussion des améliorations obtenues grâce à l'optimisation

L'optimisation du modèle SVM a permis d'améliorer ses performances par rapport à sa version non optimisée.

Tableau 02 : Matrice de confusion avant et après l'optimisation

	Classe 0 (non infectée)	**Classe 1 (infectée)**
Avant	93 (vrais négatifs)	7 (vrais positifs)
	2 (faux positifs)	9 (faux négatifs)
Après	89 (vrais négatifs)	10 (vrais positifs)
	6 (faux positifs)	6 (faux négatifs)

Avant l'optimisation, le modèle SVM présentait une matrice de confusion avec 93 prédictions correctes pour la classe 0 et 7 prédictions correctes pour la classe 1. Cependant, il y avait également 9 prédictions erronées pour la classe 0 et 2 prédictions erronées pour la classe 1. Le rapport de classification indiquait une précision de 91% pour la classe 0, mais seulement 78% pour la classe 1. Le rappel (Recall) était de 98% pour la classe 0, mais seulement de 44% pour la classe 1.

Après l'optimisation, les meilleurs hyperparamètres sélectionnés étaient : 'svc__gamma' : 0.0001,'svc__C':3000,'pipeline__selectkbest__k':53et'pipeline__polynomialfeatures__degree': 3. Le modèle SVM optimisé a présenté une matrice de confusion avec 89 prédictions correctes pour la classe 0 et 10 prédictions correctes pour la classe 1. Les prédictions erronées ont été réduites à 6 pour la classe 0 et 6 pour la classe 1.

Le rapport de classification du modèle SVM optimisé a montré une amélioration globale des performances. La précision pour la classe 0 était de 94%, soit une augmentation de 3% par rapport à la version non optimisée. La précision pour la classe 1 était de 62%, une augmentation notable par rapport aux 78% de la version non optimisée. Le rappel pour la classe 0 était de

94%, montrant une stabilité par rapport à la version non optimisée, tandis que le rappel pour la classe 1 était également de 62%, une amélioration significative par rapport aux 44% précédents.

L'optimisation du modèle SVM a conduit à des améliorations significatives dans la prédiction de l'infection au COVID-19 à partir des données cliniques fournies. Les prédictions erronées ont été réduites et les performances globales du modèle se sont améliorées en termes de précision et de rappel pour les deux classes. Ces améliorations sont essentielles pour une prédiction plus précise et fiable de l'infection au COVID-19, ce qui peut contribuer à une meilleure gestion de la pandémie et à la prise de décisions éclairées en matière de santé publique.

6.4. Interprétation des résultats et implications pratiques

L'interprétation des résultats de notre étude sur la prédiction de l'infection au COVID-19 à partir de données cliniques révèle des informations essentielles sur les performances des algorithmes de classification et l'importance des facteurs cliniques. Ces résultats ont des implications pratiques significatives pour la prise en charge des patients et la surveillance de la maladie.

En ce qui concerne les performances des algorithmes de classification, nous avons observé que SVM et KNN ont obtenu les meilleurs résultats en termes de précision globale et de score F1. Ces deux algorithmes ont montré une capacité relativement élevée à prédire l'infection au COVID-19. Cependant, il est important de noter que tous les algorithmes ont montré des résultats variables en termes de rappel (Recall) pour la classe positive (infection au COVID-19), ce qui indique une certaine difficulté à identifier correctement tous les cas positifs. Cette information est cruciale pour les décisions cliniques, car le rappel est une mesure importante pour minimiser les faux négatifs et identifier les cas infectés.

En ce qui concerne les facteurs cliniques contribuant à la prédiction de l'infection au COVID-19, notre analyse a révélé que certains paramètres étaient plus significatifs que d'autres. Par exemple, l'âge ainsi que les variables de type sanguines ont montré une forte corrélation avec la prédiction de l'infection. Ces résultats soulignent l'importance de ces facteurs dans l'évaluation du risque d'infection au COVID-19 et peuvent guider les efforts de dépistage et de surveillance des patients.

Sur la base de ces résultats, plusieurs implications pratiques peuvent être envisagées. Tout d'abord, l'utilisation de modèles de prédiction basés sur des algorithmes tels que SVM et KNN

peut être bénéfique pour identifier les individus à risque d'infection au COVID-19. Ces modèles peuvent être intégrés dans des outils de dépistage et d'évaluation du risque pour soutenir les décisions médicales et la gestion des ressources.

De plus, la prise en compte des facteurs cliniques importants, tels que l'âge, les symptômes spécifiques et les antécédents médicaux, dans les protocoles de dépistage et de surveillance peut améliorer l'efficacité des mesures de contrôle et de prévention. Par exemple, l'identification des patients présentant des symptômes spécifiques et des antécédents médicaux pertinents peut permettre une intervention précoce et une prise en charge appropriée.

Cependant, il est important de noter que notre étude présente certaines limitations, telles que la disponibilité limitée de données cliniques spécifiques et l'utilisation d'un seul ensemble de données. Des études supplémentaires et des validations externes sont nécessaires pour confirmer nos résultats et élargir l'applicabilité des modèles de prédiction proposés.

En conclusion, notre étude démontre que l'utilisation d'algorithmes de classification et l'analyse des facteurs cliniques peuvent contribuer à la prédiction de l'infection au COVID-19. Ces résultats ont des implications pratiques pour améliorer la prise en charge des patients, la surveillance de la maladie et l'allocation des ressources. Cependant, des recherches supplémentaires sont nécessaires pour approfondir ces résultats et les appliquer dans des contextes réels de soins de santé.

7.Conclusion

En conclusion, après avoir examiné les matrices de confusion, les rapports de classification et les courbes d'apprentissage, nous pouvons conclure que le SVM se distingue comme l'algorithme de classification le plus performant. Il a démontré une précision supérieure dans l'identification des individus infectés par le virus et a également présenté une meilleure capacité de généralisation et d'évitement du surajustement par rapport aux autres modèles. L'optimisation du modèle SVM a entraîné des améliorations significatives dans la prédiction de l'infection au COVID-19 à partir des données cliniques fournies. Les prédictions erronées ont été réduites et les performances globales du modèle se sont améliorées en termes de précision et de rappel pour les deux classes. Ces améliorations sont essentielles pour une prédiction plus précise et fiable de l'infection au COVID-19.

CONCLUSION GENERALE

Nous avons pu explorer les principaux résultats et contributions de notre étude sur la prédiction de l'infection au COVID-19 à partir de données cliniques. En récapitulant nos principaux points, nous pouvons conclure ce travail de la manière suivante :

Notre recherche visait à identifier l'algorithme de classification le plus performant pour la détection précise de l'infection au COVID-19. Grâce à une analyse approfondie des performances des différents algorithmes, nous avons constaté que le SVM (Support Vector Machine) s'est avéré être l'algorithme le plus performant, démontrant une capacité supérieure à identifier avec précision les individus infectés par le virus. Cette découverte est d'une importance capitale pour la mise en place de mesures de prévention et de traitement efficaces.

En outre, nous avons exploré les techniques d'optimisation des hyperparamètres pour améliorer les performances du modèle SVM. En ajustant les paramètres du modèle, nous avons pu optimiser sa capacité à prédire l'infection au COVID-19 à partir des données cliniques. Cela démontre l'importance de l'optimisation des modèles pour obtenir des résultats de prédiction plus précis et fiables.

Nos résultats et conclusions apportent des contributions significatives à la recherche sur la prédiction de l'infection au COVID-19. En identifiant le SVM comme l'algorithme de classification le plus performant, nous fournissons aux professionnels de la santé et aux décideurs des informations précieuses pour la prise de décisions éclairées en matière de dépistage et de prévention de la propagation du virus.

De plus, notre étude met en évidence l'importance de l'utilisation de données cliniques pour la prédiction de l'infection au COVID-19. En exploitant ces données, nous avons pu développer des modèles prédictifs fiables, ce qui renforce l'idée que les informations cliniques jouent un rôle clé dans la lutte contre la pandémie et la protection de la santé publique.

En conclusion, notre recherche a permis de déterminer le SVM comme l'algorithme de classification le plus performant pour la prédiction de l'infection au COVID-19 à partir de données cliniques. Ces résultats offrent des perspectives prometteuses pour améliorer les capacités de dépistage et de prévention de cette maladie grave. Cependant, il est important de

reconnaître les limites de notre étude, telles que les contraintes des données disponibles et les limites des méthodes utilisées.

Pour la recherche future, il est recommandé d'explorer d'autres techniques d'apprentissage automatique, d'envisager des ensembles de données plus vastes et diversifiés, et d'intégrer d'autres caractéristiques cliniques pertinentes. De plus, il est essentiel de continuer à évaluer et à optimiser les modèles à mesure que de nouvelles données deviennent disponibles, afin d'améliorer la précision et la fiabilité des prédictions.

En somme, cette recherche offre des perspectives prometteuses pour la prédiction de l'infection au COVID-19, mais elle représente également une base solide pour de futures études visant à améliorer la détection, la prévention et le contrôle de cette maladie. En combinant les connaissances acquises avec les avancées technologiques et les progrès scientifiques à venir, nous sommes convaincus que nous pourrons faire face plus efficacement aux défis posés par cette pandémie mondiale.

Bibliographie

[1] HAN, Jiawei, KAMBER, Micheline, et PEI, Jian. Data mining concepts and techniques third edition. University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University, 2012.

[2] PROCESSUS KDD, [En ligne]. Available : https://www.researchgate.net/figure/Representation-standard-du-processus-KDD-decouverte-de-connaissances-dans-une-base-de_fig5_271764486, Consulté le 29/03/2023.

[3] Lynda SELLAMI, Approche Data Mining pour la Détection d'Intrusions, Université Abderrahmane Mira de Bejaia.

[4] technique datamining, [En ligne]. Available : https://www.wideskills.com/data-mining-tutorial/data-mining-techniques, Consulté le 27/03/2023.

[5] architecture datamining, [En ligne]. Available : https://www.researchgate.net/figure/Architecture-of-a-Typical-Data-Mining-System-1_fig2_290212859, Consulté le 27/03/2023.

[6] Ian H. Witten, Eibe Frank and Mark A. Hall,Data Mining: Practical Machine Learning Tools and Techniques, Third Edition,2011,Elsevier,https://doi.org/10.1016/C2009-0-19715-5.

[7] DENOYER, Ludovic et GALLINARI, Patrick. Bayesian network model for semi-structured document classification. Information processing & management, 2004, vol. 40, no 5, p. 807-827.https://doi.org/10.1016/j.ipm.2004.04.009.

[8] classification supervise, [En ligne]. Available : https://fr.linedata.com/quest-ce-que-lapprentissage-supervise, Consulté le 04/04/2023.

[9] Lydia BATACHE,Samia DRICI, Etude et recherche bibliographique sur les bibliographique sur les méthodes de classification, 2019, ummto.

[10] ARARBI Assia,AOUAKLI Farida,Application des techniques de datamining pour la classification automatique des données, 2020 / 2021,ummto.

[11] Mariam Tanana. Evaluation formative du savoir-faire des apprenants à l'aide d'algorithmes de classification : application à l'électronique numérique. Autre [cs.OH]. INSA de Rouen, 2009. Français. ffNNT : 2009ISAM0007ff. fftel-00442930v2f.

[12] kan, [En ligne]. Available : https://glassus.github.io/premiere_nsi/T4_Algorithmique/4.7_Algorithme_KNN/07_Algorithme_KNN/, Consulté le 04/04/2023.

[13] svc, [En ligne]. Available : https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323, Consulté le 06/04/2023.

[14] Aaron Hertzmann, David J. Fleet and Marcus Brubaker, Machine Learning Engineer Nanodegree Supervised Learning Project: Finding Donors For CharityML, 2015.

[15] Isabelle Bernard, Comparaison des performances prédictives de deux algorithmes construisant des forêts aléatoires,2018, HEC MONTRÉAL.

[16] Sebastien Gadat ,´Laboratoire de Statistique et Probabilites´,UMR 5583 CNRS-UPS,https://www.math.univ-toulouse.fr/~gadat/Ens/M2SID/11-m2-Random_Forests.pdf.

[17] apprentissage supervise et non supervise, [En ligne]. Available : https://penseeartificielle.fr/comparer-algorithme-machine-learning-regression-classification/, Consulté le 04/04/2023.

[18] DJAFRI Razik,DATA MINING Classification par apprentissage supervisé pour prédire le remboursement d'un crédit,2018/2019,Université Abderahmane Mira de Béjaia.

[19] Kateb Nabila, Une approche multi agents pour Le Data Mining, 2010/2011, Université Larbi Ben M'hidi Oum El Bouaghi.

[20] Chamek Linda,Localisation des mobiles par une stratégie de prédiction,2010/2011, Uniersité M'hamed Bougara Boumerdes.

[21] RobustScaler, [En ligne]. Available : https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html, Consulté le 01/06/2023.

[22] CHAKRABARTI, Soumen, NEAPOLITAN, Richard E., PYLE, Dorian, et al. Data mining: know it all. Morgan Kaufmann, 2008.

[23] MÜLLER, Andreas C. et GUIDO, Sarah. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.", 2016.

[24] PolynomialFeatures, [En ligne]. Available : https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html#examples-using-sklearn-preprocessing-polynomialfeatures, Consulté le 01/06/2023.

[25] GUYON, Isabelle et ELISSEEFF, André. An introduction to variable and feature selection. Journal of machine learning research, 2003, vol. 3, no Mar, p. 1157-1182.

[26] RASCHKA, Sebastian. Python machine learning. Packt publishing ltd, 2015.

[27] ZHENG, Alice et CASARI, Amanda. Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc.", 2018.

[28] T. D. Science, «Feature Engineering Techniques for Machine Learning, » [En ligne]. Available : https://towardsdatascience.com/feature-engineering-techniques-for-machine-learning-48a66d15b48b,Consulté le 03/06/2023.

[29] LIU, Huan et MOTODA, Hiroshi. Feature selection for knowledge discovery and data mining. Springer Science & Business Media, 2012.

[30] K. H. &. P. J. Kim, A review of feature selection methods in healthcare applications. Healthcare informatics research, 2017, pp. 68-76.

[31] KRAVCHENKO, Yuri, DAKHNO, Natalia, LESHCHENKO, Olga, et al. Machine Learning Algorithms for Predicting the Results of COVID-19 Coronavirus Infection. In : IT&I Workshops. 2020. p. 371-381.

[32] Moulaei, K., Shanbehzadeh, M., Mohammadi-Taghiabad, Z. et al. Comparing machine learning algorithms for predicting COVID-19 mortality. BMC Med Inform Decis Mak 22, 2 (2022). https://doi.org/10.1186/s12911-021-01742-0.

[33] Muhammad LJ, Islam MM, Usman SS, Ayon SI. Predictive Data Mining Models for Novel Coronavirus (COVID-19) Infected Patients' Recovery. SN Comput Sci. 2020;1(4):206. doi: 10.1007/s42979-020-00216-w. Epub 2020 Jun 21. PMID: 33063049; PMCID: PMC7306186.

[34] dataset, [En ligne]. Available : https://www.kaggle.com/datasets/einsteindata4u/covid19,Consulté le 22/04/2023.

[35] randomizedsearchcv, [En ligne]. Available : https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html.

[36] anaconda, [En ligne]. Available : https://docs.anaconda.com/.

[37] Mendez, K.M., Pritchard, L., Reinke, S.N. et al. Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. Metabolomics 15, 125 (2019). https://doi.org/10.1007/s11306-019-1588-0.

[38] MAMEN ABDELKARIM, Développement d'une architecture CNN pour la classification des images radiologiques d'infections pulmonaires, Université Mohamed Khider–BISKRA ,2020-2021.

[39] pandas, [En ligne]. Available : https://pandas.pydata.org/docs/. Consulté le 09/06/2023.

[40] matplotlib, [En ligne]. Available : https://matplotlib.org/stable/contents.html. Consulté le 09/06/2023.

[41] scikitlearn, [En ligne]. Available : https://scikit-learn.org/stable/documentation.html. Consulté le 09/06/2023

[42] SEABORN, [En ligne]. Available : https://seaborn.pydata.org/. Consulté le 09/06/2023

[43] Jason Brownlee, "How to Learning Curves for Machine Learning Model Performance", Machine Learning Mastery (2019). Disponible en ligne : https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

[44] OpenAI, Pandas Cheat Sheet [En ligne]. Available: https://pandas.pydata.org/Pandas Cheat Sheet.pdf. Consulté le 01/06/2023.

Résumé

La prédiction de l'infection au COVID-19 à partir de données cliniques revêt une grande importance, car elle permet d'identifier les individus à risque, de prendre des mesures préventives adaptées et de mieux allouer les ressources de santé. Nous démontrons dans ce mémoire notre étude sur l'efficacité des algorithmes de classification dans la prédiction de l'infection au COVID-19 à partir de données cliniques. En optimisant les hyperparamètres, nous avons constaté une amélioration des performances prédictives du modèle SVC. Ces résultats revêtent une importance cruciale pour la gestion de la pandémie et ouvrent de nouvelles perspectives pour les recherches à venir dans ce domaine.

Mots clés : COVID-19, classification, hyperparamètres, SVC

Abstract

The prediction of COVID-19 infection based on clinical data is of great importance as it enables the identification of individuals at risk, implementation of appropriate preventive measures, and better allocation of healthcare resources. In this paper, we demonstrate our study on the effectiveness of classification algorithms in predicting COVID-19 infection using clinical data. By optimizing the hyperparameters, we observed an improvement in the predictive performance of the SVC model. These findings are of paramount significance for pandemic management and open new avenues for future research in this field.

Keywords : COVID-19, classification, hyperparameters, SVC