Séminaire Mathématique de Béjaïa (LaMOS)
Volume 18, 2020, pp. 65 - 84

4

Application des Méthodes d'Apprentissage dans la Prédiction du Diabète de Type 2.

Mehidi D., Medjoudj S., Adel-Aissanou K. et Aïssani D.

Research Unit **LaMOS** (Modeling and Optimization of Systems) and Faculty of Exact Sciences, Bejaia University, Bejaia 06000, Algeria
lamos_bejaia@hotmail.com , ak_adel@yahoo.fr

Résumé L'utilisation des systèmes experts et les techniques dites intelligentes en diagnostic médical ne cesse d'augmenter graduellement. L'apprentissage automatique est une méthode parmi d'autres utilisées dans le diagnostic médical. Cet article présente une approche d'apprentissage supervisé basé sur les algorithmes K-Plus Proche Voisin (K-PPV), SVM, Naïve Bayes, Radom Forest, Décision Tee classifier (Arbre de décision), pour reconnaitre les personnes susceptibles de développer un diabète en utilisant deux bases de données différentes, à savoir, celles du CHU et du cabinet privé du Dr Djamel MEHIDI. Les performances des classifieurs ont été comparées en fonction du taux de précision, temps d'exécution. Les plus hauts taux de classification obtenus par l'application de Radom Forest et Naïve Bayes sont respectivement 86% et 85%, en appliquant l'approche 10-folds cross-validation.

Mots clés : Prédiction du diabète, K-Plus Proche Voisin (K-PPV), SVM, Naïve Bayes, Radom Forest, Arbre de décision.

4.1 Introduction

Le diabete mellitus, ou diabète sucré, aussi appelé simplement diabète est l'une des maladies majeures de notre monde moderne. En 2019, la Fédération Internationale pour le Diabète a estimé que 463 millions d'adultes dans le monde sont diabétiques et que la maladie est directement à l'origine de 4.2 millions de morts [7]. La prévalence de la maladie dans le monde est prédite d'évoluer de 9.3% à 10.9% (700 millions) d'ici 2045. Voir la figure4.1.

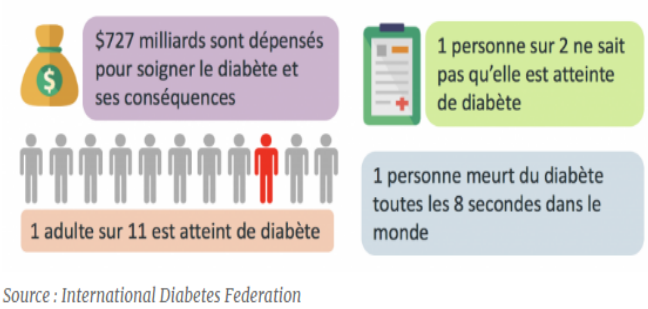

FIGURE 4.1: Le Diabète dans le monde

En Algérie, sa prévalence en 2022 est estimée à 14.4%, soit près de 4.5 millions de personnes. C'est pourquoi, il y a un véritable besoin de sensibilisation et de prévention de cette

maladie, encore trop ignorée à ce jour.

Nous entendons chaque jour le terme intelligence artificielle, l'IA définie, plus généralement, comme la capacité d'une machine capable d'agir, par elle-même ou sous le contrôle de l'homme à reproduire des actions ou des fonctions qui sont habituellement celles des êtres humains. Aujourd'hui, nous la retrouvons dans nos machines informatiques, objets connectés, applications, réseaux sociaux, transports, et dans le secteur médical. L'application de l'IA à la médecine offre une perspective essentielle à l'essor de ces nouvelles technologies, qu'il s'agisse de renforcer le lien entre patients et médecins, de poser des diagnostics plus rapides et plus précis, ou encore d'optimiser la création de nouveaux traitements. L'innovation a pour objectif de combattre la mort et la maladie. Quoi de plus noble ?

L'IA permet aux médecins de gagner du temps en laissant la machine analyser elle-même les données et fournir des estimations. Le but à plus long terme : réussir à prédire de nombreuses maladies, afin que les médecins puissent intervenir le plus tôt possible.
Le diagnostic médical est un processus de classification. L'utilisation de l'informatique pour la réalisation de cette classification devient de plus en plus fréquente. Même si la décision de l'expert est le facteur le plus important lors du diagnostic, les systèmes de classification fournissent une aide substantielle, car elles réduisent les erreurs dues à la fatigue et le temps nécessaire pour le diagnostic. Actuellement, la plupart des hôpitaux modernes sont bien équipés avec des dispositifs de collecte de données. L'augmentation du volume de données entraîne des difficultés à extraire des informations utiles pour l'aide à la décision. Les méthodes traditionnelles d'analyse de données peuvent être complètes par l'utilisation de méthodes dites intelligentes.

L'apprentissage est la capacité de s'améliorer avec l'expérience, de se rappeler les décisions antérieures et les résultats afin de faire de meilleurs choix à l'avenir dans des situations similaires. L'apprentissage automatique est une discipline de l'intelligence artificielle. L'apprentissage automatique cherche à trouver le moyen de construire des programmes informatiques qui s'améliorent automatiquement avec l'expérience. L'une des métaphores utilisées dans le domaine de l'apprentissage automatique et qui considère la résolution de problèmes comme un typed'apprentissage qui, une fois le problème résolu, est capable de reconnaitre la problématique et réagir en utilisant la stratégie apprise.

Dans ce travail, nous nous intéresserons à la prédiction du diabète type 2 qui est un dysfonctionnement du système de régulation de la glycémie. Beaucoup de travaux ont été menés afin d'effectuer la classification ou le diagnostic du diabète. Hung-Chun Lin et al. ont obtenu une précision de 62.8% en utilisant un ensemble de cellules mémoires qui ont subit un apprentissage leur permettant d'effectuer une classification et cela grâce à la méthode K-plus proche voisin, dans la classification du diabète. Une autre méthode a utilisé un réseau de neurones pour

effectuer la classification du diabète. L'apprentissage du réseau de neurones a été effectué pour chaque patient sur une période de 24h et le test a été effectué sur une autre période de 24h. L'erreur de prédiction sur les patients a été plus faible que $10mg/dl$. Keller et al ont proposé en 1985 la classification du diabète qui est effectuée par l'algorithme K-ppv flou considérée comme étant l'une des plus importantes méthodes parmi les algorithmes non-paramétriques. Les auteurs ont proposé le classifieur K-ppv flou. Pour cela, l'algorithme K-ppv flou alloue au vecteur de données un degré d'appartenance à une classe donnée. Cet algorithme a permis d'améliorer le taux de classification de 6.42% par rapport aux autres algorithmes classiques.

La méthode appliquée dans ce travail pour la prédiction du diabète de type 2 est l'application d'algorithmes à apprentissage supervisé. Notre méthode est une évolution d'algorithmes étudiés. Nous effectuons un apprentissage supervisé sur la base de données du CHU de Bejaia et du cabinet médical de Dr Mehidi. En plus, nous améliorons le meilleur algorithme qui donnera comme résultat une classification des patients qui peuvent être susceptibles d'être atteint du diabète et qui a comme but l'aide à la décision du médecin en termes de temps d'examination et de taux de précision du résultat.

4.2 Travaux antérieurs

Dans [8], Les auteurs ont proposé un modèle de prédiction du diabète de type1 basé sur les réseaux de neurones, ce modèle est testé sur 22 patients (12 femmes et 10 hommes). Dont 14 sont diabétiques et 8 en bonne santé. Parmi les 22, 16 patients ont suivi un traitement à injection d'insuline et 6 avec une pompe. L'évolution du niveau de glucose a été enregistrée pour chaque patient, toutes les 5 minutes pendant 3 jours. Chaque patient a eu une vie normale avec repas et activités autant à la maison qu'au bureau. Trois cas ont été pris en compte : un patient avec un traitement classique à base d'injection (P1), un patient avec un traitement basé sur la pompe à insuline (P2) et une personne en bonne santé (P3). Pour P1, une grande variabilité du niveau de glucose a été mesurée. Pour P2 la variabilité est plus réduite mais encore importante par rapport à la personne en bonne santé. L'apprentissage du réseau de neurones a été effectué pour chaque patient sur une période de 24h et le test a été effectué sur une autre période de 24h. L'erreur de prédiction sur les 3 patients a été plus faible que 10mg/dl. Hung-Chun Lin et al [9] ont utilisé l'algorithme AIRS2 (Artificial Immune Recognition System) qui est inspiré du système immunitaire pour la prédiction du diabète du type 2 avec ensemble de cellules mémoires qui ont subit un apprentissage leur permettant d'effectuer une classification et cela grâce à l'algorithme AIRS. Les auteurs ont combiné une hybridation de deux algorithmes, le premier est l'AIRS2 qui est une évolution de l'algorithme AIRS. Le deuxième algorithme est le K-plus proche voisin flou. L'algorithme AIRS2 effectuera un apprentissage supervisé sur la base de données du diabète et donnera comme résultat un ensemble de cellules mémoires. Le K-plus proche voisin effectuera la classification de la base de test en utilisant l'ensemble de cellules mémoires généré dans la phase d'apprentissage. Keller et al [10], ont proposé en 1985, le classifieur K-ppv flou pour palier aux limitations du K-ppv.L'algorithme K-ppv flou

alloue au vecteur de données un degré d'appartenance à une classe donnée. Le principe est d'allouer le degré d'appartenance à une classe en fonction de la distance du vecteur de ces K-ppv et de l'appartenance de ces voisins à la classe.la classification est effectuée usuellement par l'algorithme K-ppv. L'algorithme a été remplacé par le K-ppv flou et est l'une des plus importantes méthodes parmi les algorithmes non paramétriques, qu'il ne spécifie pas le degré d'appartenance du vecteur à la classe qui lui a été attribué.

4.3 Notre proposition

Les algorithmes d'apprentissage automatique peuvent nous aider à détecter l'apparition du diabète. La détection précoce du diabète peut réduire les risques pour la santé du patient. Les médecins, les patients et les proches du patient peuvent bénéficier des résultats de la prédiction. Dans les milieux cliniques à faibles ressources, il est nécessaire de prédire l'état du patient après l'admission pour répartir les ressources de manière appropriée. Dans l'étude qui suit, l'objectif principal est d'appliquer ces différents algorithmes de classification sur des données concernant le risques de développer un diabète de type 2 récoltés au niveau de deux infrastructures différentes dans le but d'aboutir à un résultat d'aide à la prédiction de ce type de diabète.
En effet, avant de se focaliser sur les outils, nous avons procédé aux points suivant :

1. Collecter les données, et les rendre propre (renseigner les valeurs vides, supprimer et remplacer les valeurs aberrantes).
2. Sélectionner les variables pertinentes (table de corrélation).
3. Application de la fonction train_test_split pour découper nos données en données d'apprentissage et test.
4. Apprentissage et évaluation, en appliquant la technique cross validation.
5. Evaluation de performance de chaque algorithme.
6. Optimisation de l'algorithme RandomForest (chercher les bonnes valeurs pour chaque paramètre) via la fonction RandomizedSearchCV.
7. Exportation du model (meilleur algorithme + prétraitement) sous format binaire avec la librairie pickle.
8. Développement d'une application web en python (à base du framework Flask) qui permet aux visiteurs de saisir leurs données et de prédire s'ils sont susceptibles de développer un diabète ou pas.

La prévalence du diabète de type 2 est sous estimée car cette anomalie glycémique asymptomatique peut évoluer de façon insidieuse et silencieuse pendant de nombreuses années avant que le diagnostic ne soit porté, le travail mentionné est utile aux médecins comme outil d'aide à la décision.

4.3.1 Méthodes

Pour répondre à cette recherche comparative, nous utiliserons les deux bases de données issues du Centre Hospitalo-Universitaire de Bejaia et une du cabinet médical du Dr Djamel MEHIDI qui contiennent respectivement des informations de 268 et 500 patients tous genres confondus (femmes et hommes). Elles seront utilisées dans la classification des algorithmes d'apprentissage supervisé tel que Naives Bayes, le Plus Proche Voisin et SVM.

4.3.2 Outils utilisés

Python : Est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est facile à interpréter, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. L'interpréteur Python et sa vaste bibliothèque standard sont disponibles librement, sous forme de sources ou de binaires, pour toutes les plateformes majeures depuis le site Internet https ://www.python.org/ et peuvent être librement redistribués. Ce même site distribue et pointe vers des modules, des programmes et des outils tiers. Enfin, il constitue une source de documentation. L'interpréteur Python peut être facilement étendu par de nouvelles fonctions et types de données implémentés en C ou C^{++} (ou tout autre langage appelable depuis le C). Python est également adapté comme langage d'extension pour personnaliser des applications [1].
Scikit-Learn : Est une bibliothèque qui fournit une gamme d'algorithmes d'apprentissage supervisés et non supervisés via une interface cohérente en Python. La vision de la bibliothèque est un niveau de robustesse et de support requis pour une utilisation dans les systèmes de production. Cela signifie qu'il faut se concentrer sur des préoccupations telles que la simplicité d'utilisation, la qualité du code, la collaboration, la documentation et les performances [2].
Pandas : [3] Est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques. Pandas est un logiciel libre sous licence. Les principales structures de données sont :

- Les séries (pour stocker des données selon une dimension - grandeur en fonction d'un index).
- Les DataFrames : pour manipuler des données aisément et efficacement avec des index pouvant être des chaines de caractères (stocker des données selon 2 dimensions - lignes et colonnes).
- Les Panels pour représenter des données selon 3 dimensions.
- Format de lecture et écriture des données structurées en mémoire depuis et vers différents formats : fichiers CSV, fichiers textuels, fichier du tableur Microsoft Excel, base de données SQL.

Spyder : [4] Est un environnement de développement pour Python, libre et multiplateforme (Windows, Mac OS, GNU/Linux), il intègre de nombreuses bibliothèques d'usage scientifique.

Spyder a un ensemble unique de fonctionnalités - multiplateforme, open-source, écrit en Python et disponible sous une licence non-copyleft. Spyder est extensible avec des plugins, comprend le support d'outils interactifs pour l'inspection des données et incorpore des instruments d'assurance de la qualité et d'introspection spécifiques au code Python.
Il offre une combinaison unique de fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les capacités de visualisation d'un package scientifique.
Python-Seaborn (Visualisation de données statistiques pour python) : Seaborn est une bibliothèque pour créer des graphiques statistiques attrayants et informatifs en Python[5].

4.3.3 Les données

C'est la totalité des éléments pris en considération, sur lesquels nous utiliserons l'apprentissage automatique dans le but d'appliquer la classification de l'apprentissage supervisé dans la prédiction du diabète de type 2.
La collecte de données a été effectuée au niveau de deux infrastructures différentes ; l'une au Centre Hospitalo Universitaire de Bejaia (CHU), au service de médecine interne dirigé par le professeur BOUALI, la seconde au niveau du cabinet médical privé de Dr Djamel MEHIDI, où nous avons effectué une collecte de données respectivement de 267 et 500 patients. Les problèmes d'apprentissage sont énoncés sous forme de données. Ces séries caractérisant une série d'instances du phénomène à apprendre, que l'on nomme patient. Chaque patient P est constitué d'une description D et d'une sortie S

$$P = (D, S)$$

Où :

$D \in X = \{\text{Genre}, \text{Âge}, \text{Groupe sanguin}, \text{Pression artérielle}, \text{IMC}, \text{Antecedants familliaux}, \text{Sportif}, \text{Fumeur}\}$

$$S \in Y = \{0, 1\}$$

On nomme respectivement les ensembles X et Y l'espace d'entrée et l'espace de sortie. La description $X = \{x_1, x_2, \ldots.., x_n\}$ de l'ensemble des données s'avère être un vecteur de valeurs réelles de dimension n, où n est le nombres des attributs de patients . Selon le phénomène représenté, un attribut peut être numérique (par exemple, âge) ou de type libraire (par exemple, le genre a été déterminé arbitrairement par la valeur 0 équivaut à « homme » et 1 équivaut à « femme »). La sortie Y du patient détermine sa classe d'appartenance. Les problèmes étudiés dans ces études seront tous des problèmes de classification binaire, c'est-à-dire des problèmes de classification à seulement deux classes. Pour chaque problème, on détermine arbitrairement que les patients appartenant à une classe sont des patients malade « 0 » et que les autres patients ne sont pas malade « 1 ».
Durant la phase de collecte des données, nous avons constaté que certaines informations étaient manquantes notamment dans l'âge et qu'il existait des valeurs aberrantes particulièrement

dans la pression artérielle. C'est pourquoi nous avons dû effectuer un prétraitement avant de commencer notre étude. Les ensembles de données contiennent 10 variables particulières qui ont été considérées comme des facteurs à risque élevés de développer un diabète. Cette base de données contient des patients de tout genre et âge confondus. Les valeurs sont de différents types ; le tableau 4.1 la description de notre ensemble de données :

Paramètre physiologique	Description	Valeurs	Analyse de données
Genre	Homme ou Femme	0 ou 1	Homme :397(M) Femme : 371 (F)
Âge	Age de la personne		Âge 5 à 20 ans : 30 Âge 21 à 35 ans : 131 Âge 36 à 50 ans : 142 Âge 51 à 78 ans : 97
Groupe sanguin	Groupe sanguin	A ou B ou AB ou O	A : 242 B : 144 AB : 164 O : 218
IMC	$Imc = ((taille) * 2/poids)$	Bon ou Mauvais	
Activité physique	Oui Non	1 ou 0 ou Non renseigné	Oui : $81(1)$ Non : $137(0)$ Non renseigné :(551)
Pression artérielle	Optimale Normale Elevée	≥ 14 < 14	767 4
Fumeur	Oui Non	1 ou 0 ou Non renseigné	Oui : $234(1)$ Non : $534(0)$ Non renseigné :(1)
Antécédents familiaux	Oui Non	1 ou 0 ou Non renseigné	Oui : 231 Non : 227 Non renseigné :310
Outcome	Si une personne est diabétique ou pas	0 ou 1	Non : $335(1)$ Oui : $432(0)$

TABLE 4.1: Analyse des données collectées

4.3.4 Préparation et nettoyage des données

Afin de sélectionner les variables les plus significatives pour les modèles, nous commençons par épurer les données : élimination de cer-taines valeurs, traitement des valeurs manquantes, détection des va-leurs aberrantes et bien d'autres types d'incohérences qui peuvent gêner l'analyse.

La figure4.2 représente une description des données obtenu

description :	Age	Pression_arterielle	Insuline	IMC	Glucose	\
count	764.000000	768.000000	768.000000	768.000000	768.000000	
mean	33.267016	70.928385	79.799479	31.992578	122.572917	
std	11.782850	32.322266	115.244002	7.884160	32.458296	
min	21.000000	0.000000	0.000000	0.000000	0.000000	
25%	24.000000	64.000000	0.000000	27.300000	100.000000	
50%	29.000000	72.000000	30.500000	32.000000	118.500000	
75%	41.000000	80.000000	127.250000	36.600000	144.000000	
max	81.000000	594.000000	846.000000	67.100000	199.000000	

	Sportif	Fumeur	Malade
count	768.000000	768.000000	768.000000
mean	0.397135	0.304688	0.436198
std	0.489623	0.460575	0.496236
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	1.000000	1.000000	1.000000

FIGURE 4.2: Description des deux bases de données

D'après la représentation des données, on constate que :

– Des effectifs très grands (768 patients) ;
– La moyenne de chaque donnée est calculée ;
– Les bornes min et max sont affichées ;
– Les quartiles 25% 50% 75% sont donnés.

Pendant la récolte des données, nous avons constaté que certaines valeurs d'âge étaient manquantes et que des valeurs aberrantes de pression artérielle existaient, ce qui poserait des problèmes pendant la phase d'apprentissage. Nous avons remédié à cette situation en utilisant 2 méthodes :

– **Remplacer les valeurs manquantes d'âge**
  Pour ce faire, nous avons créé une fonction calculer moyenne qui compte le nombre de ligne vide ou égale à zéro de la colonne âge, une fois le nombre de case trouvé ; la moyenne globale d'âge des patients sera calculée (résultat retourné : moyenne âge $=$ 33 ans) puis introduite dans les cases vides ou égale à zéro.
– **Remplacer les valeurs abérrantes de la pression artérielle**
  Nous avons fait appel à la méthode *data.loc* dans le but de repérer les valeurs supérieures à 140 mm hg, le résultat est donné dans la figure4.3

N° Patient	Pression artérielle
13.0	260
19.0	470
24.0	594
767.0	270

FIGURE 4.3: Données abérrantes de la pression artérielle

Une fois le résultat retourné, nous les avons remplacés par des 0.0 puis calculer la moyenne globale de cette dernière : *moyenne_Pression_arterielle*, une fois le résultat retourné nous les avons remplacé par cette nouvelle valeur.

4.3.5 Choix des attributs

Afin de sélectionner les variables les plus significatives pour les modèles, nous avons réalisé une étude des corrélations sur notre base de données. cette étude est nécessaire afin de répertorier les différentes relations entre ces données. La corrélation étudie l'intensité de la liaison qui peut exister entre ces variables [6]. Le coefficient de corrélation entre deux variables aléatoires réelles X et Y ayant chacune une variance est définie par r

$$r = \frac{Cov(X,Y)}{\sigma_x \sigma_y}$$

Où $Cov(X,Y)$ désigne la covariance des variables X et Y et
σ_x, σ_y leurs écarts types respectifs. Le tableau donné dans la figure4.4, représente les relations entre les différentes variables :

	Genre	Age	Groupe_sanguin	Pression_arterielle	IMC	Antécedants_familliaux	Sportif	Fumeur	Malade
Genre	1.000000	0.013412	0.008795	-0.039266	0.001274	0.093322	0.035481	-0.022866	0.111236
Age	0.013412	1.000000	-0.063793	-0.159273	0.038377	0.074763	-0.017677	0.012488	0.263138
Groupe_sanguin	0.008795	-0.063793	1.000000	0.016537	0.008306	-0.016168	-0.024722	-0.004888	-0.054919
Pression_arterielle	-0.039266	-0.159273	0.016537	1.000000	-0.136239	-0.041651	0.000341	-0.046338	-0.086456
IMC	0.001274	0.038377	0.008306	-0.136239	1.000000	0.147906	0.038119	-0.007347	0.273021
Antécedants_familliaux	0.093322	0.074763	-0.016168	-0.041651	0.147906	1.000000	0.028286	-0.025816	0.489008
Sportif	0.035481	-0.017677	-0.024722	0.000341	0.038119	0.028286	1.000000	-0.022720	0.042712
Fumeur	-0.022866	0.012488	-0.004888	-0.046338	-0.007347	-0.025816	-0.022720	1.000000	-0.023219
Malade	0.111236	0.263138	-0.054919	-0.086456	0.273021	0.489008	0.042712	-0.023219	1.000000

FIGURE 4.4: Table de corrélation des deux bases de données

– **Interprétation du tableau**
  Il existe une forte corrélation entre les différentes variables telle que ; malade et genre, malade et âge, malade et IMC, malade et antécédents familiaux, malade et sportif ;
  Il existe une faible corrélation entre les différentes variables telle que ; malade et groupe sanguin, malade et pression artérielle, malade et fumeur.
– **Sélection de variables**
  Cette table de corrélation préposée pour le problème de sélection des variables à hauts risques de développer un diabète peut être décomposée en deux classes ; une classe à forte corrélation qui comporte des variables à hauts risques de développer un diabète : genre, âge, IMC, antécédents familiaux et sportif, une autre classe à faible corrélation qui comporte à son tour des variables à faible risque : pression artérielle, groupe sanguin ainsi que fumeur. Dans ce qui suit, dans notre système d'apprentissage, nous utiliserons les variables ayant une forte corrélation entre elles car celles ayants une faible corrélation n'ont pas un grand impact dans la prédiction du diabète.

4.3.6 Apprentissage

Notre but est une aide à la décision dans la prédiction de développement d'un diabète chez une personne. Nous nous intéresserons à la classification dans l'apprentissage supervisé, il est essentiel dans tout programme d'apprentissage d'avoir un mécanisme qui sépare l'ensemble des données en deux groupes. Afin d'assurer ce mécanisme, nous aurons recours à la méthode de Cross Validation.
Data-train :(ensemble d'apprentissage) qui consiste à faire un apprentissage des données sur elle-même. Elle comprend la partie apprentissage et la partie validation.
Data-test : (ensemble test) qui consiste à tester les données restantes dans la Data-train, pour le test final de notre modèle.

Apprentissage et test

Dans un premier temps, nous avons partagé nos deux différentes bases de données comme suit

- Cas 1 : base de données du cabinet médical.
- Cas 2 : base de données du CHU.

Dans cette partie, la fonction *train_test_split()* prend en paramètres nos variables d'entrées (X) des deux cas, leurs variables de sortie (Y) et un pourcentage de découpage '*test_size*$= 0.2$'. Voir figure4.5

FIGURE 4.5: Répartition des données en un ensemble d'apprentissage et un ensemble test

Et elles retournent 8 variables ; quatre dans le cas 1 et quatre autres dans le cas 2 qui sont : X*_train*, y*_train* pour l'apprentissage et X*_test*, y*_test* pour le test final.

Validation (validation croisée)

Afin d'éviter le sur apprentissage, nous avons opté pour la validation croisée qui est similaire à *train_test_split()* , mais qui consiste à diviser nos données en k sous-ensembles différents. Nous utilisons $k - 1$ sous-ensembles pour entrainer nos données et laisser le dernier sous-ensemble comme données de test. Nous faisons ensuite la moyenne du modèle par rapport à chacun des sous-ensembles, puis finalisons notre modèle. Après cela, nous le testons par rapport à l'ensemble de test. La figure4.6 détaille ce schéma.

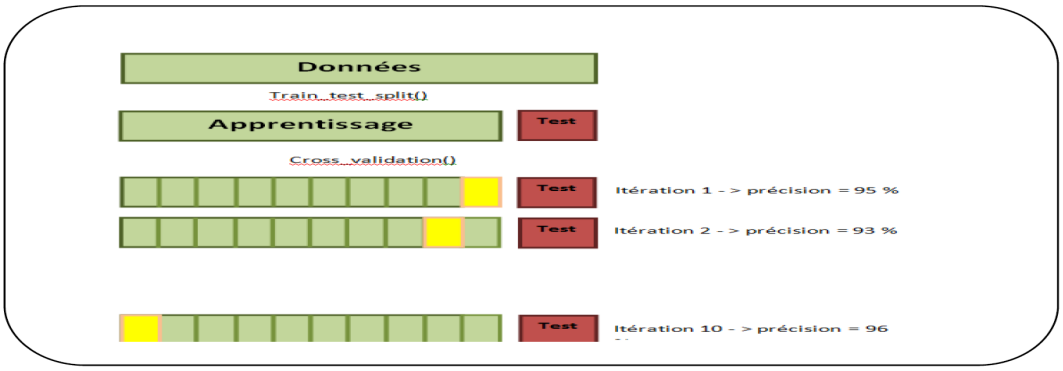

FIGURE 4.6: Schéma récapitulatif de la Validation croisée

Afin de déterminer la valeur améliorée des paramètres de l'algorithme d'apprentissage, nous avons mené plusieurs exécutions en faisant varier les valeurs de ces paramètres. Pour chaque expérience, la classification a été effectuée avec les mêmes ensembles obtenus en utilisant cinq méthodes différentes

1. Random Forest
2. K-plus proche voisin
3. Arbre de décision
4. SVM
5. Naives Bayes.

Random Forest (La forêt d'arbres aléatoires)

Les forêts d'arbres décisionnels[11] (ou forêts aléatoires de l'anglais *random forest classifier*) ont été premièrement proposées par Ho en 1995 [12] et ont été formellement proposées en 2001 par Leo Breiman[13] et Adele Cutler[14]. Elles font partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de bagging. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

Dans cette section, nous formerons notre classificateur de forêts d'arbres aléatoires en utilisant notre base de données *data_train*. Le résultat de précision de chaque itérations est donné dans la figure4.7 et a figure4.8

– **Cas** 1 **:**

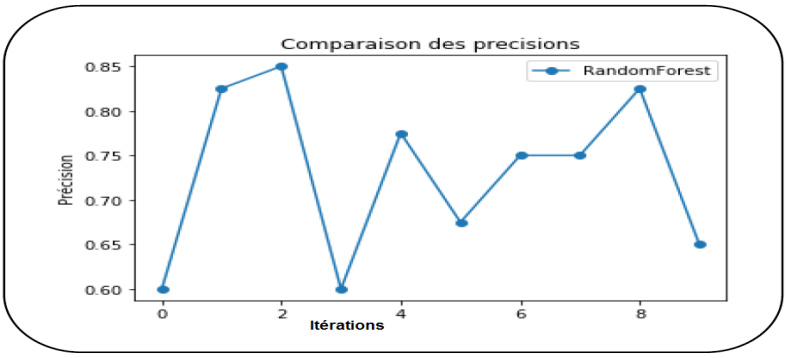

FIGURE 4.7: Représentation graphique de Random Forest : Cas1

Le graphe représente le taux de précision en fonction du nombre d'itérations. Nous remarquons que la meilleure itération est l'itération 2 avec un taux de précision de 85%.

– **Cas 2 :**

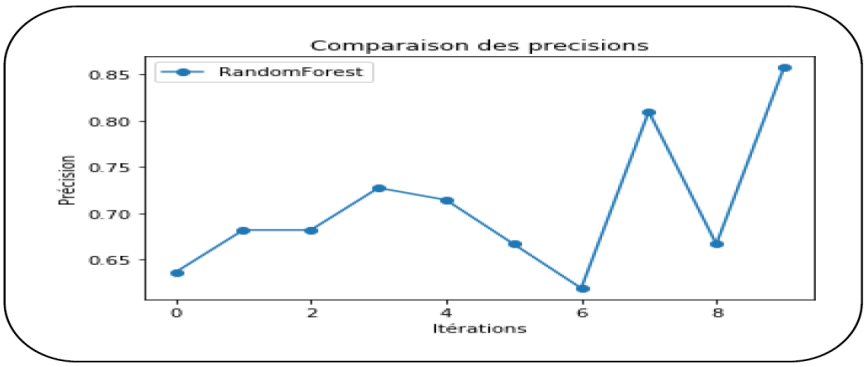

FIGURE 4.8: Représentation graphique de Random Forest : Cas2

Le graphe représente le taux de précision en fonction du nombre d'itérations.La meilleure itération est l'itération 9 avec un taux de précision de 85%.

Decision Tree Classifier (Arbre de décision)

L'apprentissage par arbre de décision désigne une méthode basée sur l'utilisation d'un arbre de décision comme modèle prédictif. On l'utilise notamment en fouille de données et en apprentissage automatique. Dans ces structures d'arbre, les feuilles représentent les valeurs de la variable-cible et les embranchements correspondent à des combinaisons de variables d'entrée qui mènent à ces valeurs. En analyse de décision, un arbre de décision peut être utilisé pour représenter de manière explicite les décisions réalisées et les processus qui les amènent. En apprentissage et en fouille de données, un arbre de décision décrit les données mais pas les décisions elles-mêmes, l'arbre serait utilisé comme point de départ au processus de décision.

C'est une technique d'apprentissage supervisé : on utilise un ensemble de données pour lesquelles on connaît la valeur de la variable-cible afin de construire l'arbre (données dites éti-

quetées), puis on extrapole les résultats à l'ensemble des données de test. Les arbres de décision font partie des algorithmes les plus populaires en apprentissage automatique [15, 16].

– **Cas** 1 **:**

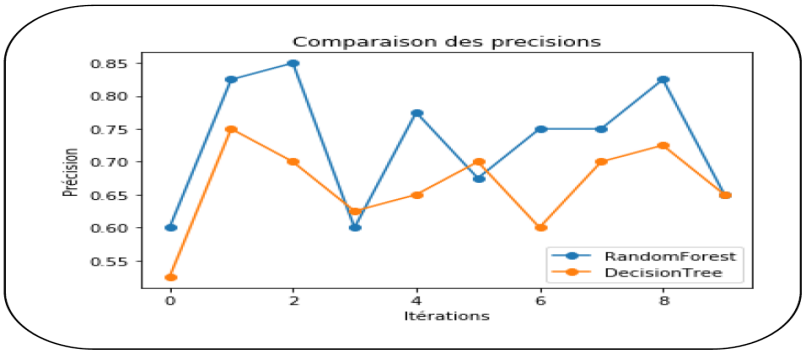

FIGURE 4.9: Comparaison graphique entre le l'algorithme Random Forest et Decision Tree Classifier : Cas1

Le graphe donnée dans la figure4.9 représente le taux de précision en fonction du nombre d'itérations pour les deux méthodes : Random Forest et Decision Tree Classifier. Nous remarquons qu'à nouveau la meilleure itération est celle de Random Forest avec un taux de précision de presque 85% comparé à celui de Decision Tree Classifier qui est de 75%.

– **Cas** 2 **:**

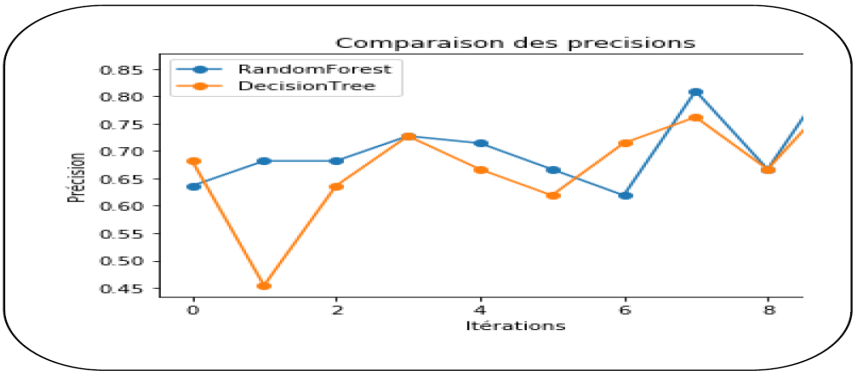

FIGURE 4.10: Comparaison graphique entre le l'algorithme Random Forest et Decision Tree Classifier : Cas2

Le graphe donnée dans la figure4.10 représente le taux de précision en fonction du nombre d'itérations pour les deux méthodes : Random Forest et Decision Tree Classifier. Nous remarquons qu'à nouveau la meilleure itération est celle de RandomForest avec un taux de précision de presque 85% comparé à celui de Decision Tree Classifier qui est de 80%.

K-Nearest Neighbors(*K*-plus proche voisin)

Dans ce cadre, on dispose d'une base de données d'apprentissage constituée de N couples "entrée-sortie". Pour estimer la sortie associée à une nouvelle entrée x, la méthode des k plus

proches voisins consiste à prendre en compte (de façon identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x, selon une distance à définir. Puisque cet algorithme est basé sur la distance, la normalisation peut améliorer sa précision [17, 18].

– **Cas** 1 **:**

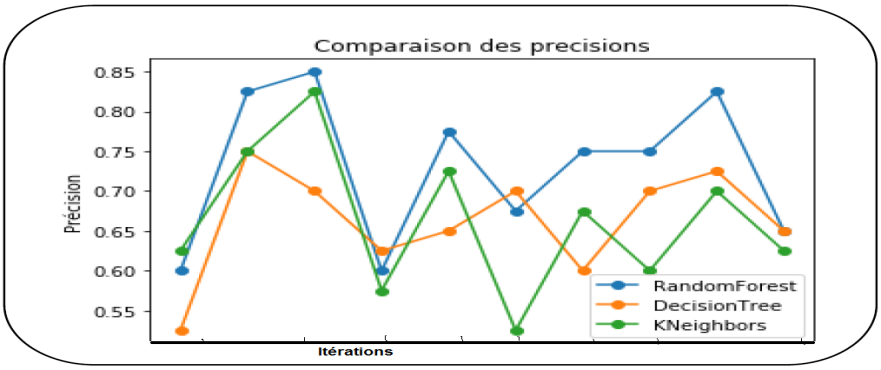

FIGURE 4.11: Représentation graphique des itérations en fonction du taux de précision : Cas1

Le graphe donnée dans la figure4.11 représente le taux de précision en fonction du nombre d'itérations pour les méthodes : Random Forest, Decision Tree Classifier et la méthode du K-plus proche voisin. L'algorithme Random Forest ayant encore le meilleur taux de précisons contrairement à l'algorithme K-plus proche voisin avec un taux de précision de 82.5%.

– **Cas** 2 **:**

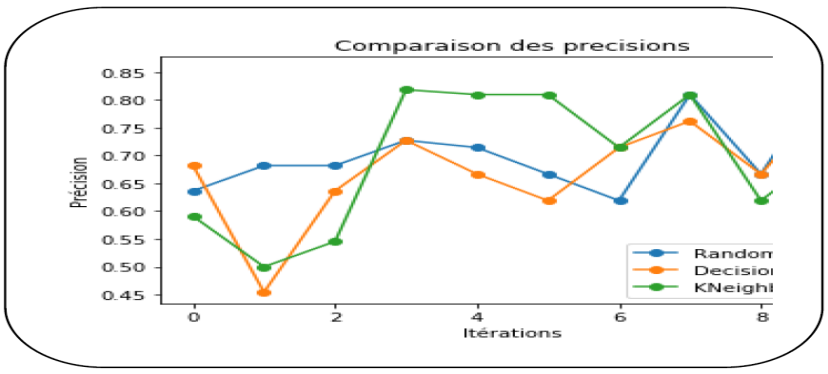

FIGURE 4.12: Représentation graphique des itérations en fonction du taux de précision : Cas2

Le graphe donnée dans la figure4.11 représente le taux de précision en fonction du nombre d'itérations pour les méthodes : Random Forest, Decision Tree Classifier et la méthode du K-plus proche voisin pour le cas 2. L'algorithme Random Forest ayant encore le meilleur taux de précisons précédé par l'algorithme K-plus proche voisin avec un taux de précision de 81% à l'itération 3.

Support Vector Machine (Machines à vecteurs de support SVM)

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de classification et de régression. Les SVM sont une généralisation des classifieurs linéaires. Les séparateurs à vastes marges reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. Ces deux notions existaient depuis plusieurs années avant qu'elles ne soient mises en commun pour construire les SVM [19].

– **Cas** 1 **:**

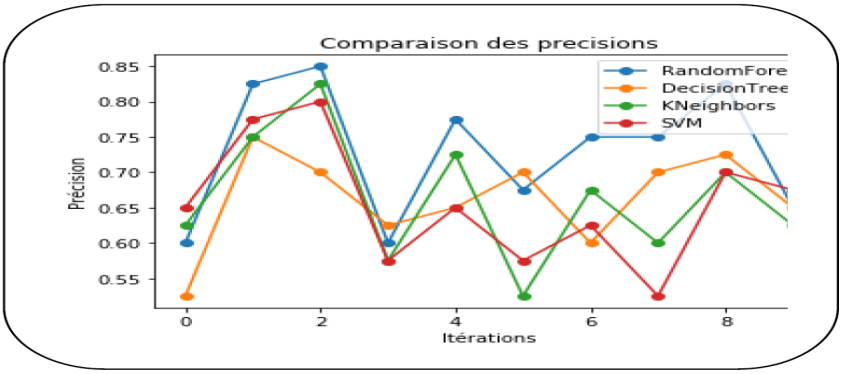

FIGURE 4.13: Représentation graphique des itérations en fonction du taux de précision : Cas1

Le graphe donnée dans la figure4.13 représente le taux de précision en fonction du nombre d'itérations pour les méthodes : Random Forest, Decision Tree Classifier, la méthode du K-plus proche voisin et SVM. La meilleure itération est toujours celle de l'algorithme RandomForest avec un taux de précision qui dépasse les 80% contrairement aux 3 autres qui ont tous le même taux de précisons le plus faible qui est égale à $52,5\%$.

– **Cas** 2 **:**

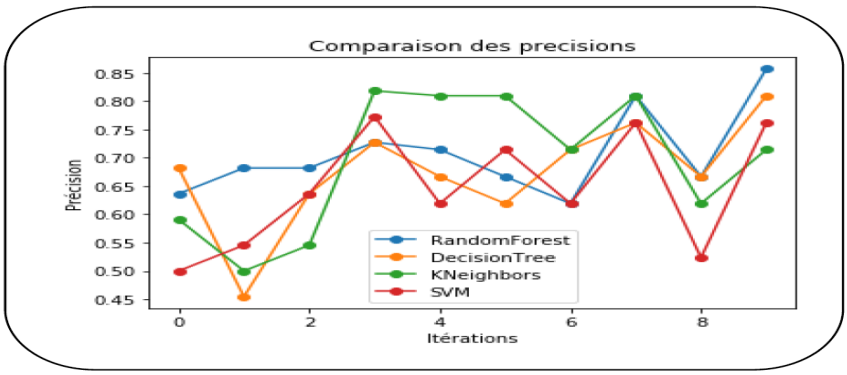

FIGURE 4.14: Représentation graphique des itérations en fonction du taux de précision : Cas2

Le graphe donnée dans la figure4.14 représente le taux de précision en fonction du nombre d'itérations pour les méthodes : Random Forest, Decision Tree Classifier, la méthode du

K-plus proche voisin pour le cas 2. L'algorithme Random Forest ayant le meilleur taux de précisons comparé à SVM qui a le taux de précision le plus faible à l'itération 1.

Naives Bayes(Classification naïve bayésienne GNV)

La classification naïve bayésienne est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésien naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs linéaires. Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classifieurs bayésiens naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue [20].

– **Cas** 1 **:**

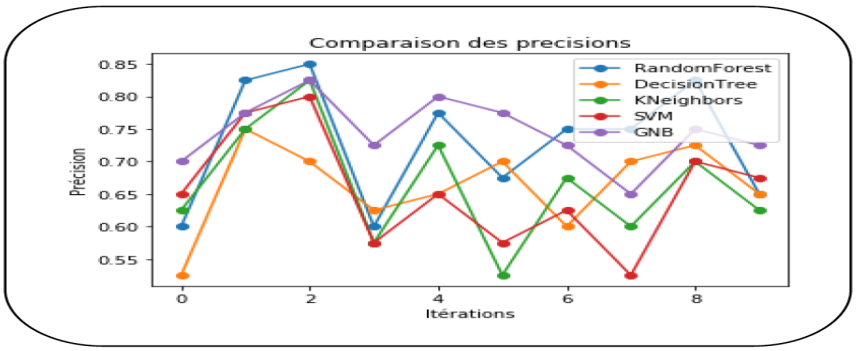

FIGURE 4.15: Représentation graphique des itérations en fonction du taux de précision : Cas1

Le graphe donnée dans la figure4.15 représente le taux de précision en fonction du nombre d'itérations entres les quatre précédants algorithmes et celui de Naives Bayes pour le cas 1. Comme les précédentes comparaisons celui de Random Forest garde le meilleur taux de précisions sois 85%.

– **Cas** 2 **:**

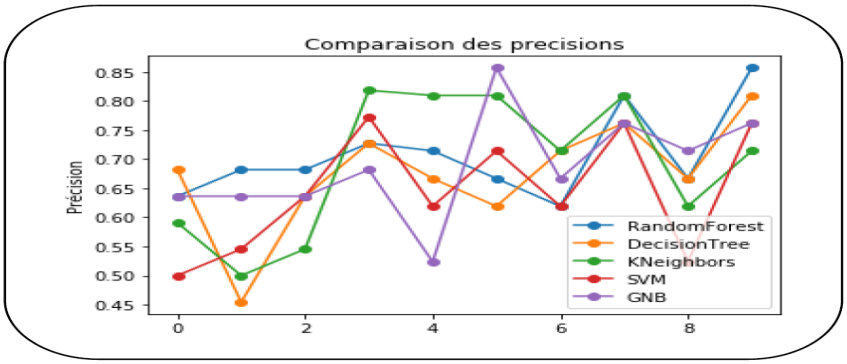

FIGURE 4.16: Représentation graphique des itérations en fonction du taux de précision : Cas2

Le graphe donnée dans la figure4.14 représente le taux de précision en fonction du nombre d'itérations pour les quatre précédants algorithmes et celui de Naives Bayes pour le cas 1. Contrairement aux autres traitements, l'algorithme Naives Bayes est le meilleur algorithme avec un taux de précision de 86% à l'itération 5.

4.4 Discussion

Parmi les algorithmes que nous avons testé dans les deux précédents cas (les deux ensembles de données), à savoir celui du cabinet médical et du CHU, nous avons constaté que les algorithmes Naïves Bayes et Random Forest ont le plus grand taux de précision avec 86% pour le premier et85% pour le deuxième.
Afin de choisir un des deux et obtenir des prédictions encore meilleures, trois techniques classiques sont envisageables :

1. Collecte de données : augmentez le nombre d'exemples d'apprentissage ;
2. Traitement des variables : ajoutez d'autres variables et un meilleur traitement des entités ;
3. Réglage des paramètres du modèle : envisagez d'autres valeurs pour les paramètres d'apprentissage utilisés par l'algorithme.

Nous avons opté pour le réglage des paramètres. Pour cela, nous ferons appel à de la bibliothèque *Scikit-learn* qui nous propose la fonction *RandomizedSearchCV* qui consiste à tester un ensemble de valeurs prédéfinies par l'utilisateur pour chaque paramètre, présenté sous forme de liste. La fonction retourne l'ensemble des paramètres du modèle avec les bonnes valeurs qui correspond au meilleur taux de précision. Ces valeurs ont été utilisées dans l'apprentissage de nos modeles RandomForest et Naives Bayes une deuxième fois, et nous avons constaté un gain de précision de 16% pour l'algorithme RandomForest soit 87%, contrairement à celui de Naives Bayes où il n'y a eu aucune amélioration.

```
Précision de random Forest est : 0.8660606060606
Le resultat precédent  est :     0.85060606060606
Le gain de precision est :      16.393939393939396 %
```

Ce modèle est sauvegardé sous format binaire avec la libraire Pickle, pour l'utiliser dans notre application web.

4.5 Application de prédiction web

Le résultat de notre travail, nous a permi de développer une application dans le but de permettre à différents individus de prédire si oui ou non ils peuvent être susceptibles de développer

un diabète de type2. La page d'accueil ne nécessite pas une authentification. Elle permet à l'utilisateur d'introduire ses données personnelles, à savoir : son genre, l'âge, IMC, antécédents familiaux, groupe sanguin ainsi que la sédentarité.(Voir la figure4.17)

FIGURE 4.17: Page d'accueil de l'application.

Pour l'implémentation du meilleur algorithme qui s'est avéré être celui de Random Forest, nous utiliserons dans notre interface principale un formulaire avec huit champs :

- Taille, Poids, Age de type float ;
- Pression artérielle et groupe sanguin de type sélection ;
- Genre, Sportif et sportif type booléen

Afin de traiter la requête : tous les champs doivent être remplis, sinon un message d'erreur s'affiche.(Voir la figure4.18)

FIGURE 4.18: Message d'erreur.

Une fois, que la requête est validé, les informations du formulaire sont récupérées. Notre application fait appel au modèle (Random Forest) : sauvegarder en format binaire et en lui passant les données saisis par l'utilisateur. Notre modèle va nous retourner un résultat 0 pour non-malade et 1 pour malade. En plus des informations supplémentaires, le temps d'exécution et la précision.(Voir la figure4.19)

FIGURE 4.19: Résultat final de l'application.

4.6 Conclusion

Dans ce travail, nous avons mené une étude critique des travaux proposés sur la prédiction du diabète type 2. Pour ce faire, nous avons commencé par la définition des critères d'évaluation des différentes solutions existantes. Ensuite, nous avons fait une comparaison des travaux passés en revue, dans laquelle nous avons repris l'essentiel des avantages et inconvénients des travaux proposés. Pour finir par proposer notre méthode qui est une amélioration des travaux étudiés. Nous avons présenté une méthode pour la prédiction du diabète basée sur l'application des algorithmes d'apprentissage automatique supervisé. Nous avons testé les algorithmes d'apprentissage supervisé sur les deux différentes bases de données, à savoir celle du CHU de Béjaia et du cabinet privé du Dr Djamel MEHIDI. Par la suite, nous avons amélioré le meilleur algorithme d'apprentissage, à savoir RamdanForest, en termes de taux de précision, temps d'exécution. Le taux de classification obtenu avec notre méthode est parmi les meilleurs résultats obtenus pour la classification du diabète de type 2, par rapport aux autres algorithmes des travaux de l'état de l'art, soit un taux de précision de 86%.

En termes de perspectives, la prédiction du diabète à l'aide des méthodes d'apprentissage peut être élargie en utilisant les méthodes de base de connaissance pour augmenter l'interopérabilité du diagnostic.

Références

1. D. Hellmann, "The Python Standard Library by Example", Addison-Wesley, 2017.
2. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss and V. Dubourg, "Scikit-learn : Machine learning in Python ", Journal of Machine Learning Research, vol. 12, pp. 2825 – 2830, ? 2011.
3. A. C. Müller and S. Guido, "Introduction to Machine Learning with Python", O'Reilly, 2016.
4. https ://www.spyder-ide.org/
5. https ://software.opensuse.org/package/python-seaborn ?locale=fr
6. http ://grasland.script.univ-paris-diderot.fr/STAT98/stat98_6/stat98_6.htm
7. I. D. Federation. Atlas du diabete de la fid neuvième édition 2019, 2019. URL https ://www.federationdesdiabetiques.org/.
8. L. N. De Castro and F.J. Von Zuben, "Learning and Optimization Using the Clonal Selection Principle," IEEE Transactions on Evolutionary Computation ; Special Issue on Artificial Immune Systems, 2001.

9. D. Goodman, L. Boggess, and A. Watkins, "An Investigation into the Source of Power for AIRS ; an Artificial Immune Classification System," Proceedings of the International Joint Conference on Neural Networks (IJCNN'03)., pp. 1678 - 1683. 2003.
10. J.M. Keller, M.R. Gray, and J.A. Givens, "A fuzzy k-nearest neighbor algorithm," IEEE Transactions on Systems, Man and Cybernetics, SMC-15, pp.580 – 585,1985.
11. R. Nisbet, J. Elder, and G. Miner, "Handbook for Statistical Analysis And Data Mining", Academic Press, Page 247, Edition 2009.
12. Ho, T. Kam, " Random Decision Forests ", Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, ? 14-16 august 1995, pp. 278-282, 1995.
13. L. Breiman, "Random Forests", Machine Learning, vol. 45, no 1, ? pp. 5–32, 2001.
14. A. Liaw, "Documentation for R package randomForest" [archive], 16 octobre 2012, 2012.
15. X. Wu, V. Kumar, J. R. Quinlan and J. Ghosh, "Top 10 algorithms in data mining", Knowledge and Information Systems, vol. 14, no 1, ? pp. 1–37, 2008.
16. S. M. Piryonesi and T. E. El-Diraby, "Data Analytics in Asset Management : Cost-Effective Prediction of the Pavement Condition Index", Journal of Infrastructure Systems, vol. 26, no 1, pp. 04019036, 2020.
17. S. M Piryonesi and T. E. El-Diraby, "Role of Data Analytics in Infrastructure Asset Management : Overcoming Data Size and Quality Problems", Journal of Transportation Engineering, Part B : Pavements, vol.146, no 2, 2020.
18. T. Hastie, R. Tibshirani and J. Friedman, "The elements of statistical learning : data mining, inference, and prediction" , Springer, 2001.
19. B. Boser, I. Guyon, and V. Vapnik, "Pattern recognition system using support vectors". US Patent 5,649,068, 1997.
20. D. Pedro and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss", Machine Learning, vol. 29, pp. 103–137, 1997.