MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABBES LAGHROUR KHENCHELA
FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département de Mathématique et Informatique

N° de série :……

Mémoire de fin d'études

Pour l'obtention du diplôme de Master (L.M.D)

Spécialité : Informatique
Option : sécurité et technologie web

THEME

Sélection d'attributs et classification pour le diagnostique des maladies chroniques

Réalisé par :
- KHANOUF Romaissa
- BELOUAAR Maissa

Dirigé par :
Dr.HOUASSI Hichem

Membres de jury :

M.	Président
M.	Examinateur
M.	Examinateur

Présenté le /06/2022

بسم الله الرحمن الرحيم

Résumé

Dans ce mémoire, nous avons conçu les modèles pour les maladies chroniques afin de réduire le risque et la survenue de complications de ces maladies sur la santé du patient. Pour concevoir ces modèles, nous avons utilisé deux approches, nous présentons une méthode adaptatifs appelés SelectKBest qui est de la méthode de sélection d'attribut. Il serait combiné avec différents classificateurs pour trouver les meilleurs attributs, qui atteignent la plus grande précision dans la classification. Enfin, nous présentons les algorithmes K voisins les plus proches (KNN), Arbres de décision, Machine à vecteurs de support (SVM) et Naive Bayes pour découvrir les règles de classification, Nos approches proposées ont été expérimentées sur des ensembles de données de référence (Wisconsin Breast Cancer, Diabète Pima Indian, CAD). Les résultats expérimentaux prouvent que les précisions de classification sont puissantes pour les différents ensembles de données.

Mots clés : Maladie chronique, sélection d'attribut, classification, K voisins les plus proches, Arbres de décision, Les séparateurs à Vaste Marge SVM .

Abstract

In this dissertation, we designed the models for chronic diseases to reduce the risk and occurrence of complications of these diseases on the health of the patient. To design these models, we used two approaches, we present an adaptive method called SelectKBest which is the attribute selection method. It would be combined with different classifiers to find the best attributes, which achieve the highest accuracy in classification. Finally, we present the K-nearest neighbor (KNN), Decision trees, Support vector machine (SVM) and Naive Bayes algorithms to discover the classification rules, Our proposed approaches have been experimented on benchmark datasets (Wisconsin Breast Cancer, Diabetes Pima Indian, CAD). The experimental results prove that the classification accuracies are powerful for the different datasets.

Keywords: chronic disease, attribute selection, classification, K nearest neighbors, Decision Trees, Wide Margin Separators SVM.

ملخص

في هذه الأطروحة، قمنا بتصميم نماذج للأمراض المزمنة لتقليل خطر حدوث مضاعفات هذه الأمراض على صحة المريض. لتصميم هذه النماذج ، استخدمنا نهجين ، نقدم طريقة تكيفية تسمى SelectKBest وهي طريقة اختيار السمة. سيتم دمجها مع مصنفات مختلفة للعثور على أفضل السمات ، والتي تحقق أعلى دقة في التصنيف. أخيرا ، نقدم أقرب جار K، وأشجار القرار ، وآلة ناقلات الدعم (SVM) وخوارزميات Naive Bayes لاكتشاف قواعد التصنيف ، وقد تم تجربة نهجنا المقترحة على مجموعات البيانات القياسية (سرطان الثدي في ويسكونسن ، والسكري بيما الهندي ، و CAD). تثبت النتائج التجريبية أن دقة التصنيف قوية لمجموعات البيانات المختلفة.

الكلمات المفتاحية: الأمراض المزمنة، اختيار السمات، التصنيف، أقرب الجيران K، أشجار القرار، فواصل الهامش الواسع SVM.

Remerciements

Ce travail est le fruit de la combinaison d'efforts de deux étudiantes.

Je remercie Tout d'abord le Tout puissant par sa grâce m'a permis

d'arriver à la boutade mes efforts en me donnant la santé, la force,

le courage et en me faisant entourer des merveilleuses personne dont je tiens à remercier :

Mon Directeur de mémoire, Dr Houassi Hichem pour son encadrement sans faille, Son soutien moral, Sa rigueur au travail, ses multiples conseils,

ses orientations et sa disponibilité malgré ses multiples occupations.

Tous les enseignants de spécialité informatique, pour leurs enseignements

de qualité et leurs conseils qui nous ont permis de poursuivre notre itinéraire académique jusqu'à présent.

Je remercie chaleureusement mes amis et mes collègues qui m'ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Mes dernières pensées vont vers mes familles, et surtout mes parents, qui

M'ont encouragé à poursuivre mes études jusqu'à aujourd'hui.

Dédicace

Je dédie ce mémoire

A ma mère CHERIFA et mon père MAHBOUBI pour leur amour inestimable, leurs sacrifices, leur confiance, leur soutien et toutes les valeurs qu'ils ont su m'inculquer ;

A ma grand-mère AIDA pour toute l'affection qu'elle m'a donnée et pour son précieux encouragement ;

A mon frère Adel et son fils Iskander et Ahmed Samer , et mes sœurs Ahlem et son mari et ses filles (Ghofrane , Rihane et Nihane) , Sabrine , houda, Amel et son mari , Sirine ,et sa mère Salima pour leur tendresse, leur complicité et leur présence ;

A mes tantes : Dalila , Soumia Moufida et Makkia et Mlouka leur conjoints et leur enfants et touts la familles belouaar, pour leurs mots d'encouragement et leur gentillesse ;

A mes amis ; Samra , Romaissa, Hakim ;

A mon chat basous qui est resté avec moi pour terminer mon mémoire ;

A tous mes camarades de l'Université abbés laghrour en général et promo de deuxième année master Informatique 2022 Et Groupe De STW Spécialement.

BELOUAAR MAISSA

Dédicace

Je dédié ce mémoire

A mes chers parents ma mère et mon père

Pour leur patience, leur amour, leur soutient et leurs encouragement

A mon seule frère

A mes sœurs

A mes neveux et les maries de mes sœurs

A mes oncles

A tous ceux que j'aime

Sans oublier tous les professeurs que ce soit du primaire, du moyen, du secondaire ou de l'enseignement supérieur.

Qu'ils trouvent ici l'expression de toute ma reconnaissance.

KHANOUF ROMAISSA

Tables des Matières

Introduction générale 13

Chapitre I : fouille de données

1.1. Introduction 16

1.2. Processus d'extraction de connaissances à partir des données 16

1.3. Datamining (fouille de donnée) 17

1.3.1. Définition de datamining 17

1.4. Les techniques de datamining 19

1.4.1. Nettoyage et préparation des données 19

1.4.2. Classification et prédiction 19

1.4.3. Association 20

1.4.4. Regroupement (clustering) 20

1.4.5. Régression 20

1.4.6. Visualisation 21

1.5. Les Méthode de classification 21

1.5.1. Les arbres de décision 21

1.5.2. Réseaux de neurones 22

1.5.3. SVM 23

1.5.4. Les K plus proche voisins (k-ppv) 24

1.5.5. Naïve Bayes 25

1.6. Conclusions 25

Chapitre II : Sélections d'attributs pour la classification

2.1. Introduction 27

2.2. Formulation du problème de classification 27

2.3. Classification de données de grande dimension 28

2.4. Réduction de la dimensionnalité 28

2.4.1. Sélection des attributs (fonctionnalités) 28

2.4.2. Extraction des attributs 29

2.5. Sélection des attributs 29
2.5.1. Définitions 29
2.5.2. Pourquoi utiliser la sélection des fonctionnalités 30
2.5.3. Processus de sélection d'attributs 31
2.5.4. Les procédures de recherches 32
2.5.5. Les procédures d'évaluations 32
2.5.6. Les critères d'arrêt 34
2.5.7. La validation 34
2.6. Catégorisation des méthodes de sélection d'attributs 34
2.6.1. Catégorisation basée sur les critères d'évaluation 34
2.6.2. Catégorisation basée sur la stratégie de recherche 35
2.7. Conclusion 36
Chapitre III : Approche de sélection d'attribut et implémentation
3.1. Introduction 39
3.2. Méthodologie de sélection d'attributs 39
3.3. Outils utilisés 40
3.4. Prétraitement des données 40
3.5. Approche de Sélection d'attribut 43
3.6. Approche de classification 43
3.6.1 Diviser les données d'entraînement et de test 44
3.6.2 Appelez les classificateurs 45
3.7 Expérimentation et résultats 46
3.7.1 Les bases de données 46
3.7.2 Critère d'évaluation utilisée 48
3.7.3. Résultats et discussion 49
3.7.3.1. Sans sélection d'attribut 50
3.7.3.2. Avec sélection d'attributs 51

3.8. Conclusion 53

Conclusion générale 55

Bibliographique 57

Liste des figures

Figure.1.1 : Schéma d'un processus de fouille de donnée 18

Figure.1.2 : Image montrant la Définition data mining 19

Figure 1.3. : Exemple d'un arbre de décision 21

Figure 1.4 : Exemple de réseau de neurones 22

Figure1.5 : Représentation des 7 données dans l'espace 23

Figure 1.6 : Deux droites séparatrices possibles pour notre exemple 24

Figure1.7: illustration des k-ppv d'un point 25

Figure2.1 : Définition Sélections d'attributs 30

Figure.2.2 : Processus de sélection d'attributs 31

Figure 2.3 : Sélection d'attributs à base de filtre 33

Figure 2.4 : Sélection d'attributs à base Wrapper 33

Figure 2.5 : Sélection d'attributs à base Embedded 34

Figure 2.6 : Catégorisation des méthodes de sélection de caractéristique 35

Figure 3.1 **:** Schéma d'approche de sélection d'attribut et de classification 39

Liste des Tableaux

Tableau 3.1 : Les caractéristiques de l'ensemble de données utilisées 47

Tableau 3.2 : Les résultats obtenus pour différents Data sets sans sélection d'attributs 50

Tableau 3.3 : les résultats obtenus avec une approche de sélection d'attributs 51

Tableau 3.4 : les meilleurs attributs sélectionné 52

Introduction Générale

La propagation des maladies chroniques ne fait qu'augmenter les problèmes économiques et sociaux, et la question des maladies chroniques est considérée comme cruciale car le nombre de personnes infectées augmente rapidement. Cependant, il est difficile de le comprendre car il est invisible et ses multiples traductions dans le travail quotidien, C'est pourquoi des programmes informatiques sont intervenus dans le domaine médical pour aider les médecins et alléger un peu la pression sur eux.

La fouille de données peut avoir plusieurs tendances selon les types de données traitées, comme la fouille de texte, la fouille de Web et la fouille de données médicales. Cependant, les utilisateurs des systèmes de fouille de données installés sont intéressés également par leurs technologies associées, et le seront d'autant plus que la plupart de ces installations devront être mises à jour à l'avenir au fur et à mesure avec les évolutions qui peuvent s'apparaitre au niveau des systèmes étudiés, ou bien dans des bases ou des entrepôts de données qui se modifient constamment avec le temps. La fouille de données est une discipline émergente combinant les mathématiques/statistiques et l'informatique et faisant usage des algorithmes d'apprentissage automatique.

Vue l'explosion de la quantité de données due des différents sources et suite à la complexité, la sensibilité à traiter, manipuler, stocker et exploiter efficacement ces données, il est devenu indispensable de trouver de nouvelles méthodes puissantes permetent d'analyser et de classer cette volumineuse masse d'informations avec une meilleure façon. Parmi ces technique on trouve la sélection de données qui permettent une réduction importante des volumes de données par la sélection les données les plus utiles afin de faciliter les tâches de la fouille de données.

Dans ce mémoire sous le titre sélection d'attributs et classifications pour le diagnostique des maladies chronique, nou nous intéressont au problème de sélection des attributs au niveau bases de données médicales (concernant les maladies chroniques comme le diabète) utilisé pour l'apprentissage des modèles de classififcation.

Ce mémoire est organisé en deux parties : la partie théorique et la partie pratique.

La partie théorique comprende deux chapitres:

- Chapitre I : fouille de données
- Chapitre II : Sélection d'attributs pour la classification

La partie pratique comprende un chapitre qui présente notre travail:

- Chapitre III : Approche de sélection d'attribut et implémentation

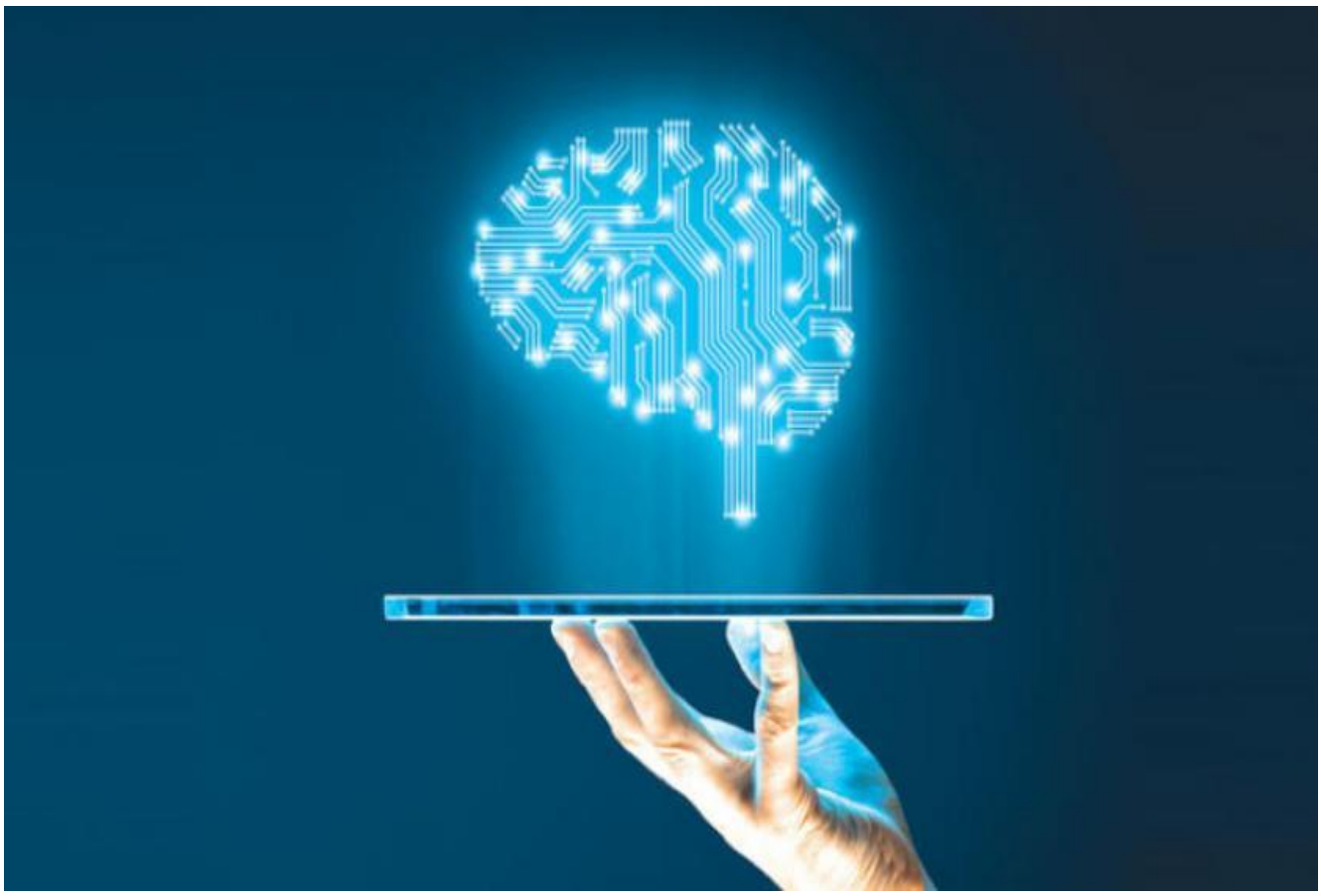
Dans le premier chapitre, nous avons abordé le processus d'extraction de connaissances à partir de données, le datamining, les différent technique de datamining et en fin les méthodes de classification.

Dans le deuxième chapitre, nous avons présenter la réduction de la dimensionnalité qui est divisé en deux vollets : sélection des attributs et extraction des attributs mais dans ce mémoire nous intéressons à la sélection des attributs. Nous avons expliquer d'une part la motivation du d'utilisation de la sélection des attributs et d'une autre part les processus de sélection d'attributs et les méthodes de sélection d'attributs.

Enfin, dans le troisième chapitre, Nous avons présenter les résultats trouvées à partir de l'application d'une méthode de sélection d'attributs sur des bases de données des différentes maladies chroniques au biais d'utilisation des différents algorithmes de classification comme SVM,KNN,Naive-bayes,DecisionTree).

Chapitre I :

Fouille de donné



1.1 Introduction

Depuis quelques années, une masse grandissante de données est générée dans différents domaines tel que les réseaux sociaux, la médecine, l'industrie, etc. Les techniques usuelles analysant ces données sont insuffisantes d'où le besoin d'une nouvelle génération d'outils et de théories pour aider à extraire les informations utiles (les connaissances) à partir de ces volumes de données qui croissent rapidement. Ces théorise et outils sont le sujet d'un nouveau domaine appelé extraction de connaissances à partir de données ECD ou (KDD : Knowledge Discoveryfrom Data en anglais) dont le cœur est la fouille de données. La fouille de données est un domaine qui est apparu avec l'explosion des quantités d'informations stockées, avec le progrès important des vitesses de traitement et des supports de stockage.

1.1. Processus d'extraction de connaissance à partir de données

La découverte de connaissances à partir de données KDD (Knowledge Discovery in Data bases) est une modélisation et analyse exploratoire automatique d'un ensemble de données de grande taille. Le KDD est un processus organisé pour l'identification de configurations valides, nouvelles, et compréhensibles à partir d'un ensemble de données larges et complexe. La fouille de données est le cœur du processus du KDD. Elle utilise des algorithmes qui explorent les données, développent un modèle et découvrent des configurations inconnues. Les modèles de fouille de données sont utilisés pour comprendre les phénomènes de l'analyse et de la prévision des données.

Le processus de découverte de connaissances est itératif et interactif, il suit plusieurs étapes qui sont illustrées par le schéma de la figure 1.1, ce processus commence par la détermination des buts du KDD et se termine par l'implémentation des connaissances découvertes.[1]

a) Compréhension du domaine d'application et détermination des butes

C'est une phase préparatoire dans laquelle les buts de l'utilisateur final et l'environnement de développement dans lequel se déroule le processus du KDD doivent être cernées, puis l'ensemble d'opérations de prétraitements sera lancé comme sera décrit dans les prochaines étapes.

b) Sélection des données

La sélection des données sur lesquels sera effectuée la découverte est primordiale. Cela inclue, la découverte des données valables, l'obtention des données nécessaires puis leur intégration dans un seul ensemble de données .Ce processus est très important car la fouille de données apprend et découvre à partir de données valables. Si quelques attributs importants manquent alors l'étude entière peut échouer.

c) Prétraitement des données

Dans cette étape, la fiabilité des données est améliorée. Les données bruitées sont supprimées tel que les données doubles qui peuvent se révéler gênantes parce qu'elles vont donner plus d'importance aux valeurs répétées. Un contrôle sur les domaines des valeurs permet de retrouver des valeurs aberrantes.

d) Transformation des données

Dans cette étape les méthodes incluent la réduction de dimension (telle que la sélection et l'extraction des caractéristiques) et la transformation des attributs (la discrétisation des attributs numériques et la transformation fonctionnelle). Cette étape est souvent cruciale pour le succès du projet de KDD mais elle est usuellement spécifique au projet.

e) Fouille de données

Cette étape consiste a faire les point suivant

- **Choix d'une tache de fouille de données appropriée :** Dans cette étape le choix d'une tâche de fouille de données est possible, tel que la classification, la régression, le clustering etc. Ceci dépend généralement des buts du KDD et également des étapes précédentes.
- **Choix de l'algorithme de fouille de données :** Cette étape inclue la sélection des méthodes spécifiques qui seront utilisées pour la recherche des connaissances.
- **Application de l'algorithme de fouille de données :** Finalement l'implémentation de l'algorithme de fouille de données est réalisée. Dans cette étape l'algorithme peut être utilisé plusieurs fois jusqu''a ce qu'un résultat satisfaisant soit obtenu.

f) Evaluation des modèles

Dans cette étape l'évaluation et l'interprétation des configurations extraites avec le respect des buts définis dans la première étape auront lieu. Ici les étapes de prétraitements sont considérées ainsi que leurs effets sur les résultats des algorithmes de fouilles de données.

Cette étape se concentre sur la compréhensibilité et l'utilité du modèle induit. Ici la connaissance découverte est également documentée pour des utilisations futures.

g) Utilisation des connaissances découvertes

Dans cette étape la connaissance est prête à être incorporée dans un autre système pour d'autres actions. En fait le succès de cette étape détermine l'efficacité é du processus entier de KDD.

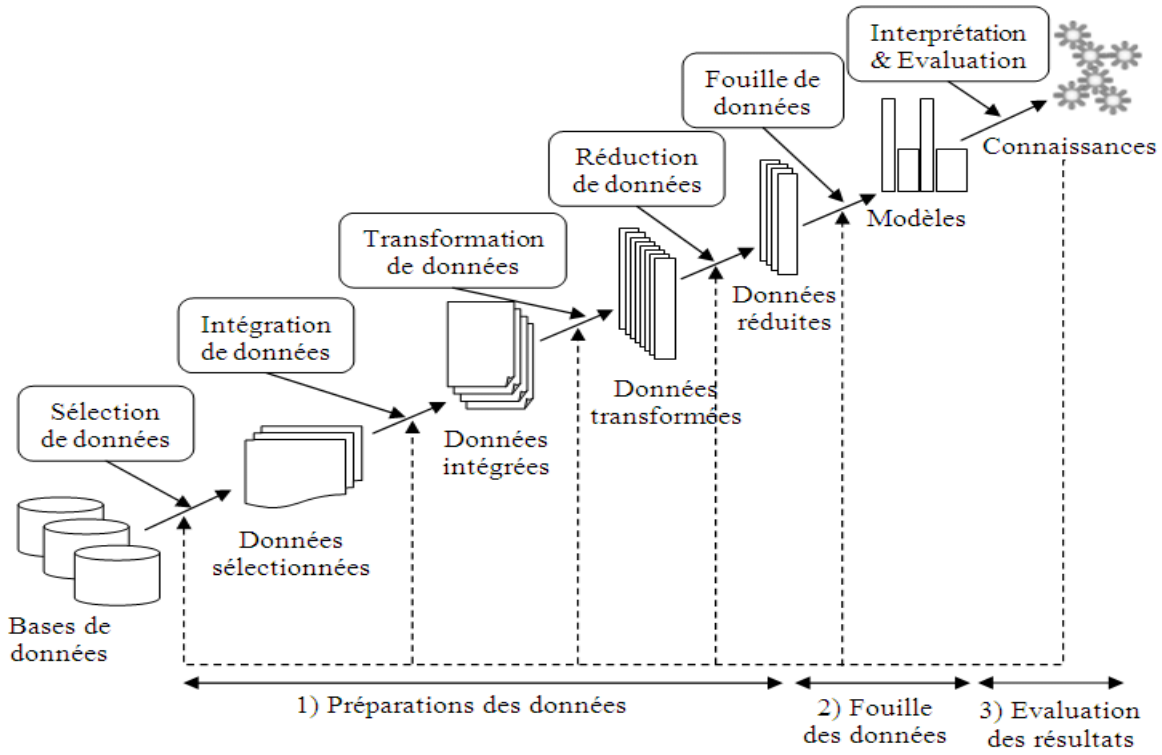


Figure.1.1 : Schéma d'un processus de fouille de donnée.[2]

3.1 Fouille de donnée (Data mining)

3.1.1. Définition du data mining

Le «data mining» que l'on peut traduire par «fouille de données» apparaît au milieu des années 1990 aux États-Unis comme une nouvelle discipline à l'interface de la statistique et des technologies de l'information : bases de données, intelligence artificielle et apprentissage automatique (« machine learning »).

Le data mining est l'exploration et l'analyse de données volumineuses pour découvrir des modèles et des règles significatives. Le Data mining est en fait un terme générique englobant toute une famille d'outils facilitant l'explorationet l'analyse. On confondra ici le «

data mining », au sens étroit qui désigne la phase d'extraction des connaissances, avec la découverte de connaissances dans les bases de données (KDD ou Knowledge Discovery in Databases).

David Hand (1998) en donne la definition suivante: « *Data mining consists in the discovery of interesting, unexpected, or valuable structures in large data sets*» [3].

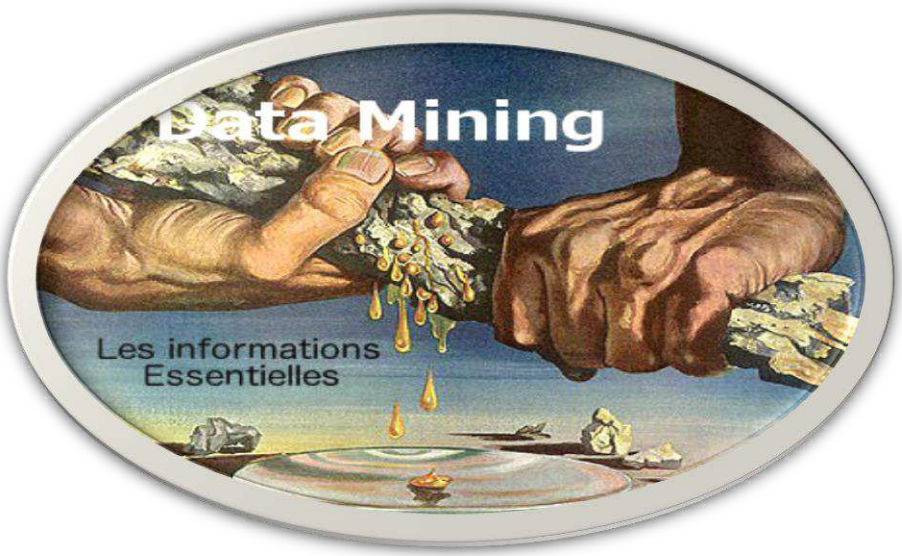


Figure.1.2 Image montrant la Définition data mining [4]

1.4. Les techniques de data mining

L'exploration de données contient une gamme de techniques [5], dont certaines nous sont utiles dans notre domaine d'étude :

1.4.1. Nettoyage et préparation des données

La Nettoyage et la préparation des données sont des étapes essentielles du processus d'exploration des données. Les données brutes doivent être nettoyées et formatées pour être utilisées dans différentes méthodes d'analyse. Le nettoyage et la préparation des données comprennent différents étapes de modélisation, de transformation, de migration des données et d'agrégation des données. Sans cette étape, les données n'ont aucun sens ou ne sont fiables pour l'entreprises en raison de la qualité des donnée, ces étapes sont également nécessaires pour la qualité des donnée.

1.4.2. Classification et prédiction

Les techniques d'exploration de données à des fins de classification impliquent l'analyse de divers attributs associés à différents types de données. Une fois que les entreprises ont

identifié les principales caractéristiques de ces types de données peuvent être les catégoriser en conséquence. Cette étape est essentielle pour identifier, par exemple, les informations personnelles qu'une entreprise peut souhaiter protéger ou supprimer des documents.

La prédiction est un aspect très puissant de l'exploration de données et représente l'une des principales branches de l'analyse. L'analyse prédictive utilise des modèles établis à partir de données actuelles ou historiques pour l'étendre dans le futur. Il fournit aux entreprises une analyse des tendances, qui sera ensuite reflétée dans leurs données. Il existe différentes façons d'utiliser l'analyse prédictive. Parmi les plus avancées figurent l'apprentissage automatique et l'intelligence artificielle.

1.4.3. Association

L'association est une technique d'exploration de données liée aux statistiques. Il est utilisé pour indiquer quand certaines données (ou événements identifiés dans les données) sont liées à d'autres données ou événements pilotés par les données. Il est similaire au concept de cooccurrence dans l'apprentissage automatique, où la probabilité d'un événement basé sur les données est représentée par l'existence d'un autre événement basé sur les données.

1.4.4. Regroupement (Clustering)

Le clustering est une technique analytique qui s'appuie sur des méthodes de visualisation pour bien comprendre les données. Les mécanismes de clustering utilisent des graphiques pour montrer où la distribution des données est liée à différents types de métriques. Ils utilisent également différentes couleurs pour illustrer la distribution des données.

Les méthodes graphiques sont idéales pour l'analyse de groupe. Avec les graphiques et les clusters en particulier, les utilisateurs peuvent identifier les tendances liées à leurs objectifs commerciaux grâce à des informations visuelles.

1.4.5. Régression

Les techniques de régression peuvent être utilisées pour identifier la nature des relations entre les variables d'un ensemble de données. Ces relations peuvent être causales dans certains cas, ou simplement corrélées dans d'autres. La régression est une simple technique de test en boîte blanche qui peut clairement révéler la relation entre les variables. Les techniques de régression sont utilisées dans certains aspects de la prévision et de la modélisation des données.

1.4.6. Visualisation

La visualisation des données est une autre partie importante de l'exploration des données. Ils donnent aux utilisateurs un aperçu des données basées sur la perception sensorielle. Les visualisations de données d'aujourd'hui sont dynamiques, utiles pour les flux de données en temps réel et ont des couleurs différentes pour révéler différentes tendances et modèles de données.

Les tableaux de bord sont un moyen puissant d'utiliser la visualisation des données pour découvrir des informations sur l'exploration de données. Les entreprises peuvent baser leurs tableaux de bord sur différentes métriques et utiliser des visualisations pour mettre en évidence des modèles dans leurs données, plutôt que de simplement utiliser la sortie numérique de modèles statistiques.

1.5. Les Méthodes de classification

Nous disposons d'un ensemble de méthodes de classification, nous aborderons certaines d'entre elles qui nous intéressent dans cette étude, qui sont :

1.5.1. Les arbres de décision

Un arbre de décision est une représentation graphique d'une procédure de classification. Les nœuds internes de l'arbre sont des tests sur les champs ou attributs, les feuilles sont les classes. Lorsque les tests sont binaires, le fils gauche correspond à une réponse positive au test et le fils droit à une réponse négative. Pour classer un enregistrement, il suffit de descendre dans l'arbre selon les réponses aux différents tests pour l'enregistrement considéré. [6]

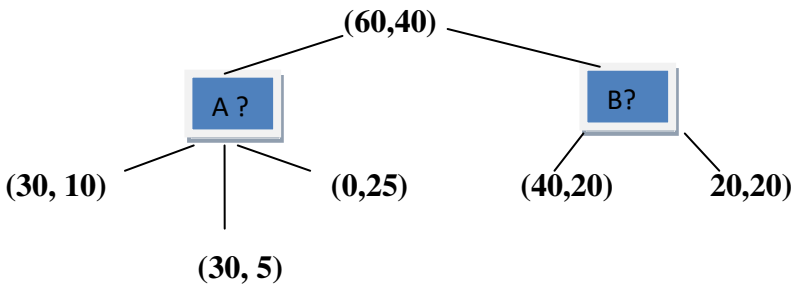


Figure 1.3. Exemple d'un arbre de décision

Apprentissage des arbres de décision

Avec en entrée, un échantillon de m enregistrements classés (x, c(X)), un algorithme d'apprentissage doit fournir en sortie un arbre de décision. La plupart des algorithmes

procèdent de façon descendante, c'est-`a-dire qu'ils choisissent la racine de l'arbre (en général un test) puis, récursivement, choisissent l'étiquette des fils. Pour simplifier la présentation, nous nous limitons, dans cette section à des problèmes où les attributs sont discrets et le nombre de classes égal à 2.

1.5.2. Les réseaux de neurone

Les Réseaux de Neurones Artificiels sont des modèles statistiques adaptatifs, basés sur une analogie avec le cerveau humain. Leur adaptabilité revient du fait qu'ils peuvent apprendre à estimer les paramètres d'un ensemble de données en utilisant un nombre d'exemples. Dans sa structure de base, un réseau de neurones artificiels est constitué d'un ensemble d'unités simples qui sont les neurones. Ces dernières sont interconnectées à l'aide d'un ensemble de connexions pondérées, chaque unité correspond à une caractéristique d'une forme à analyser.

Les Réseaux de Neurones Artificiels sont généralement organisés en couches. La première couche est la couche d'entrée. Elle est responsable de la transmission de l'information à analyser vers le réseau. La dernière couche est la couche de sortie. Elle reçoit le résultat final qui est la réponse du réseau. Les couches intermédiaires sont les couches cachées, le nombre de ces couches est défini selon le type du problème à résoudre. Elles relient les deux couches d'entrée et de sortie [7].

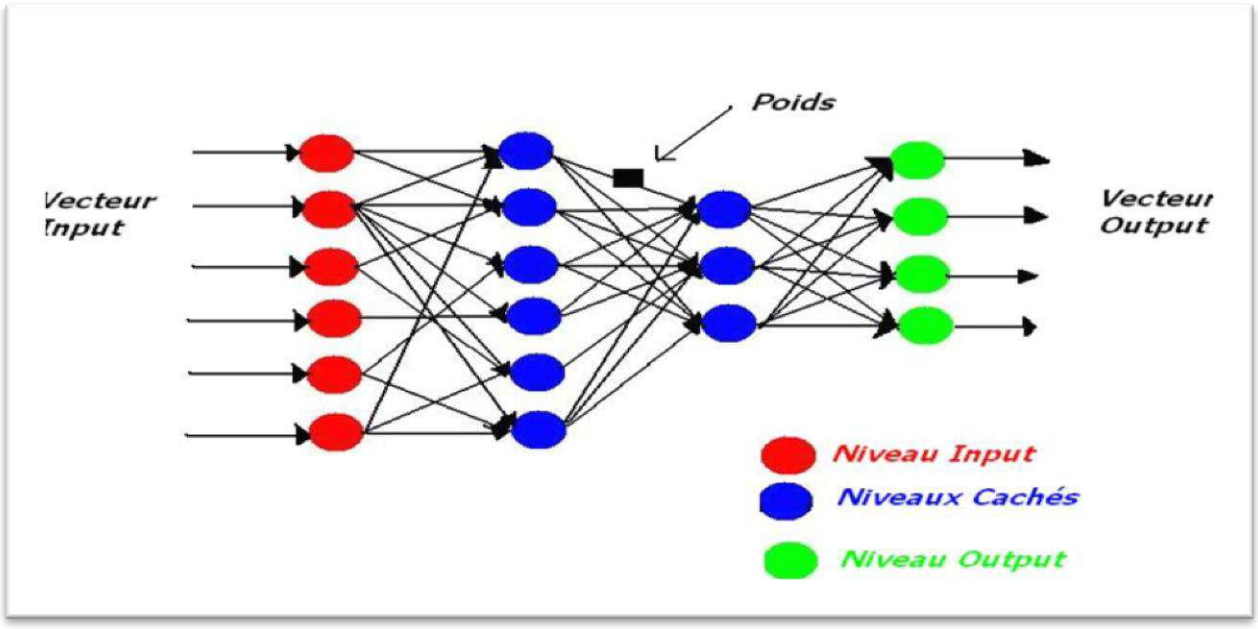


Figure 1.4 : Exemple de réseau de neurones [8]

Algorithme de réseaux neurones

La large majorité des réseaux de neurones possède un algorithme « d'entraînement » qui consiste à modifier les poids synaptiques en fonction d'un jeu de données présentées en entrée du réseau. Le but de cet entraînement est de permettre au réseau de neurones d'« apprendre » à partir des exemples. Si l'entraînement est correctement réalisé, le réseau est capable de fournir des réponses en sortie très proches des valeurs d'origine du jeu de données d'entraînement. Mais tout l'intérêt des réseaux de neurones réside dans leur capacité à généraliser à partir du jeu de test. Il est donc possible d'utiliser un réseau de neurones pour réaliser une mémoire ; on parle alors de mémoire neuronale.

1.5.3. SVM

Prise en charge des machines vectorielles ("SVM" ou "Separator Grandes marges" dépend de la traduction !) est une méthode très puissante. Dérivé d'une analyse mathématique précise et avancée des problèmes d'apprentissage séparateur binaire dans l'espace vectoriel.

Un SVM est un séparateur binaire, c'est-`a-dire qu'il vise à séparer les données étiquetées en deux sous-ensembles disjoints. Pur un problème a deux classes, les données d'apprentissage sont décrites par des points dans un espace vectoriel : les SVM s'appliquent donc essentiellement à des données numériques, comme celles de la Figure 1.5 Dans cet espace, les séparateurs les plus simples sont des hyperplans. [9]

Jusqu'à récemment, ces méthodes donnaient généralement meilleurs résultats.

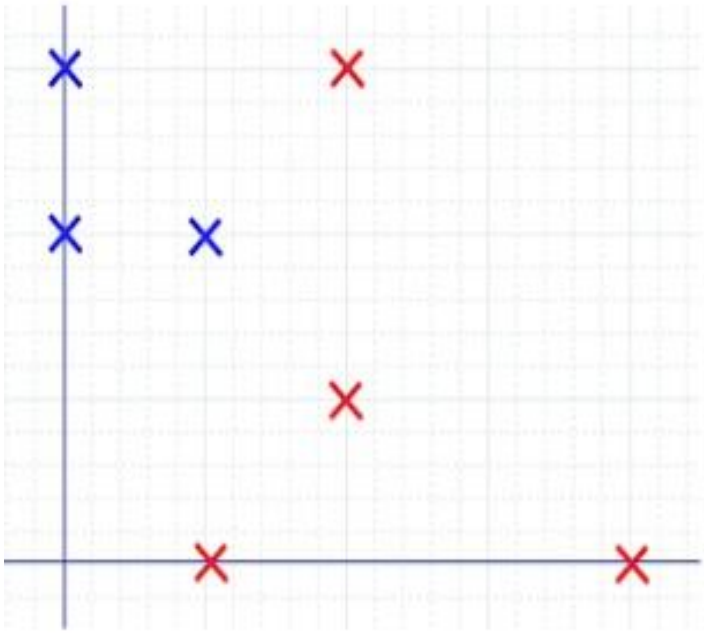


FIG.*1.5 : Représentation des 7 données dans l'espace* **[9]**

Suivant les données d'apprentissage, il peut n'exister aucun hyperplan séparateur parfait (par exemple, aucun point ne permet de séparer les "points bleus" des "points rouges" dans la Figure1.5*)* ou au contraire une infinité de séparateurs différents possibles : les dessins de la Figure 1.6 montrent deux droites possibles pour nos données d'exemple.

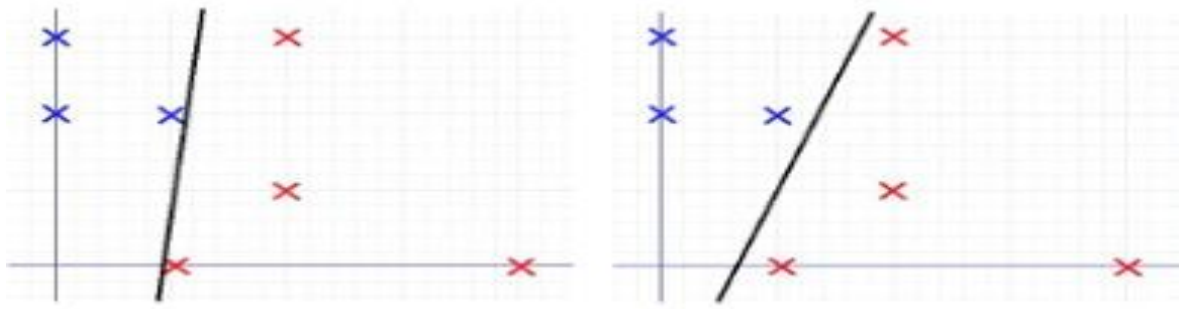


Figure.1.6 : Deux droites séparatrices possibles pour notre exemple [9]

1.5.4. Les k plus proches voisins (K-PPV) :

La méthode des plus proches voisins (noté parfois k-PPV ou k-NN pour -Nearest-Neighbor) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, la plus classique est la distance euclidienne, et le nombre de voisins à prendre en compte. Cette méthode supervisée et non-paramétrique est souvent performante. De plus, son apprentissage est assez simple, car il est de type apprentissage par cœur (on garde tous les exemples d'apprentissage). Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte [Berrani&al, 2002].[10]

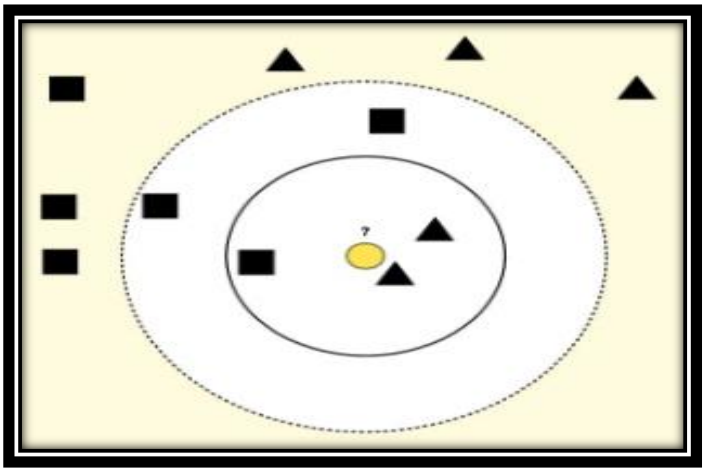


Figure.1.7: illustration des k-pp d'un point [10]

1.5.5. Naïve Bayes

Il s'agit d'un classificateur qui présente des avantages par rapport aux algorithmes de régression logistique et de l'algorithme k-plus proche voisin, lorsqu'il travaille avec une quantité limitée de données disponibles pour former un modèle. L'algorithme Naïve Bayes est également un bon choix lorsque les ressources CPU et mémoire sont un facteur limitant. L'algorithme Naïve Bayes est souvent le premier algorithme que les scientifiques essaient d'utiliser pour analyser du texte. C'est un algorithme de classification qui applique

une estimation de densité aux données.

L'algorithme exploite le théorème de Bayes et suppose que les données prédites sont conditionnellement in dépendantes comme présente l'équation suivante.[11]

$$P(A/B) = \frac{p(B/A)\times P(A)P(B)}{p(B)} \qquad (1)$$

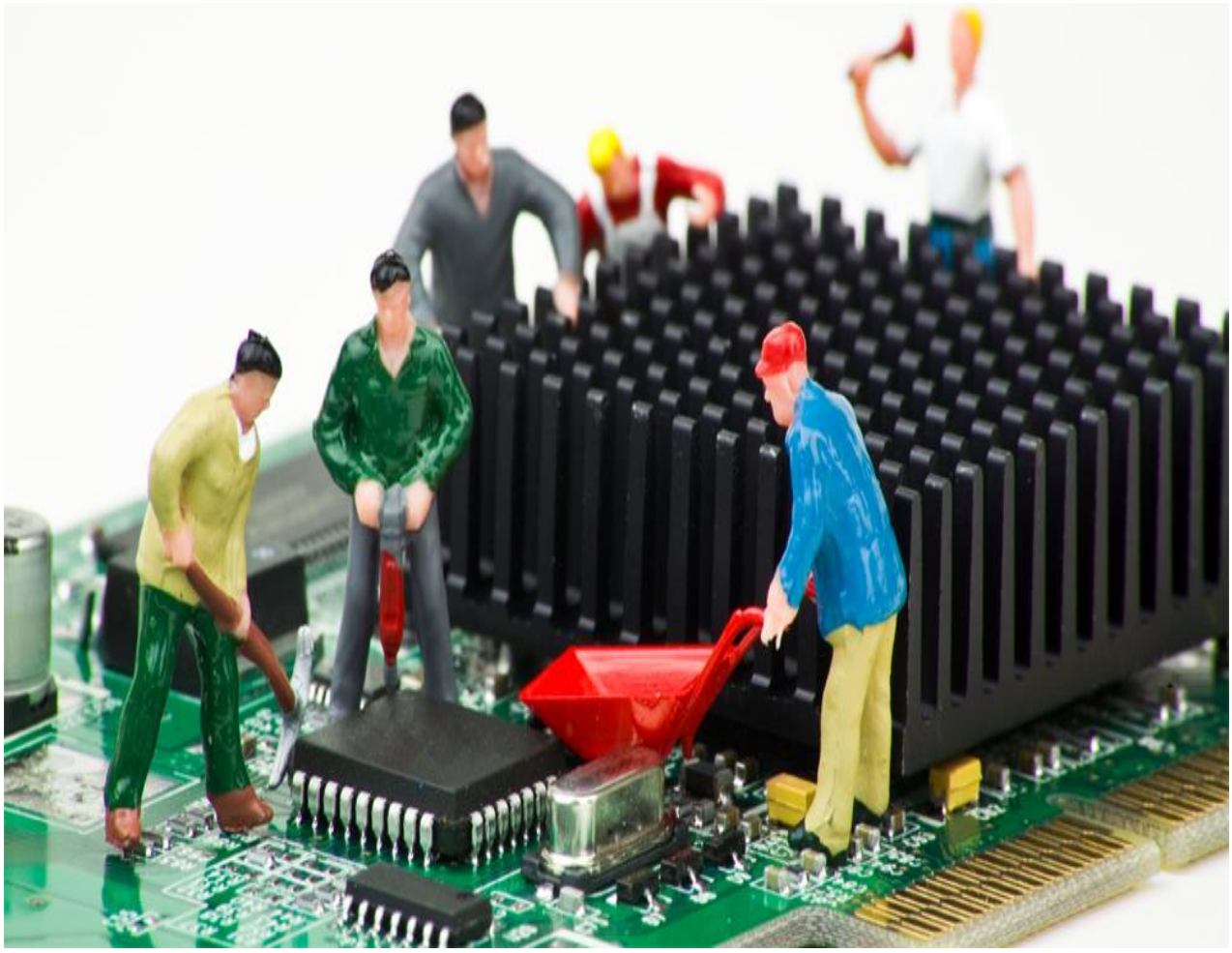
1.6. Conclusions

La fouille de données est une discipline émergente combinant les mathématiques/ statistiques et l'informatique et faisant usage des algorithmes d'apprentissage automatique. Ce chapitre vise principalement à présenter les techniques de fouille de données.

La fouille de données vise à extraire à partir de grandes quantités de données, les informations utiles qui peuvent aider à comprendre les données ou à prédire le comportement des données futures. Le datamining utilise plusieurs outils de statistiques et d'intelligence artificielle pour atteindre ses objectifs. Ce domaine est souvent appelé le datamining. Après ce chapitre, nous passerons au deuxième chapitre sous le titre sélection d'attributs pour la classification.

Chapitre II :

Sélection d'attributs pour la classification



2.1. Introduction

Actuellement les bases de données des entreprises sont en augmentation continue, dont beaucoup de données sont non significatifs ou redondantes; ce qui crées du bruit et rendre les modèles incapables de faire la bonne prédiction des données. En effet pour remédier à ce problème, et pour plus de performance, nous utilisons la sélection des attributs (de variables) comme une phase préliminaire pour réduire la dimensionnalité des données afin de permettre de découvrir les meilleurs modèles et de réduire la complexité temporelle de génération des modèles. La sélection des données est devenue un sujet de recherche très actif depuis des années dans le domaine de traitement d'image, de fouilles de données et les domaines d'apprentissage artificiels en général. La sélection d'attributs consiste à choisir parmi un ensemble d'attributs de très grande taille, un sous-ensemble d'attributs plus petit qui sont importants (significatifs) pour le problème étudié. Les principales conséquences de la sélection d'attributs sont : permet de déterminer les attributs considérés comme pertinents pour avoir un sous ensemble de données optimal d'attributs, permet de supprimer le bruit généré par les variables non pertinentes et / ou redondantes ainsi que leur sur-apprentissage, le petit sous-ensemble choisi permet de réduire l'espace de représentation et de réduire aussi le coût de la phase d'apprentissage et d'exécution des algorithmes de fouille de données. Elle permet aussi une meilleure compréhension du problème et améliore la précision et augmente la performance.

2.2. Formulation du problème de classification

Le problème de classification rentre dans le cadre de l'apprentissage automatique supervisé. Le but est de prévoir la classe y d'un vecteur p-dimensionnel x en se basant sur les mesures des variables qui l'expliquent avec pour seule information celle contenue dans un échantillon d'apprentissage S.

Dans le cas d'une classification bi-classe, nous supposons que les données sont des couples (X, Y), où X désigne l'espace des variables explicatives et $Y = \{-1, +1\}$ et l est la taille de l'échantillon. L'appartenance d'une observation son étiquette à une classe ou à une autre est matérialisée ici par la valeur -1 ou $+1$ de . L'échantillon d'apprentissage S est souvent dénoté par:

$$S = \{(x_1,y_1), (x_2,y_2) \dots (x_l,y_l)\} \in (X*Y)^l \quad (1)$$

Le problème consiste alors, en s'appuyant sur l'ensemble d'exemples S à prédire la classe de toute nouvelle donnée. En termes de vocabulaire on utilise le mot « étiquette » comme synonyme de « classe ».On parle de classification binaire quand le nombre de classes *Y* est égal à 2; il peut naturellement être quelconque. Dans tous les cas, il s'agit d'un attribut qualitatif pouvant prendre un nombre fini de valeurs.

Un « classifieur » est un algorithme qui, à partir d'un ensemble d'exemples, produit une prédiction de la classe de toute nouvelle donnée. D'une manière générale, un classifieur procède par « induction » : à partir d'exemples, on construit une connaissance plus générale.[12]

2.3. Classification de données de grande dimension

Les difficultés qui apparaissent lors de la manipulation des donnés de grande dimension, sont attribués au phénomène appelé« fléau de dimensionnalité ». Quand la dimension des données augmente, les propriétés bien connus de l'espace euclidien usuel 2D et 3D ne sont plus valables. Ces données de grande dimension posent différents problèmes, dans le contexte de la classification supervise ainsi que la classification non supervisée ou clustering.

2.4. Réduction de la dimensionnalité

La réduction de la dimensionnalité consiste à récupérer des données dans un espace de grande dimension et à les remplacer par des données dans un espace plus petit. Nous parlons du fléau de la dimensionnalité, produisant trop de résultats à corréler ou à comparer, sans parler du temps qu'il faut pour traiter ces quantités massives de données. Pour réduire la dimensionnalité, il existe différentes approches, telles que la sélection de certains attributs (sélection d'attributs) ou la recréation complète de nouvelles données dans un espace plus limité (extraction d'attributs).

2.4.1. Sélection des attributs (fonctionnalités)

La sélection des fonctionnalités est une partie importante de l'apprentissage automatique. Elle désigne le processus visant à réduire les entrées à traiter et à analyser, ou à identifier les entrées les plus pertinentes. La sélection des fonctionnalités est toujours effectuée avant l'apprentissage du modèle. Avec certains algorithmes, les techniques de sélection des fonctionnalités sont « intégrées » pour exclure les colonnes non pertinentes et détecter automatiquement les meilleurs attributs. Chaque algorithme a son propre ensemble de techniques par défaut pour appliquer intelligemment la réduction de fonctionnalités. Toutefois, il est possible également de définir manuellement des paramètres pour influencer le comportement de la sélection des fonctionnalités

Lorsque le calcul du score de sélection des fonctionnalités est terminé, seuls les attributs et les états que l'algorithme sélectionne sont inclus dans le processus de génération de modèle de fouille de données.

2.4.2. Extraction des attributs

L'extraction des propriétés pertinentes est une tâche cruciale et incontournable pour la caractérisation des données. Elle désigne le processus visant à extraire des nouvelles informations ou des attributs utiles à partir de données existantes avec une équivalence entre les deux ensembles de données.

2.5. Sélection des attributs

2.5.1. Définitions

- **Attributs pertinents**

Plusieurs définitions de pertinence existent dans la littérature. Dans (John et al. [1994], John [1997]), les auteurs stipulent qu'il existe trois types de attributs : très pertinent, peut pertinent et non pertinent. [13]

- **Très pertinent** : Un attribut t_i est dit très pertinent si son absence cause une dégradation significative de la performance du classificateur
- **Peu pertinente** : Un attribut t_i est dit peu pertinent s'il n'est pas "très pertinent" et s'il existe sous-ensemble V tel que la performance de $V \cup \{t_i\}$ soit nettement meilleure que celle de V.
- **Non pertinente** : Les attributs qui ne sont ni "peu pertinents" ni "très pertinents" représentent les attributs non pertinents. Ces attributs seront en général supprimés de l'ensemble d'attributs originaux.

- **Attribut redondant**

Un attribut est dit redondant s'il peut être dérivé de tout autre attribut ou ensemble d'attributs. Les incohérences dans la dénomination des attributs ou des dimensions peuvent également entraîner des redondances dans les ensembles de données.

- **Sélections d'attributs**

En général, le problème de sélection d'attributs peut être défini comme suit : Etant donné un espace d'attributs $S=\{\mathbf{x_i} \mid i=1\ldots p\}$, *Y* la sortie, on veut trouver un sous-ensemble de ces attributs, qui prédit bien la sortie, soit $Sm=\{\mathbf{x_{i1}, x_{i2}, \ldots, x_{im}}\}$, avec $m < p$ (Figure 2.1). Pour sélectionner ce sous-ensemble, on cherche en général à formaliser le problème comme un

problème consistant à optimiser une fonction objectif (ou critère) *J*. Ce critère peut, par exemple, être la probabilité d'erreur dans le cas d'un problème de classification, dans ce cas, la sélection des attributs consiste à trouver le sous-ensemble de plus faible dimension qui conduit à une faible erreur de classification. Ce sous ensemble d'attributs est utilisé ensuite pour l'apprentissage d'une règle de classification.[14]

Figure 2.1. Sélections d'attributs [14]

2.5.2. Pourquoi utiliser la sélection des attributs

La sélection des attributs est essentielle à la création de modèles réussis pour un certain nombre de raisons. La première est que la sélection des fonctionnalités implique un certain degré de réduction de la cardinalité pour limiter le nombre d'attributs pouvant être pris en compte lors de la création d'un modèle. La sélection des attributs améliore la qualité du modèle tout en optimisant le processus de modélisation. Lors de la sélection des attributs, les analystes ou les outils ou algorithmes de modélisation sélectionnent ou ignorent activement les attributs en fonction de leur utilité pour l'analyse.

Les analystes peuvent effectuer une ingénierie des attributs pour ajouter des fonctionnalités et supprimer ou modifier des données existantes, tandis que les algorithmes d'apprentissage automatique calculent souvent des scores de colonne pour valider leur utilité dans les modèles. Dans l'ensemble, la sélection des attributs peut éviter soit d'avoir trop de données inintéressantes, soit de ne pas avoir assez de données utiles. Donc l'objectif est de déterminer le nombre minimal de colonnes de la source de données qui sont importantes pour le modèle à créer.

2.5.3. Processus de sélection d'attributs

Une procédure générale pour élaborer une méthode de sélection d'attributs est illustrée par la Figure 2.2 On distingue 4 étapes pour la sélection des attributs en commençant par l'ensemble initial des attributs : la génération du sous-ensemble, l'évaluation du sous-ensemble, le critère d'arrêt, et la validation des résultats.[14]

- **La génération de sous-ensembles :** est une stratégie de recherche pour déterminer un sous-ensemble d'attributs candidats à l'évaluation
- **Évaluation des sous-ensembles** : estimer certains critères d'évaluation pour mesurer la qualité des sous-ensembles candidats. En répétant ce processus, le sous-ensemble associé à la meilleure valeur du critère est sélectionné.
- **Critère d'arrêt**: il est nécessaire que chaque sous-ensemble d'attributs après l'évaluation soit comparé au critère d'arrêt pour vérifier si l'objectif visé est atteint. Si les exigences sont vérifiées, la sélection d'attributs s'arrête et le sous-ensemble courant est considéré comme le résultat final; sinon le processus de recherche continue.
- **La validation**: le sous-ensemble choisi doit généralement être validé par différents tests avec des données du monde réel ou non réel.

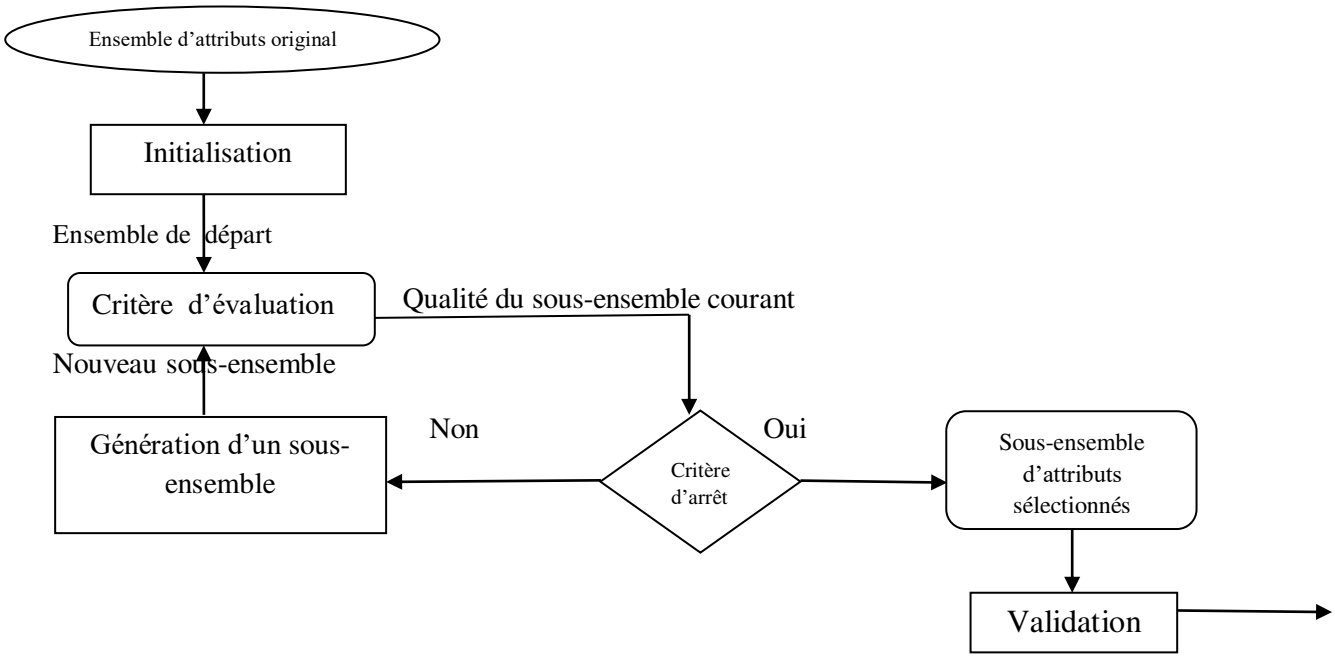


Figure.2.2.Processus de sélection d'attributs [15]

2.5.4. Les procédures de recherches

En général, les stratégies de recherche pour la génération des sous-ensembles dans le processus de sélection des attributs peuvent être classées en trois catégories : exhaustive, heuristique et aléatoire [14].

a) **Génération exhaustive ou complète:** Dans cette approche, on examine tous les sous-ensembles d'attributs possibles afin de sélectionner le "meilleur" sous-ensemble d'attributs. Dans ce cas il est garanti de trouver le sous-ensemble optimal, mais cette méthode nécessite trop de calcul et de temps.

b) **Génération heuristique:** Les algorithmes qui utilisent la recherche heuristique sont généralement des algorithmes itératifs. Les algorithmes basés sur la recherche heuristique sont simples et rapides, mais ils peuvent conduire à une solution sous optimale.

c) **Génération aléatoire:** Pour la recherche aléatoire il n'existe pas une méthode prédéfinie pour le choix. L'approche de recherche aléatoire consiste à générer aléatoirement les sous-ensembles d'attributs afin de sélectionner le meilleur.

2.5.5. Les procédures d'évaluation

L'évaluation constitue une partie importante de la sélection d'attributs. On peut distinguer trois catégories pour l'évaluation dans les algorithmes de sélection: "filter", "wrapper" et "embedded"[14]

- **Filter méthodes**: Cette technique est généralement utilisée pour la sélection d'attributs avec une évaluation basée sur les résultats de classifieur. Cette technique est considérée comme une étape de prétraitement avant le processus d'apprentissage, cela signifie que l'évaluation de cette technique est indépendante du classificateur. Le meilleur sous-ensemble d'attributs obtenus par cette technique est présenté en entrée de l'algorithme de classification.

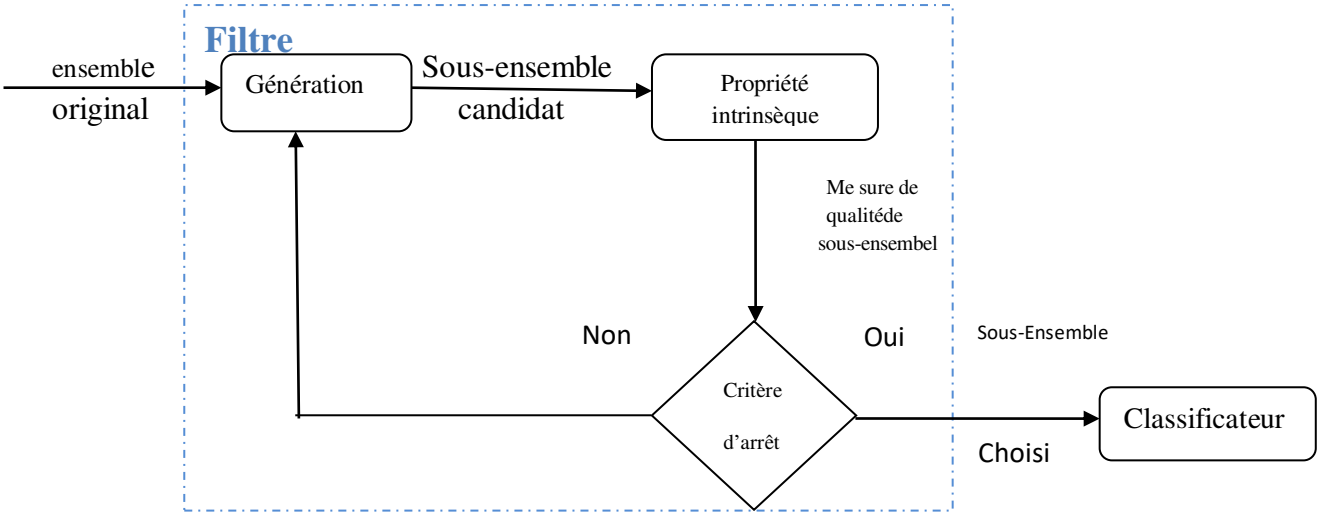


Figure 2.3. Sélection d'attributs à base de filtre [14]

- ***Wrapper méthode :*** Les méthodes wrapper sont basées sur des algorithmes de recherche avide, car elles évaluent toutes les combinaisons possibles de caractéristiques et sélectionnent la combinaison qui produit le meilleur résultat pour un algorithme d'apprentissage automatique spécifique. L'inconvénient de cette approche est que le test de toutes les combinaisons possibles de caractéristiques peut être très coûteux en termes de calcul, en particulier si l'ensemble de caractéristiques est très grand.

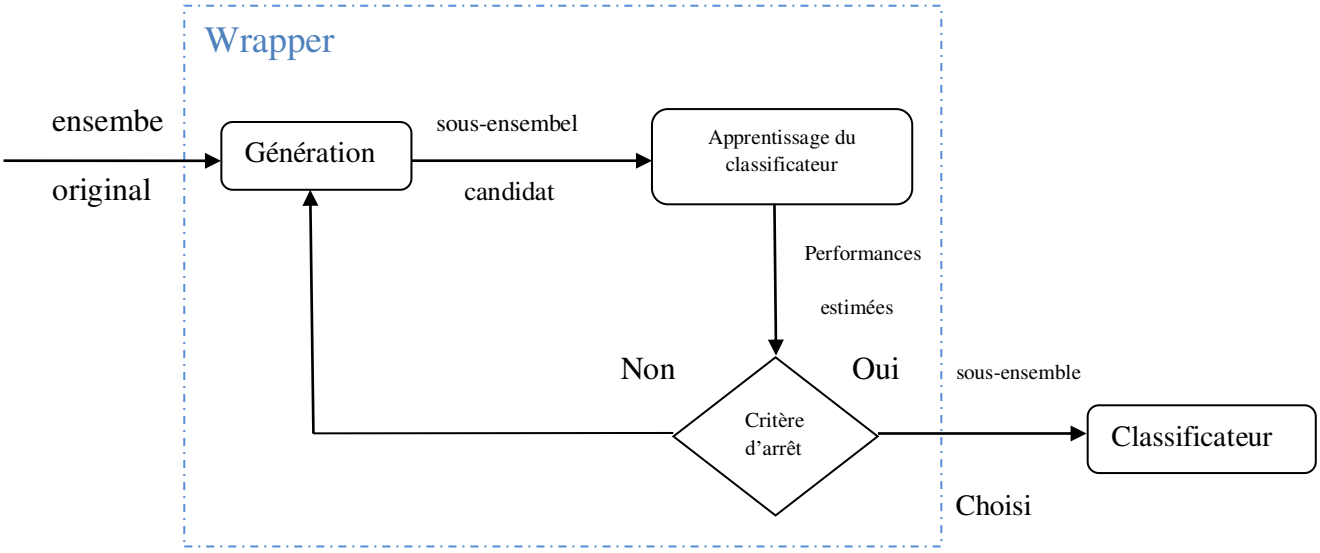


Figure 2.4. Sélection d'attributs à base Wrapper. [14]

- **Embedded méthode:** Récemment, les méthodes embedded ont été proposées en classification pour diminuer le sur-apprentissage. L'algorithme d'apprentissage utilise son propre algorithme de sélection des attributs et nécessite donc de pouvoir caractériser a priori ce que serait une bonne sélection, ce qui limite leur utilisation. Ces approches incorporent la sélection des attributs lors du processus d'apprentissage, sans étape de validation, pour maximiser la qualité de l'ajustement et minimiser le nombre de variables.

Cette méthode est proche de celle de wrapper car elle combine le processus d'exploration avec l'algorithme d'Apprentissage .Cependant, l'ajout des méthodes wrapper est que le classifieur sert non seulement à évaluer un sous ensemble candidats mais aussi à guider le mécanisme de sélection. Un exemple très connu est celui des arbres de décision, où les variables sélectionnées sont celles présentes au niveau de la division de chaque nœud.

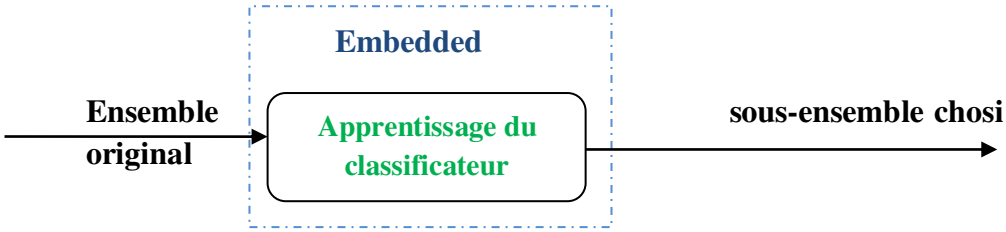


Figure 2.5. Sélection d'attributs à base Embedded [14]

2.5.6. Les critères d'arrêt

Certains critères doivent être définis pour arrêter le processus de recherche sur les sous-ensembles d'attributs. Certains critères d'arrêt couramment utilisés sont basés sur l'ordre d'attributs classés selon un score de pertinence.

2.5.7. La validation

Un processus de validation sert àvérifier si le sous-ensemble trouvé est valide on non. Ce processus n'appartient pas, à proprement parler, à la procédure de sélection elle-même, mais cette étape est à envisager en pratique. Il s'agit de vérifier la validité du sous-ensemble en faisant différents tests, et de comparer les résultats soit par validation croisée, soit en comparant les méthodes de sélection avec des jeux de données matricielles et/ou réelles.

2.6. Catégorisation des méthodes de sélection d'attributs

Dans la littérature, différent méthodes ont été développées et utilisées pour la sélection d'attributs. Les auteurs proposent plusieurs catégorisations de ces méthodes selon les stratégies de recherche et les fonctions d'évaluations utilisées.

2.6.1. Catégorisation basée sur les critères d'évaluation

Certains auteurs préfèrent classer les méthodes de sélection de caractéristiques en trois approches (filtres, wrappers et hybrides) en fonction de leurs dépendances ou de leur indépendance par rapport à l'algorithme d'apprentissage.

Alors que nous avons abordé précédemment les méthodes (filtres, wrappers) (voir section 2.5.5) Et nous vous expliquerons la méthode hybrides .

- **L'approche hybride**

Les algorithmes hybrides ont été plus récemment proposés par rapport aux précédents (filtre et wrapper). Dans ces algorithmes, d'abord, une mesure de «goodness» de sous-ensembles de caractéristiques, basée sur les caractéristiques des données, est utilisée pour choisir les meilleurs sous-ensembles pour une cardinalité donnée. Par la suite, la validation croisée est exploitée pour décider du meilleur sous-ensemble final entre les différentes cardinalités. Ces algorithmes se concentrent principalement sur la combinaison des algorithmes Filtre et Wrapper pour obtenir les meilleures performances possibles, avec un algorithme d'apprentissage particulier et une complexité temporelle comparable à celle des algorithmes de type Filtre.

2.6.2. Catégorisation basée sur la stratégie de recherche

Une autre classification des méthodes de sélection de caractéristiques est la classification basée sur les stratégies de recherche. Yu Liu proposent de séparer les méthodes selon la stratégie utilisée (complète, heuristique ou aléatoire).[16]

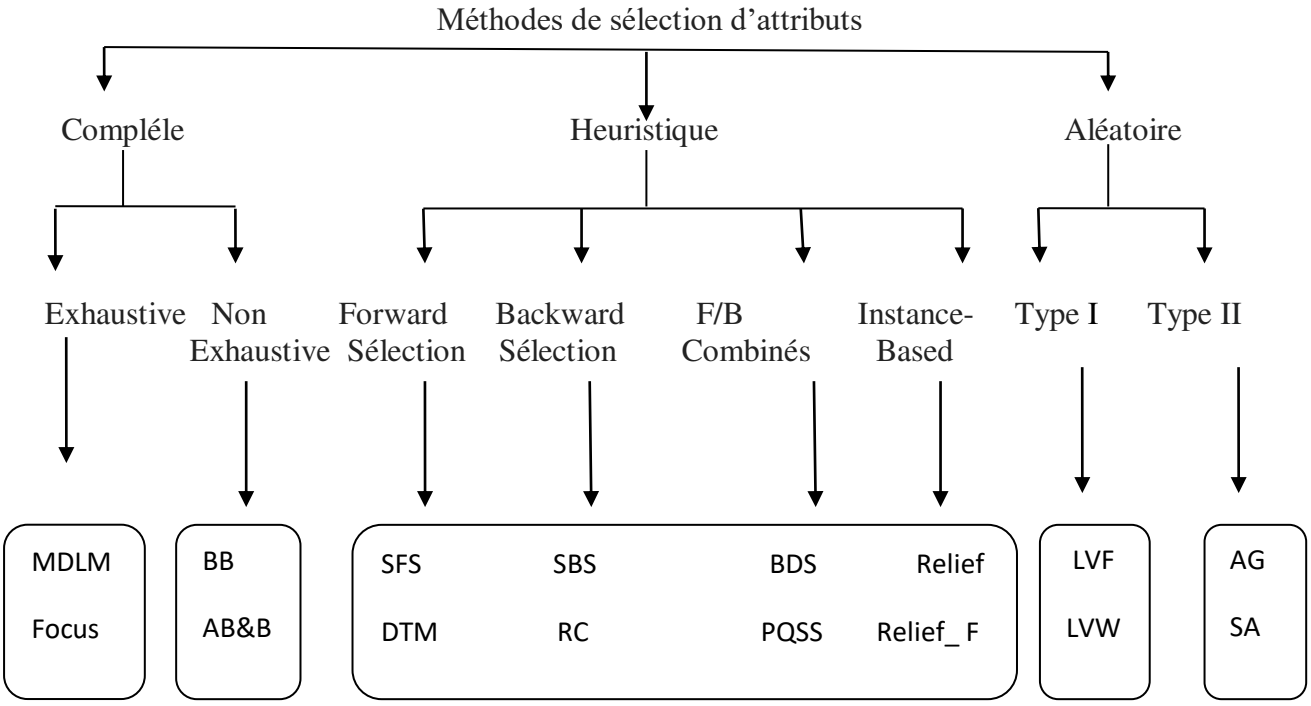


Figure 2.6. Catégorisation des méthodes de sélection de caractéristiques [16]

La Figure 2.6 présente une classification des méthodes de sélection de caractéristiques basée sur les trois types de procédures de génération.

- **La procédure de génération**

Le processus génératif est un processus de recherche qui permet l'exploration de l'espace de recherche pour construire différentes combinaisons de caractéristiques. En fait, le choix d'un point dans l'espace du sous-ensemble d'entités pour démarrer la recherche affecte la direction de la recherche. Ce dernier peut commencer avec un ensemble vide de fonctionnalités, puis ajouter des fonctionnalités de manière séquentielle. Au lieu de cela, la recherche peut commencer par toutes les fonctionnalités et en supprimer certaines à tour de rôle. Pour un jeu de données avec caractéristiques, il existe 2N sous-ensembles candidats.

Par conséquent, différentes stratégies ont été explorées: recherche complète, séquentielle et aléatoire.

- **Recherche complète**

Les méthodes de cette catégorie effectuent une recherche complète du meilleur sous-ensemble de la fonction d'évaluation sélectionnée. Utilise différentes fonctions d'évaluation pour réduire l'espace de recherche sans perdre la chance de trouver le meilleur sous-ensemble.

- **Recherche séquentielle ou heuristique**

Cette catégorie rassemble des algorithmes itératifs, chaque itération peut sélectionner ou rejeter une ou plusieurs fonctionnalités. Les algorithmes à génération séquentielle sont faciles à mettre en œuvre et produisent des résultats rapidement, en utilisant un espace de recherche d'ordre O (2N). Dans ces algorithmes, l'exhaustivité est abandonnée, il y a donc un risque de perdre le meilleur sous-ensemble.

- **Recherche aléatoire**

Par rapport aux deux premières classes, l'utilisation de la génération aléatoire dans le processus de sélection des caractéristiques est à la pointe de la technologie. Le processus commence par un sous-ensemble sélectionné au hasard et se déroule de deux manières différentes. Le choix des valeurs appropriées pour les paramètres du processus de recherche est une étape très importante pour obtenir des résultats satisfaisants.

2.7. Conclusion

La sélection des attributs est un problème majeur dans de nombreux domaines vu l'augmentation des tailles des bases de données. Par conséquent, c'était un sujet d'intérêt pour de nombreux chercheurs, de sorte que nous avons discuté dans ce chapitre de l'importance et de la nécessité de réduction de la dimension. Ensuite, dans le chapitre suivant, nous avons utilisé quelques techniques de sélection d'attributs pour des bases de données médicales afin de déduire les attributs les plus importants de ce domaine médicale. Après ce chapitre, nous passerons au théorisâmes chapitre de pratique sous le titre Sélection d'attributs et classifications pour le diagnostique des maladies chronique.

Chapitre III : *Approche de sélection d'attribut et implémentation*

3.1 Introduction

Dans ce dernier chapitre de notre mémoire nous présentons notre participation dans la sélection d'attributs et classification pour le diagnostique des maladies chroniques. Nous présentons d'abord les bases de données (data set) utilisées pour l'apprentissage et l'évaluation de la technique de sélection des attributs (CAD, PID et BCDS) .Nous avons choisir une seule base de données pour une tel maladie, et nous décrivons ses caractéristiques. Quatre algorithmes de classification ont été appliqués (SVM, KNN, Naive-bayes, DecisionTree). On a également présenté et identifié tous les outils et packages utilisés dans l'implémentation des algorithmes. Ce chapitre est divisé en deux parties, la première partie expose l'implémentation de la méthode de classification SelectKBest, alors que la seconde partie est consacré à exposer les résultats des expérimentations.

3.1.Méthodologie de sélection d'attributs

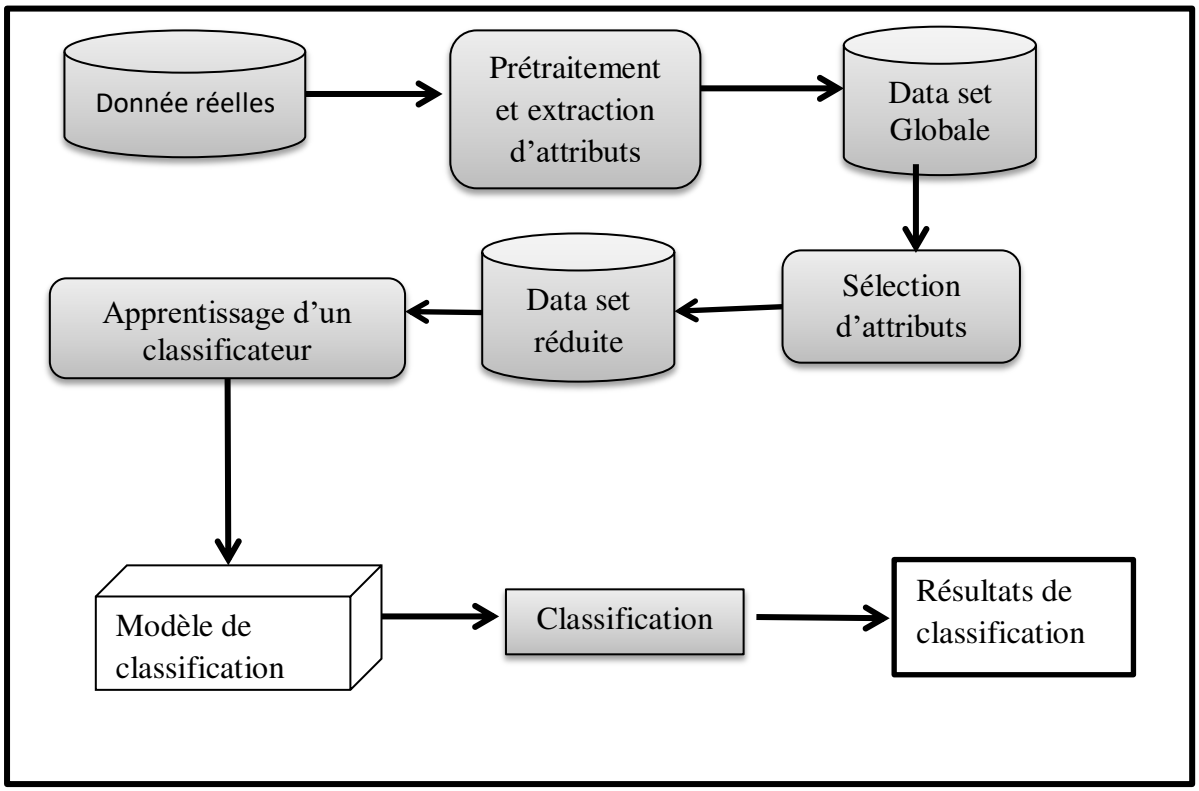


Figure 3.1. Schéma d'approche de sélection d'attribut et de classification

3.2. Outils utilisés

Pour implémenter l'approche de sélection des attributs pour le diagnostic médicale on à utiliser le langage de programmation le plus adéquate pour ce type de programmation qui est Python. Pour cela plusieurs packages sont utilisés.

- **Jupyter Notebook**

Anciennement appelé IPython Notebooks. Le Jupyter Notebook est une application web open-source permettant de créer et de partager des documents contenant du code, des équations, des visualisations, et du texte narratif. Il s'agit d'un environnement de calcul interactif basé sur le web permettant de créer des documents notebooks. [17]

- **Python**

Python est un langage de programmation multi-paradigme et le langage de programmation dominant dans la science des données (data science) avec de nombreuses implémentations ce qui le rend encore plus intéressant. Concernant le domaine de l'apprentissage automatique, Python se distingue tout particulièrement en orant une pléthore de librairies de très grande qualité, couvrant tous les types d'apprentissages disponibles qui combine la facilité d'utilisation et d'apprentissage avec la puissance des librairies qu'elles possèdent.

3.3. Prétraitement des données

Le prétraitement est l'ensemble de technique qui transforme les données brutes en un format compréhensible. Les données ne peuvent pas être envoyées via des modèles car cela entraînerait certaines erreurs et ces données du monde réel (données brutes) sont toujours incomplètes. C'est pourquoi nous devons prétraiter les données avant de les envoyer via un modèle. Le processus de prétraitement des données est divisé en plusieurs étapes:

- Lecture de l'ensemble de données.
- Divisez l'ensemble de données en indépendant et dépendant.
- Gestion des valeurs manquantes.
- Gestion des valeurs catégorielles.
- Standardisation / mise à l'échelle des fonctionnalités.

a) Importation des bibliothèques

La première étape consiste généralement à importer les bibliothèques qui seront nécessaires dans le programme. Une bibliothèque est essentiellement une collection de modules qui peuvent être appelés et utilises, nous allons utiliser la bibliothèque Pandas et Numpy.

Pandas : Nous utilisons des pandas pour la manipulation et l'analyse des données.

Numpy : C'est un package fondamental pour le calcul scientifique avec Python.

b) Lecture des bases de données (data sets)

Les bases de données sont au format .csv, .Arff et .xlsx, Il est important de conserver l'ensemble de données dans le même dossier dans le programme et de le lire à l'aide d'une méthode appelée read_csv et pd.read_excel qui se trouvent dans la bibliothèque pandas.

```
PID = read_csv('C:\\Users\\romai\OneDrive\\Documents\\PID.csv')
PID
```

c) Division des données en fonctionnalités indépendantes et dépendantes

Nous allons créer une matrice de caractéristiques (attributs) dans notre jeu de données en créant une variable indépendante (X) et une variable dépendante (Y). Pour lire les colonnes, nous utiliserons *iloc* de pandas qui prend deux paramètres [sélection de ligne, sélection de colonne] :

```
x = CAD.iloc[:,4:-2].values
y = CAD.iloc[:,-1].values
```

d) Gestion des données catégoriques

Nos données sont sous forme de texte. Nous pouvons trouver des catégories sous forme de texte. Cela devient compliqué pour les machines pour comprendre les textes et de les traiter car les modèles sont basés sur des équations et des calculs mathématiques. Par conséquent, nous devons encoder les données en nombres. Dans le même temps, nous pouvons également conserver des variables catégorielles pour la visualisation des données.

```
num = {'class' : {'tested_positive':0,'tested_negative':1}}
PID.replace(num, inplace=True)
num
```

e) Standardisation / mise à l'échelle des fonctionnalités

Il est utilisé pour standardiser les valeurs des variables indépendantes. C'est une méthode utilisée pour limiter la plage de variables afin de pouvoir les comparer facilement. La plupart des modèles d'apprentissage automatique sont basés sur des distances euclidiennes. La différence carrée avec la valeur inférieure par rapport à la valeur beaucoup plus grande sera presque traitée comme si elle n'existait pas. Nous ne voulons pas que cela se produise. C'est pourquoi il est nécessaire de transformer toutes nos variables dans la même échelle. Nous devons importer StandardScaler à partir de la bibliothèque de prétraitement sklearn et créer un objet de cette classe.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_tr = sc.fit_transform(x_tr)
x_ts = sc.fit_transform(x_ts)
```

f) Importation de la bibliothèque Scikit learn

Scikit-learn est une bibliothèque centrale dans l'utilisation de la machine learning en Python. De plus cette bibliothèque s'utilise conjointement avec les autres bibliothèques Python : Numpy, Pandas (pour la manipulation des tableaux et de séries) .De plus, la similarité des syntaxes ne permet pas de comparer facilement plusieurs modèles pour affiner les prédictions et trouver le meilleur modèle. C'est très appréciable. Scikit-learn s'installe comme la majorité des paquets Python [18].

Les algorithmes de classification utilisés de Scikit-learn comprennent :

- les machines à vecteurs de support (SVM).
- Voisins les plus proches.
- Naive_bayes.
- DecisionTree.

3.5 Approche de Sélection d'attribut

La sélection d'attribut est l'une des étapes importantes lors de la création d'un modèle d'apprentissage automatique. L'utilisation de la sélection d'attribut permet d'éliminer les données redondantes ou non pertinentes, et cette élimination peut renforcer les résultats de l'apprentissage automatique et peut augmenter la précision et améliorer les performances du classificateur. Parmi les méthodes de sélection d'attributs" il en existe deux méthodes qui sont les plus utilisées : la méthode filtrage et la méthode emballage .Nous avons utilisé la méthode filter par ce qu'elle est très rapides, peu couteux et sont très efficaces pour supprimer les caractéristiques dupliquées, corrélées et redondantes. Classe les attributs par ordre d'importance" ce qui nous laisse le choix de choisir les attributs qu'on veut garder et Calcule l'importance de chaque attribut dans la prise de décision. La méthode filtrer consiste de plusieurs techniques, la technique qui on a utilisé est SelectKBest. Il compare les valeurs observées de différents attributs de l'ensemble de données à sa valeur attendue.

- **Implémentation de la méthode** SelectKBest

Set all feature → selecting the best subset → learning algorithme ⟶ performance.

Tout d'abord nous appliquerons le modèle SelectKBest à la classification. Nous définirons le modèle en utilisant la classe SelectKBest. Pour la classification, nous définirons la méthode 'chi2' comme fonction de notation. Le nombre cible de fonctionnalités est défini par le paramètre k. Ensuite, nous ajusterons et transformerons la méthode sur la formation des données x et y. Nous avons sélectionné les meilleurs attributs dans x data. Pour identifier les attributs sélectionnées, nous utilisons la fonction get_support() et les filtrons de la liste des noms d'attributs. L'objet z contient les données x sélectionnées.

3.6 Approche de classification

L'algorithme de classification est une technique d'apprentissage supervisé qui est utilisée pour assimiler la catégorie de nouvelles observations sur la base de données d'apprentissage. Dans la classification, un programme apprend à partir d'un ensemble de données ou d'observations données, puis classe les nouvelles observations dans un certain nombre de classes ou de groupes. Dans cette partie, nous essayerons de générer des classifieurs en python en utilisant les algorithmes suivant:

- K-Nearest Neighbors (K plus proche voisins)
- Kernel SVM.
- Naïve Bayes (réseaux bayésiens) .
- Classification par arbre de décision.

Ces algorithmes sont implémentés et ses résultats sont ensuite comparés.

Le fondement de Scikit-learn est à chaque fois le même :

- instanciation du modèle.
- entraînement du modèle sur les features X et les targets Y.
- utilisation du modèle pour prédire de nouvelles données.
- mesure de l'efficacité du modèle.

Comme tout algorithme d'apprentissage automatique, avnant d'appeler l'algorithme il faut d'abord diviser le data set en deux sous datasets l'une pour l'apprentissage et l'autre pour le test.

3.6.1. Division des données d'entraînement et de test

Nous pouvons entraîner le modèle en utilisant des données que nous appelons données d'entraînement ou ensemble d'entraînement. Les données d'apprentissage sont celles qui présentent déjà la valeur réelle que le modèle aurait dû prédire. L'algorithme modifie donc la valeur des paramètres pour tenir compte des données de l'ensemble d'apprentissage. Pour cela, nous utilisons des données de test/ensemble de test qui sont essentiellement des données différentes dont nous connaissons les valeurs mais qui n'ont jamais été montrées au modèle auparavant. Ainsi, si le modèle, après la formation, est performant sur l'ensemble de test, nous pouvons dire que le modèle d'apprentissage automatique est bon.

Nous pouvons diviser les données en utilisant la fonction train_test_split de la bibliothèque model_selection du paquet Sklearn. Elle prend 4 variables en sortie que nous avons nommée comme suit :

- x_train : Données de formation en entrée.
- x_test : Données de test en entrée.
- y_train : Données de formation en sortie.
- y_test : Données de test en sortie.

Il y a un autre paramètre appelé random_state qui est utilisé pour initialiser le générateur de nombres aléatoires interne. Il décidera également pour la division des données en valeurs d'entraînement et de test. Pour cela, le test_size = 0,25 indique que 25 % des données seront conservées en tant qu'ensemble de test et que les 75 % restants seront utilisés pour la formation en tant qu'ensemble d'apprentissage.

3.6.2. Appel du classificateur

Nous avons utilisé et comparer les résultats de quatre classifieurs suivant :

a) le classificateur KNN

Nous allons utiliser l'algorithme K-Nearest Neighbors Classifier, En utilisant le package Sklearn, importez KNearestClassifier de la bibliothèque neighbors. Ici aussi, nous considérons que la valeur de n_voisins est de 7. Chaque division des données est appelée un fold. La manière efficace de choisir k est de prendre la racine carrée des données de formation et de considérer le nombre impair le plus proche et ajuster le modèle.

b) Le classificateur SVM

Nous avons importé la classe de classification SVM et adapter l'ensemble de formation à notre modèle. L'étape consiste à instancier un SVC (Support Vector Classifier) est affectée au classifieur variable, la classe SVC du module sklearn.svm est utilisée. Le noyau utilisé ici est le noyau « rbf » qui signifie Radial Basis Function. Il existe plusieurs autres noyaux tels que le noyau linéaire et le noyau gaussien qui peuvent également être implémentés. La fonction classifier.fit() est ensuite utilisée pour former et entraîner le modèle sur les features X et les targets Y : model.fit(X, Y). À la fin de l'entraînement, le modèle est prêt à être utilisé pour prédire des targets.

c) Le classificateur Naive -Bayes

Pour cet algorithme, nous introduisons la classe GaussianNB qui est utilisée à partir de la bibliothèque sklearn.naive_bayes. Ici, nous avons utilisé un modèle gaussien, il existe plusieurs autres modèles tels que Bernoulli, Catégorique et Multinomial. Ici, nous attribuons la classe GaussianNB au classificateur de variable et y adaptons les valeurs X_train et y_train à des fins de formation**.**

d) Le classifieur DecisionTree

Nous avons importé DecisionTreeClassifier de la bibliothèque tree. En utilisant le package Sklearn, DecisionTreeClassifier qui est une classe capable d'effectuer une classification multi-classes sur un ensemble de données. Comme avec les autres classificateurs, DecisionTreeClassifier prend en entrée deux tableaux : un tableau X, clairsemé ou dense, de forme (n_samples, n_features) contenant les échantillons d'apprentissage, et un tableau Y de valeurs entières, forme (n_samples,), contenant les étiquettes de classe pour les échantillons d'apprentissage.

3.7. Expérimentation et résultats

3.7.1. Les bases de données (data sets)

La fouille de données est le processus de découverte de données cachées dans une base de données. De nos jours, la fouille de données est utilisée dans plusieurs domaines et a différentes applications, par exemple en entrepreneuriat, en détection de fraude et en science découvertes. Les algorithmes de fouille de données fonctionnent généralement sur des ensembles de données, dans lesquels chaque enregistrement de données possède plusieurs les attributs. Un attribut spécial est l'étiquette de classe, qui détermine la catégorie de données.

- L'UCI

Machine Learning Repository est un ensemble de bases de données, de théories de domaine et de générateurs de données qui sont utilisés par la communauté de l'apprentissage automatique pour l'analyse empirique des algorithmes d'apprentissage automatique. L'archive a été créée en tant qu'archive ftp en 1987 par David Aha et d'autres étudiants diplômés de l'UC Irvine. Depuis lors, il a été largement utilisé par les étudiants, les éducateurs et les chercheurs du monde entier comme source principale d'ensembles de données d'apprentissage automatique.

Comme indication de l'impact de l'archive, elle a été citée plus de 1000 fois, ce qui en fait l'un des 100 "articles" les plus cités de toute l'informatique [19].

Nous avons appliquerons l'analyse de données ainsi qu'une approche d'apprentissage automatique pour classer si une personne souffre d'une maladie chronique ou non en plus d'une sélection d'attribut pour la réduction des tailles des bases de données. On a utilisé les bases de données les plus utilisés dans le monde des maladies. Le jeu de données Breast Cancer Data (BCDS), le diagnostic CAD et le diabète des Indiens Pima (PID) du référentiel d'UCI, comme indiquer dans ce tableau 3.1 :

Datasets	**Code**	**Attributs**	**Instances**	**Class**
Maladie coronarienne	CAD	55	303	2
Breast cancer data	BCDS	9	286	2
Pima Indian Diabets	PID	8	768	2

Tableau 3.1 : Les caractéristiques des bases de données utilisées

a) CAD

La maladie cardiaque la plus courante est la maladie coronarienne (CAD). La CAO est l'une des principales causes de crises cardiaques et de décès dans le monde. Diagnostic précoce de cette maladie est donc d'une grande importance. Un grand nombre de méthodes ont jusqu'à présent été conçues pour diagnostiquer la CAD. La plupart de ces techniques ont été menées sur la base de l'Irvine ensemble de données (Université de Californie) [20]. Des informations ont été recueillies de Shaheed Rajaei Cardiovascular, Medical and Research Center, entre l'automne 2011 et Hiver 2012. L'ensemble de données comprenait 303 patterns par 55 caractéristiques appartenant deux class (CAD et Normal) 216 instance CAD et 87 normal.

b) BCDS

Aujourd'hui, le cancer du sein métastatique est devenu une maladie chronique avec de nouveaux traitements et des options différentes suivant les cas. Ce changement fondamental a bouleversé l'approche du cancer, une adaptation du corps médical et une nouvelle ouverture pour les patients et leur environnement .Cette base de donné été créé par Matjaz Zwitter et Milan Soklic à l'Université du Médical Centre [21]. Cette ensembles des données comprend 286 patterns par 9 caractéristiques appartenant à deux classe (no-recurrence-events, recurrence-events) 85 instances recurrence-events et 201 instances no-recurence-events.

c) PID

Le Pima Indian Diabetes contient 768 patterns avec 8 caractéristiques appartenant à deux classes (Not Diabetic, Diabetic). 268 échantillons ont été identifiés comme positifs et les 500 autres échantillons ont été identifiés comme négatifs. Les deux classes sont réparties de manière non-uniforme avec 34,9% des instances positives et 65,1% des instances négatives.

3.7.2. Critères d'évaluation utilisés

a) Accuracy Score

Après training, nous vérifions l'accuracy à l'aide des valeurs réelles et prédites, la méthode accuracy_score est utilisée pour calculer la précision de la faction ou du nombre de prédictions correctes dans Python Scikit learn. Nous utilisons la bibliothèque metrics. Model accuracy est une métrique de performance du modèle d'apprentissage automatique qui est définie comme le rapport des vrais positifs et des vrais négatifs à toutes les observations positives et négatives. accuracy est la proportion du nombre de tuples positifs et de tuples négatifs obtenus par les algorithmes de classification dans le nombre total d'occurrences, comme indiqué dans l'Équation 3.1.

$$\text{Accuracy Score} = \frac{TP+TN}{(TP+FN+FP+FN)} \qquad \textbf{(3.1)}$$

La signification de TP, TN, FP et FN est comme suit :

- TP: True Positive.
- TN : True Negative.
- FN : False négatif.
- FP : False Positive.

b) Précision Score :

Model précision score représente la capacité du modèle à prédire correctement les positifs parmi toutes les prédictions positives qu'il a faites. Le score de précision est une mesure utile du succès de la prédiction lorsque les classes sont très déséquilibrées. Mathématiquement, il représente le rapport du vrai positif à la somme du vrai positif et du faux positif

$$\text{Precision Score} = \frac{TP}{(FP+TP)} \quad \textbf{(3.2)}$$

Le même score peut être obtenu en utilisant la méthode precision_score de sklearn.metrics.

c) Recall Score

Model recall score représente la capacité du modèle à prédire correctement les positifs parmi les positifs réels. Ceci est différent de la précision qui mesure combien de prédictions faites par des modèles sont réellement positives sur toutes les prédictions positives faites. Le score de rappel est une mesure utile du succès de la prédiction lorsque les classes sont très déséquilibrées. Mathématiquement, il représente le rapport du vrai positif à la somme du vrai positif et du faux négatif.

$$\text{Recall Score} = \frac{TP}{(FN+TP)} \quad \textbf{(3.3)}$$

3.7.3. Résultats et discussion

Dans cette étape, la fonction classifier.predict() est utilisée pour prédire les valeurs de l'ensemble de test et les valeurs sont stockées dans la variable y_pred.

Les différentes expérimentations sont portées sur les trois datasets (CAD, PID, BCDS).

Pour évaluer l'efficacité de l'algorithme de classification sur les trois jeux de données (CAD, PID, BCDS), nous avons mené deux expériences : la première est sans sélection d'attributs et

la deuxième est avec une sélection d'attributs afin de les comparées et déterminer l'influence de la sélection d'attributs sur l'efficacité des modèles de classification.

3.7.3.1. Sans sélection d'attribut

Dans cette expérience, avant d'utiliser l'algorithme de classification, nous avons divisé les données de Dataset en deux parties : la première partie d'entraînement pour entraîner le modèle et la deuxième partie pour tester le modèle le modèle et évaluer ses performances. Nous avons divisé les données comme suit : 90 % des données utilisées pour apprentissage et 10% pour les tests. Tout d'abord, les quatre classificateurs de base, KNN, Naive Bayes,SVM et DecisionTree sont utilisés dans tout le jeu de données sans sélection d'attribut. Nous adoptons le taux de classification (Accuracy), la précision et rappel (Recall) comme des facteurs de comparaison entre ces algorithmes. Les résultats obtenus sont présenté dans le tableau 3.1.

	SVM			KNN			DT			NB		
	Acc	Pr	Re	Acc	Pr	Re	Acc	Pr	Re	Acc	Pr	Re
PID	99.48%	100%	100%	98.96%	99%	99%	100%	100%	100%	100%	100%	100%
BCDS	95.83%	100%	88.88%	90.21%	100%	74.07%	100%	100%	100%	100%	100%	100%
CAD	96.05%	100%	86.36%	92.11%	86.36%	86.36%	100%	100%	100%	97.37%	100%	90%

Tableau 3.2: Les résultats obtenus pour les différente Data sets sans sélection d'attributs

Nous pouvons observer d'après le Tableau 3.2 que les résultats atteignent une précision de classification de 100% pour les différentes bases de données comme suit :

- ✓ PID avec un classificateur DicisionTree et NaiveBayes.
- ✓ BCDS avec un classificateur SVM, DicisionTree et NaiveBayes.
- ✓ CAD avec un classificateur DicisionTree.

Les résultats atteignent une précision de classification de 99.48% pour les ensembles de données de PID avec un classificateur de SVM et 98.96% pour KNN. Les résultats pour CAD avec naive_bayes 97.37%, SVM 96.05% et KNN 92.11%. Dernière résultats pour BCDS

moins bons que les deux data sets avec le classificateur SVM 95.83% et 90.21% avec KNN.

Nous sélectionnons le modèle d'arbre de décision comme le modèle le plus idéal et qui est le meilleur pour notre ensemble de données en raison de son importante précision, rappel et taux de classification.

3.7.3.2. Avec sélection d'attributs

Dans cette expérience nous avons implémenté une technique de sélection d'attributs à l'aide de SelectKBest class de la bibliothèque Python scikit-learn. Nous utiliserons l'ensemble de données PID, CAD et BCDS pour sélectionner 3 des attributs ayant les meilleures caractéristiques à l'aide d'un test statistique chi carré. Nous pouvons également résumer les données pour la sortie selon notre choix. Les résultats sont représentés sur le tableau 3.3 :

	PID			BCDS			CAD		
	Acc	Pr	Re	Acc	P	R	Acc	P	R
SVM	98.7%	98.07%	100%	98.28%	100%	95.45%	93.55%	100%	81.81%
KNN	97.40%	98.03%	98.03%	91.38%	100%	77.27%	93.55%	98.03%	81.81%
DT	100%	100%	100%	100%	100%	100%	100%	100%	100%
NB	100%	100%	100%	100%	100%	100%	96.77%	100%	90%

Tableau 3.3. Les résultats obtenus avec une approche de sélection d'attributs

Après avoir les meilleurs attributs sélectionnés, les résultats en terme de taux de classification pour les différents classificateurs SVM, KNN, NB, DT sont observés. Nous avons remarqué que la sélection d'attributs améliore les résultats en les comparants avec ceux de meilleurs résultats obtenus sans sélection d'attribut pour les différentes Datasets.

Nous pouvons observer d'après le Tableau 3.3 que les résultats de classificateur DicisionTree n'ont pas changé, nous avons les mêmes taux de classification sans sélection d'attributs avec précision de 100%.

Les attributs sélectionnés du dataset BCDS ont obtenu de meilleur taux de classification avec SVM 98 .28% et KNN de 91 .38%. Pour le dataset CAD les expérimentations

obtienne une précision élevé avec KNN de 93.55% et avec les autres classificateurs ces résultats ont moin que celles obtenus la classification sans sélection d'attributs.

Donc, les résultats de classification sans sélection d'attributs de dataset PID sont mieux que les résultats avec sélection d'attributs.

Dans le tableau 3.4., nous récapitulons l'ensemble et le nombre d'attributs sélectionnés pour chacune des bases de données utilisés

Dataset	**Nombre total d'attributs**	**Nombre d'attributs sélectionnés**	**Attributs sélectionnés**
CAD	55	8	-Age -BP -FBS -TG -ESR -WBC -Region RWMA -Cath
PID	8	3	-plas -insu -class
BCDS	9	3	-tumor-size -Inv-node -Class

Tableau 3.4 : les meilleurs attributs sélectionné

3.8. Conclusion

Ce chapitre a pour objectif de présenter les étapes suivies durant l'implémentation d'une approche de sélection d'attributs au profit d'un diagnostic médical utilisant plusieurs algorithmes de classification. Pour la sélection d'attribut, nous avons utilisé le modèle selectKbest. Pour la classification, nous avons également appliqué des algorithmes d'apprentissage à savoir ; KNN, les Arbres de décision, SVM et naïve bayes. Le taux de classification (accuracy) précision et rappelle de chacun des algorithmes utilisés a été calculé afin de choisir le meilleur algorithme. D'après notre étude comparative, nous avons trouvé que le meilleur modèle est l'arbre de décision.

Conclusion Générale

Les maladies chronique sont le problème de santé le plus important concernant la population du monde. Les maladies chroniques menées à la morbidité et l'augmentation des taux de mortalité en niveau du monde. Une maladie chronique est une maladie de longue durée qui évolue lentement et qui s'aggrave souvent avec le temps et peut rarement être guérie. Elle fait partie des applications et problématiques rencontrées fréquemment dans le domaine médical.

L'exploration de données est le processus d'extraction d'informations cachées à partir d'un grand ensemble de données. Les principales techniques d'exploration de données telles que la clusterisation, la classification et l'analyse d'association, ont été utilisées pour prédire les maladies. Mais les techniques qui ont été introduites jusqu'à présent présentaient des inconvénients au niveau de la qualité du prétraitement à cause du volume des bases de données d'une part et d'une autre part la présence de beaucoup d'informations inutiles.

Dans ce mémoire, nous avons utilisé les techniques de sélection d'attributs comme une approche de réduction de la dimensionnalité des bases de données afin d'améliorer les techniques de la fouille de données et essentiellement la classification.

Dans la première partie du mémoire, nous avons présenté une étude de la sélection des attributs comme approche de réduction de la dimensionnalité des bases de données. Les différentes étapes du processus de sélection des attributs et les différentes stratégies de recherches et les différents critères d'évaluation qui peuvent être utilisées dans les méthodes de sélection des attributs sont aussi présentées. Dans la deuxième partie, nous avons implémenté l'approche « SelectKBest » qui sélectionne les meilleurs attributs des bases de données. Cet approche est basé sur le principe « filtrage » qui consiste à sélectionner un sous ensemble d'attributs et appel ensuite le classifieur afin d'évaluer la qualité du sous ensemble sélectionné.

Nous avons réalisé de nombreuses expérimentations afin d'évaluer l'approche SelectKBest. En utilisant trois jeux de données issues de l'UCI (BCDS, CAD, PID). Les résultats obtenus sont très encourageants, que ce soit au niveau des attributs sélectionnés ou du taux de classification obtenu par les différents classifieurs.

Durant les expérimentations, nous avons appliqué quatre algorithmes d'apprentissage automatique à savoir ; K plus proches voisins, arbres de décision, SVM et Naive Bayes. La précision de chacun des algorithmes utilisés a été calculée afin de choisir le meilleur

algorithme. Les résultats expérimentaux montrent l'importance de cette approche mais avec des résultats qui ce diffèrent d'une base de données à une autre.

En perspective, plusieurs voies de recherches peuvent être suivies pour une continuité de ce travail :

- Utilisation d'autres algorithmes de classification tels que les réseaux de neurones.
- Teste de cette approche sur d'autres maladies chronique.
- Appliquer l'approche sur des bases de données volumineuse pour valider mieux les cette approche.

Bibliographie

[1] Gagaoua Meriem (2012), Apprentissage et fouille de données par les algorithmes bio-inspirés : Application à la reconnaissance de caractères arabes manuscrits, thèse Magister Faculté des Sciences, Département d'informatique, Spécialité Ingénierie des Systèmes Informatiques , Université Ferhat Abbas-Sétif.

[2] Boulila, W. (2012). Extraction de connaissances spatio-temporelles incertaines pour la prédiction de changements en imagerie satellitale (Doctoral dissertation, Télécom Bretagne, Université de Rennes 1).

[3] Samuel K. (2011). Une contribution du datamining àƒÂ la segmentation du marché et au ciblage des offres àƒÂ l'aide de la statistique de Bayes. Mémoire de Magistère, UNiversité de Kinshasa .

[4] https://www.piloter.org/business-intelligence/datamining.htm. Consulté le 05/03/2022

[5] https://www.talend.com/fr/resources/data-mining-techniques/. Consulté le 15/03/2022

[6] Fady Hamoui. (2007), D´etection de fraudes et Extraction de Connaissances, Université´ Montpellier II

— Sciences et Techniques du Languedoc

[7] Melki Rima (2019) , Apprentissage des réseaux de neurones MLP par une méthode hybride à base d'une métaheuristique , Département de Génie Electrique ,Filière : Génie Electrique, Spécialité: Electronique Des Systèmes Embarquées, Université Larbi Ben M'hidi Oum-El-Bouaghi .

[8] http://www-igm.univ-mlv.fr/~dr/XPOSE2001/seguin/final/ReseauNeuro.html. Consulté le 16/03/2022

[9] Tellier I., Introduction à la fouille de textes, université de Paris 3 - Sorbonne Nouvelle.

[10] Koudri M., mémoire fin d'étude Ingénieur en Informatique, Université abou bakr belkaid. Tlemcen, Algérie.

[11] Fatoumata Y. et Amor A. (2021), Machine learning pour la maintenance prédictive ,Thèse de master Génie Mécanique, Spécialité Maintenance industrielle , Université Larbi Ben Mhidi d'Oum El-Bouaghi ,Algérie .

[12] Mouhamadou L.SA., Fodé C., Samba N.E, Yahya S. et ESSEGHIR M. A. (2014), Approche de sélection d'attributs pour la classification basée sur l'algorithme RFE-SVM,

Faculté des Sciences et Techniques, Université Cheikh Anta Diop de Dakar SENEGAL, Tunis TUNISIE.

[13] Mouhoub B. (2021), Apprentissage statistique pour l'extraction des relations à partir de textes, Thèse de Doctorat en science : informatique, Université Ferhat Abbas - Sétif -1, Algérie.

[14] Nicole C. (2018), Contributions à la sélection des attributs de signaux non stationnaires pour la classification, Thèse de Doctoral Spécialité : Optimisation Et Sureté Des Systèmes Université de Technologie de Troyes, Français.

[15] Huerta, E. B. (2008). Logique floue et algorithmes génétiques pour le pré-traitement de données de biopuces et la sélection de gènes (Doctoral dissertation, Université d'Angers).

[16] : Menghour K. (2015), Approches Bio-inspirées pour la Sélection d'Attributs, thèse de Doctoral en science : informatique , Spécialité Sciences et Technologies de l'Information et de la Communication, Université Badji Mokhtar-Annaba , Algérie .

[17] https://datascientest.com/jupyter-notebook. Consulté le 16/04/2022

[18] https://khayyam.developpez.com/articles/machine-learning/scikit-learn/. Consulté le 10/03/2022

[19] https://archive.ics.uci.edu/ml/about.html. Consulté le 05/04/2022

[20] https://sante.lefigaro.fr/actualite/2013/07/23/21022-maladies-cardiaques-1re-cause-deces-dans-monde

[21] American Cancer Society. Breast Cancer. 2015:

http://www.cancer.org/cancer/breastcancer/index.