MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS MOSTAGANEM

Faculté des Sciences Exactes & de l'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : Ingénierie des Systèmes d'Information

Intitulé du sujet

Evaluation des méthodes de Sélection de Variables en Apprentissage supervisé

Présenté par :

Abdelkader ABDELMALEK
Walid HEBBAR

Encadré par:

Mme Tamazouzt AIT SAADI

Année Universitaire 2013/ 2014

D É D I C A C E S

Nous dédions ce modeste travail à ceux qui ont fait de nous les hommes que nous sommes aujourd'hui. Nos très chers parents, que dieu les récompense et les garde.

Nos frères et sœurs.

Tous nos amis.

Nos collègues de promotion.

A tous ceux qui nous connaissent de près ou de loin.

Merci d'être toujours là pour nous.

R E M E R C I E M E N T S

«(Et lorsque votre Seigneur proclama : "Si vous êtes reconnaissants, très certainement J'augmenterai [Mes bienfaits] pour vous)»

[Coran S14.V7]

Avant tout, nous remercions Dieu le clément et mésiricordieux qui nous a donné le courage et la volonté pour réaliser ce modeste travail.

«(CELUI QUI NE REMERCIE PAS LES GENS, NE REMERCIE PAS ALLAH.)»

[Authentique Hadith]

Que Madame AIT SADI Tamazouzt, maitre assistante à l'université Abdelhamid ibn Badis Mostaganem, Faculté des Sciences Exactes & de l'Informatique, trouve ici le témoignage de notre profonde reconnaissance et gratitude. Son encadrement ,ses conseils, ses encouragements, , sa sensibilisation, et surtout ses critiques ont largement contribué à l'accomplissement de nos travaux. Nous la remercions infiniment de nous avoir toujours poussé vers l'avant.

Nous tenons également à remercier « TOUS » - Dames et Messieurs- nos enseignants qui nous ont formé en master2, option Ingénierie des Systèmes d'Informations, pour la qualité de leurs enseignements.

Nos remerciements vont également aux membres du jury d'avoir accepté d'évaluer notre travail.

Sans oublier de remercier nos amis et nos collègues (de l'université ou dans le monde Virtuel « internet ») qui, tous d'une manière différente, ont contribué à ce que nous puissions aboutir à la réalisation de ce mémoire.

Enfin, merci à nos familles pour le soutien et l'encouragement qu'ils nous ont apporté tout au long de notre travail.

R E S U M E

La résolution de la plupart des problèmes, dans de nombreux domaines de la vie courante, se base sur le traitement de données extraites à partir des données acquises dans le monde réel, et structurées sous forme de vecteurs. La qualité du système de traitement dépend directement du bon choix du contenu de ces vecteurs. Mais dans de nombreux cas, la résolution du problème devient difficile, voire presque impossible à cause de la dimension trop importante de ces vecteurs.

Force est de constater, que aussi que le problème de la sélection de variables en classification se pose généralement lorsque le nombre de variables pouvant être utilisé pour expliquer la classe d'un individu, est très élevé.

Dans ce cadre, nous proposons dans ce mémoire l'étude d'un certain nombre de différentes méthodes de sélection de variables existantes. Ces méthodes présentent un certains nombre de caractéristiques, tel que :

- la dépendance des variables pertinentes sélectionnées par rapport au classificateur utilisé;
- la redondance entre les variables sélectionnées;
- les interactions entre les différentes variables;
- la faiblesse au niveau de leur complexité qui s'avère être parfois très élevée.

Notre contribution, et dans le but de connaitre au mieux ces difficultés, consiste à :

1-Analyser et comparer certaines méthodes de sélection de variables appliqués dans différents domaines - la bioinformatique et autres- , de types Filter, Wrapper et Embedded.

2- Analyser les forces et les faiblesses de ces méthodes au vu de la dimensionnalité des données disponibles. Les méthodes détaillées sont basées sur la sélection de classificateurs simples associés à chacune des variables.

3- Trouver une bonne combinaison entre les méthodes de sélection et les classificateurs SVM et Naïves Bayésien utilisés dans nos expérimentations capable de sélectionner un nombre réduit de caractéristiques tout en conservant des taux de classification très satisfaisants.

Nos expérimentations ont montré que les méthodes et approches adoptées ont la capacité de sélectionner un nombre réduit de variables tout en conservant des taux de classification très satisfaisant.

Mots-clés : Sélection de variables, Classification supervisée, méthodes de sélection, Support Vector Machine, Naïves Bayésien.

A B S T R A C T

Solving most of problems in many areas of life are based on the processing of extraction data from the real world, and structured as vectors. The quality of treatment system directly depends on the correct choice of the content of these vectors. But in many cases, solving the problem becomes difficult or almost impossible due to the excessive size of these vectors. It is clear that also the problem of feature selection in classification, generally arises when the number of features that can be used to explain the class of an instance, is very high. In this context, we propose in our document the study of a number of different features selection methods.

These methods have a number of characteristics, such as:

- Dependence of relevant features selected with the classifier used;
- Redundancy between the selected features;
- Interactions between different features;
- Weakness in their complexity which sometimes turns out to be very high.

Our contribution, and in order to know the most of these difficulties is to:

1- Analyze and compare some feature selection methods applied in different areas - bioinformatics, and other, like Filter, Wrapper and Embedded approach.

2 - Analyze the strengths and weaknesses of these methods in view of the dimensionality of available data. Detailed methods are based on the selection of simple classifiers associated with each variable.

3 - Look for good combination of selection methods and SVM and Bayesian Naives classifiers used in our experiments, capable of selecting a small number of features while maintaining very satisfactory classification rate.

Our experiments have shown that the adopted methods and approaches have the ability to select a reduced number of features while preserving very satisfactory classification rates.

Keywords: feature selection, supervised classification, selection methods, Support Vector Machine, Naives Bayesian.

Tables des Matières

Liste des figures :

Liste des Tableaux :

Glossaire:

AFD Analyse linéaire Discriminante de Fisher

AG Algorithme Génétique

BDD Base De Données

BDS Bi-Directional Selection

BS Backward Selection

CMIM Conditional Mutual Information Maximization

FCBF Faste Correlation Based Filter

FN Faux Négatifs

FP Faux Positifs

FS Forward Selection

KNN K Nearly Neighbors

LVF Las Vegas Filter

LVW Las Vegas Wrapper

MIFS Mutual Information Feature Selection

mRMR Max-Relevance, Min-Redundancy

NB Naïve Bayésien

PIMA Diabetes Pedigree Function

PMC Perceptron Multi Couche

SAC Sélection Adaptative de Caractéristiques

SFS Sequential Forward Selection

SBS Sequential Backward selection

SE Sensibilité

SP Spécificité

SV Sélection de Variables

SVM Support Vector Machines

TC Taux de Classification

UCI Université de Californie Irvine

VN Vrais Négatifs

VP Vrais Positifs

Introduction Générale :

Il est actuellement possible d'analyser de grandes quantités de données de dimension élevée grâce aux performances accrues des ordinateurs. Ces données sont traitées, disposant dans la plupart d'un nombre important de variables et/ou un nombre importants d'instances. Il s'avère que les méthodes classiques d'analyse, d'apprentissage ou de fouille de données peuvent se révéler inefficaces ou peuvent conduire à des résultats inexacts. De ce fait, il est nécessaire de réduire la dimension des données en sélectionnant les variables les plus intéressantes pour le problème étudié **[6, 9]**.

La sélection de variables consiste à choisir parmi l'ensemble global de variables, un sous-ensemble de variables pertinentes pour le problème étudié. Cette problématique peut concerner différentes tâches de fouille de données, elle regroupe *des méthodes* permettant de sélectionner un sous-ensemble de variables parmi un ensemble de départ, en utilisant divers critères et différentes techniques.

Cette problématique peut concerner différentes tâches de fouille de données, mais dans notre projet, nous traitons uniquement la sélection de variables réalisée en *classification supervisée* qui consiste à déterminer, sur une base d'un nombre fini d'individus, la relation entre un ensemble de variables explicative et une variable à expliquer qui s'appelle la classe.

La sélection de variables présente plusieurs avantages liés à la réduction de la quantité de données (moins de variables). D'une part, cette réduction rend beaucoup plus facile de gérer les données et d'autre part, elle aide à mieux comprendre les résultats fournis par un système basé sur ces variables. A titre d'exemple, pour un problème de classification, ce processus de sélection ne réduit pas seulement le temps d'apprentissage mais il aide aussi à mieux comprendre les résultats fournis par le classificateur et à améliorer parfois la précision de la classification, en favorisant les variables les moins bruitées.

Les méthodes de sélection de variables sont classées généralement en deux groupes : les méthodes *"filter"* et les méthodes *"wrapper"*. La première approche (méthodes de filtrage) utilise des mesures statistiques calculées sur les variables afin de filtrer les variables peu informatives. Cette étape est généralement réalisée avant d'appliquer tout algorithme de classification. Ces méthodes de filtrage présentent des avantages au niveau de leur efficacité calculatoire et de leur robustesse face au sur-apprentissage. Mais ne tiennent pas compte des choix faits pour la méthode de classification qui suit la sélection.

La seconde approche (méthodes enveloppantes ou *"wrapper"* est plus coûteuse en temps de calcul, mais en contrepartie, elle est souvent plus précise. Un algorithme de type *"wrapper"* explore l'espace des sous-ensembles de variables afin de trouver un sous-ensemble optimal pour un algorithme d'induction bien défini. Les sous-ensembles de variables sélectionnés par cette méthode sont bien adaptés à l'algorithme de classification utilisé, mais ils ne restent pas forcément valides si on change le classificateur. La complexité de l'algorithme d'apprentissage rend les méthodes *"wrapper"* très coûteuses en temps de calcul. Les méthodes "*wrapper"* sont généralement considérées comme étant meilleures que celles de filtrage et de plus, elles sont capables de sélectionner des sous-ensembles de variables de plus petite taille, néanmoins aussi performants pour le classificateur utilisé. Les méthodes *"wrapper"* présentent des limitations, d'une part au niveau de la complexité et du temps de calcul nécessaire pour la sélection et d'autre part par la dépendance des variables pertinentes sélectionnées au classificateur utilisé.

Dans notre travail nous allons :

- Présenter quelques méthodes de sélection de variables;
- Etudier et comparer les performances de chaque méthode de sélection de variables (taux de classification, rapidité d'exécution, nombre de variables sélectionnées) et son comportement par rapport au classifieur utilisé;
- Analyser la possibilité de prendre en compte les interactions entre les variables sélectionnées (variables communes) d'un point de vue performance de classification.

Pour ce faire, dans la partie expérimentation, nous allons expérimenter via un outil logiciel développée pour la circonstance plusieurs bases de données issues du domaine de l'expérimentation et disponible en accès public - bases de données UCI (Iris, Leukemia, PIMA Diabetes, Cancer....)- de comparer différents méthodes de classification et nous utiliserons les algorithmes Naïves Bayésien et SVM (Support à Vecteurs de Marges).

Le plan de mémoire est décomposé comme suit :

- Dans ***le premier Chapitre****, nous* présentons en détail les techniques de sélection de variables ainsi que leurs avantages et leurs limitations. Une revue de quelques méthodes est effectuée ainsi qu'un état de l'art des différents travaux dans le même domaine. Nous concluons le chapitre par la présentation de notre contribution dans le cadre de ce projet;

-Dans ***le deuxième Chapitre,*** nous abordons l'état de l'art des algorithmes de classification usuels, leurs limitations et la manière avec laquelle ils abordent chacun le problème de la classification. Nous détaillons la classification supervisée, objet de notre étude et expérimentation. Dans la deuxième partie de ce chapitre, nous présentons l'approche basée sur l'ensemble de classificateurs.

-Nous consacrons ***le troisième chapitre*** , à la partie expérimentation, nous présentons les techniques de sélection de variables expérimentées, puis nous décrivons les implémentations et leurs résultats obtenus avec une présentation synthétique des méthodes de sélection utilisées et associées aux algorithmes d'apprentissage SVM et Naïves Bayésien. Nous terminons ce chapitre par une comparaison entre nos résultats et ceux de la littérature. Les méthodes de sélection de variables ont été éprouvés sur plusieurs bases de données issus du domaine public, "UCI repository".

-Nous terminons par ***une conclusion générale*** et quelques perspectives que nous souhaitons pour ce projet.

CHAPITRE 1

Sélection de variables

1.1 Introduction

On assiste actuellement à un accroissement important de la taille des bases de données, ce qui pose un défi sans précédent pour la fouille de données et à l'extraction des connaissances à partir des données. Les bases de données en plus de l'accroissement de leur taille, nous constatons que de nouveaux types de données deviennent très répandus. Les chercheurs se sont rendu compte que la sélection des variables est un élément essentiel pour que la fouille de données atteigne ses objectifs**.**

Un nombre élevé de variables peut en effet s'avérer pénalisant pour un traitement pertinent et efficace des données, d'une part par les problèmes algorithmiques que cela peut entraîner (liés au coût de calcul et à la capacité de stockage nécessaire), et d'autre part certaines peuvent être non-pertinentes, inutiles et/ou redondantes perturbant ainsi le bon traitement des données. Or, il est souvent difficile voire impossible de distinguer les variables pertinentes des variables non-pertinentes.

La sélection de variables constitue une solution à ce problème. Ce processus vise en effet à déterminer un sous ensemble optimal (au sens d'un critère donné) de variables et donc à réduire du nombre de variables par élimination des variables non pertinentes ou redondantes **[1].**

1.2 Définitions

1.2 .1 La sélection

La sélection est un processus (opération volontaire et méthodique, phénomène inconscient ou automatique) par lequel certains éléments (personnes ou choses) sont choisis en fonction de caractéristiques déterminées, éventuellement impliquées par une certaine fin ou objectif **[2].**

La sélection est une action qui permet de choisir des personnes ou des objets qui conviennent le mieux **[3].**

1.2.2 La variable

Une variable est sujet à des variations, elle peut changer au cours d'une durée, selon les circonstances **[2].** Elle peut être différente selon les cas **[3].**

1.2.3 La sélection de variables

La sélection de variables consiste à sélectionner parmi un ensemble de variables de grande taille un sous-ensemble de variables intéressantes et pertinentes pour un problème donné.

Les principales motivations de la sélection de variables sont les suivantes **[5]**:

- L'utilisation d'un sous-ensemble plus petit de variables permet d'améliorer la classification si l'on élimine les variables qui sont source de bruit. Cela permet aussi une meilleure compréhension des phénomènes étudiés ;
- Des sous-ensembles de variables plus petits permettent une meilleure généralisation des données en évitant le sur-apprentissage;
- Le choix d'un sous ensemble de variables pertinentes permet de réduire le temps d'apprentissage et d'exécution et par conséquent l'apprentissage est moins coûteux.

1.3 Types de variables

1.3.1 Pertinence de variables

Une variable pertinente est une variable telle que sa suppression entraîne une détérioration des performances du système d'apprentissage **[4]**.

Kohavi et John définissent les variables pertinentes comme celles dont les valeurs varient systématiquement avec les valeurs de la classe **[5]**.

Selon John et al **[5, 6**], une variable peut être très pertinente, peu pertinente et non pertinente.

- **Très pertinente** : Une variable $\boldsymbol{f_i}$ est dite très pertinente, si son absence entraîne une détérioration significative de la performance du système de classification utilisé.
- **Peu pertinente** : Une variable $\boldsymbol{f_i}$ est dite peu pertinente si elle n'est pas "très pertinente" et s'il existe un sous-ensemble V tel que la performance de $\mathbf{V} \cup \{\boldsymbol{f_i}\}$ soit significativement meilleure que la performance de **V**.
- **Non pertinente** : Les variables qui ne sont ni "peu pertinentes" ni "très pertinentes" sont des variables non pertinentes. Ces variables seront en général supprimées de l'ensemble de variable de départ.

1.3.2 Redondance de variables

La notion de la redondance de variables est généralement exprimée en termes de corrélation entre variables. On dira que deux variables sont redondantes (entre elles) si leurs valeurs sont complètement corrélées. Cette définition ne se généralise pas directement pour un sous-ensemble de variables. Koller et Sahami **[7]** ont donné une définition formelle de la redondance. Cette définition a permis par la suite de concevoir une approche pour identifier et éliminer les variables redondantes.

1.3.3 Variables corrélées

Des variables sont considérées comme corrélées si leur combinaison est capable de déterminer les classes induites par la variable endogène **[19]**.

1.3.4 Variables bruitées

Ce sont des variables qui ne permettent pas de distinguer des individus appartenant à deux classes différentes **[19]**.

1.4 Sélection de variables

La sélection de variables est généralement définie comme un processus de recherche permettant de trouver un sous-ensemble pertinent de variables parmi celles de l'ensemble de départ. La notion de pertinence d'un sous- ensemble de variables dépend toujours des objectifs et des critères du système. En général, le problème de sélection de variables peut être défini comme suit : Soit $\boldsymbol{F} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_3}, \dots \dots \boldsymbol{f_n}\}$ un ensemble de variables de taille $\boldsymbol{N}$ où $\boldsymbol{N}$ représente le nombre total de variables étudiées. Soit $\boldsymbol{Ev}$ une fonction qui permet d'évaluer un sous-ensemble de variables. Nous supposons que la plus grande valeur de $\boldsymbol{Ev}$ soit obtenue pour le meilleur sous-ensemble de variables. L'objectif de la sélection est de trouver un sous-ensemble $\boldsymbol{F'(F' \subseteq F)}$ de taille $\boldsymbol{N'\ (N' \leq N)}$ tel que : $\boldsymbol{Ev}(F') = \max \boldsymbol{Ev}(Z)$; Z⊆F

Où $|Z| = N'$, N' peut être un nombre prédéfini par l'utilisateur ou contrôlé par une des méthodes de génération du sous-ensemble **[8].**

La sélection de variables est un processus qui permet de « sélectionner » un sous-ensemble de variables considérées par le processus comme pertinentes. Les données d'entrée du processus sont constituées par l'ensemble initial de variables qui forment l'espace de représentation et l'ensemble des données d'apprentissage du problème étudié.

Le processus de sélection de variables se décompose de la manière suivante, figure 1 **[9]** :

- A partir de l'ensemble initial des variables, le processus de sélection détermine un sous-ensemble de variables qu'il considère comme les plus pertinentes ;
- Le sous-ensemble est ensuite soumis à une procédure d'évaluation. Cette dernière permet d'évaluer les performances et la pertinence du sous-ensemble ;
- En fonction du résultat de la procédure d'évaluation, un critère d'arrêt du processus détermine si le sous-ensemble de variables peut être soumis à la phase d'apprentissage. Si tel est le cas, le processus de sélection s'arrête, sinon, un autre sous-ensemble de variables est généré.

Les principaux enjeux et conséquences de la sélection de variables sont :

- La sélection de variables permet de déterminer les variables considérées comme pertinentes ;
- La sélection de variables permet de supprimer le bruit généré par certaines variables ;
- La sélection de variables permet de supprimer les variables redondantes;
- La sélection de variables permet de réduire la taille de l'espace de représentation. Le coût de calcul de la phase d'apprentissage est ainsi réduit.

1.4.1 Processus général de la sélection de variables

Les algorithmes de sélection de variables sont caractérisés par les éléments clés suivant **[1]**:

- Une procédure de ***génération de sous-ensembles*** candidats qui détermine l'exploration de l'espace de recherche;
- Une fonction ***d'évaluation*** donnant la qualité des sous-ensembles candidats ;
- Une condition ***d'arrêt***;
- Un processus de ***validation*** pour vérifier si l'objectif souhaité est atteint.

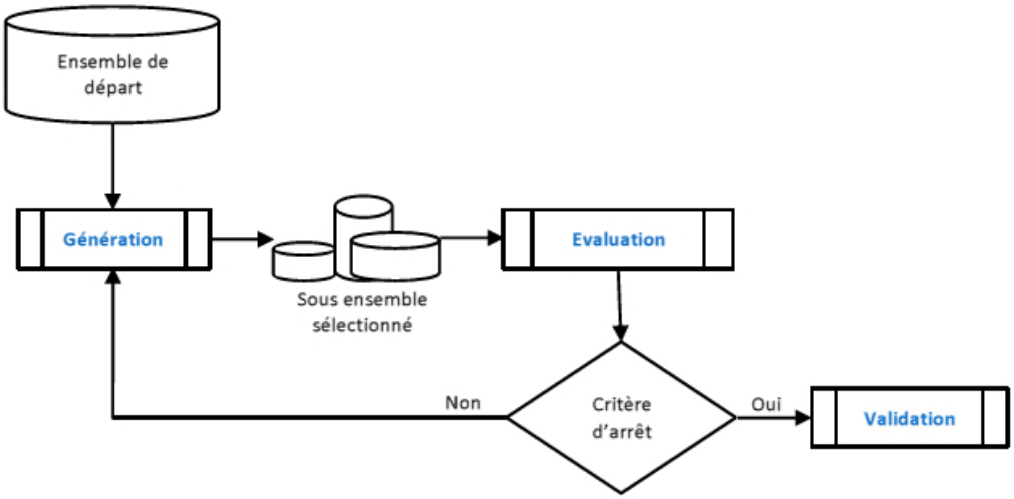


Figure 1*Processus de sélection de variables* **[1]**

1.4.1.1 Génération des sous-ensembles de variables

Dans le cadre de la sélection de variables, la procédure de génération désigne la façon de générer l'ensemble de variables candidat. Siedlecki et Sklansky **[11]** parlent aussi de procédure de recherche. Le principe général consiste à générer successivement des sous-ensembles de variables à évaluer. La procédure de génération des sous-ensembles de variables est caractérisée par une ***stratégie de recherche.***

La stratégie de recherche dépend de la taille de l'espace de recherche. Pour un ensemble de m variables, le nombre de sous-ensembles de variables candidats est 2^m-1. Il se trouve que même pour un nombre de variables raisonnable, le nombre de sous-ensembles à étudier est considérable. Pour affronter ce problème de taille de l'espace de recherche, trois stratégies de recherche sont envisageables : la génération exhaustive, la génération heuristique et la génération aléatoire.

1.4.1.1.1 Génération exhaustive

Une recherche exhaustive sur tous les sous-ensembles de variables est effectuée afin de sélectionner le "meilleur" sous-ensemble de variables. Cette stratégie de recherche garantit de trouver le sous-ensemble optimal. Le problème majeur de cette approche est que le nombre de combinaisons croît exponentiellement en fonction du nombre de variables **[1]**.

1.4.1.1.2 Génération heuristique

Les algorithmes qui utilisent cette approche sont généralement des algorithmes itératifs dont chaque itération permet de sélectionner ou de rejeter une ou plusieurs caractéristiques. Les avantages de ces algorithmes sont leur simplicité et leur rapidité. En revanche, ils ne permettent pas de parcourir totalement l'espace de recherche. Les trois sous catégories les plus utilisées sont **[1]**:

1- Forward: approche ascendante, son principe est de commencer avec un ensemble de caractéristiques vide et à chaque itération une ou plusieurs caractéristiques seront ajoutées.

2- Backward : cette approche procède d'une façon inverse à "Forward ". L'ensemble de départ représente l'ensemble total des caractéristiques et à chaque itération, une ou plusieurs caractéristiques sont supprimées. C'est une approche descendante.

3- Stepwise : cette approche est un mélange des deux précédentes, elle consiste à ajouter ou supprimer des caractéristiques au sous-ensemble courant.

1.4.1.1.3 Génération aléatoire

La procédure de recherche aléatoire consiste à générer aléatoirement un nombre fini de sous- ensembles de caractéristiques afin de sélectionner le meilleur**[1]**.

1.4.2 Evaluation des sous-ensembles

L'évaluation d'un sous-ensemble est traitée de façons très diverses tout en précisant le type d'approche utilisé et la fonction d'évaluation utilisée.

Nous distinguons trois catégories de techniques de sélection de variables **[12]**:

- Approche filtre (filter);
- Approche enveloppe (wrapper);
- Approche intégrée (embedded).

1.4.2.1 Approche filtre (Filter)

Le filtrage est utilisé en phase de prétraitement, ce qui permet de réduire à la fois la dimension des entrées et de se prémunir dans certains cas du phénomène de sur-apprentissage.

La simplicité et l'efficacité de l'approche par filtre est souvent mise en avant.

Certains auteurs ont tenté d'éliminer les variables inutiles, le procédé se déroule en deux phases : la première, additive qui entraîne des PMC (Perceptron Multi-Couches) candidats avec un nombre croissant de neurones dans les couches cachées, puis les candidats obtenus, la seconde phase concerne la sélection par des tests de Fisher**[14]**. Le processus s'arrête dès l'obtention du plus petit réseau dont les variables et les neurones cachés ont une contribution significative pour le problème de régression-classification **[15]**.

Les méthodes par filtre ne reposent généralement pas sur des méthodes assez complexes, ce sont souvent des méthodes statistiques. Dans **[16]**, les auteurs proposent pour accroître les performances de plusieurs classifieurs, la sélection, via une heuristique de sélection par corrélation. L'heuristique tient compte de l'utilité individuelle de chaque variable et de sa corrélation avec les autres variables. Dans le même ordre d'idée, d'autres auteurs proposent aussi une heuristique nommée FCBF (fast correlation-based filter) qui introduit la notion de corrélation de prédominance **[17]**. Celle-ci a l'avantage d'avoir une complexité quasi linéaire au lieu d'une complexité quadratique.

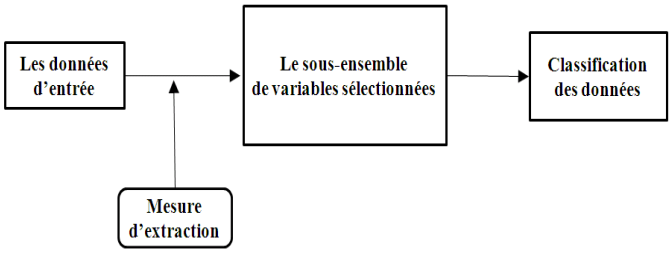


Figure 2 ***Le principe général d'une méthode de sélection de type Filter*** **[18]**

1.4.2.2 Approche enveloppe (Wrapper)

Le principe de base de ces méthodes est le suivant : nous considérons déjà acquis le classifieur,

il est utilisé pour qualifier l'utilité d'un groupe de variables (Figure 3). La difficulté de la méthode provient de la génération de ces groupes qui doit être renouvelée pour obtenir des ensembles plus performants à chaque itération. Le choix de la fonction objective et du critère d'arrêt sont eux aussi des problèmes difficiles à résoudre car ils sont la clé pour l'obtention d'une solution satisfaisante générée en un minimum d'itérations.

Il serait toujours envisageable d'énumérer toutes les combinaisons de sous-ensembles et de les tester à travers le classifieur dans le but de disposer d'une méthode exacte. En pratique, le problème est de classe NP et n'est donc pas réalisable. On utilise généralement des stratégies pour réduire le nombre de sous-ensembles à générer par des techniques de type séparation et évaluation, recuit simulé, ou méthodes basées sur des algorithmes génétiques **[5]**. Ces derniers sont les plus puissants et les plus utilisés depuis quelques années. Si le fait de considérer les classifieurs comme une boite noire permettant de conserver une sorte d'universalité du résultat obtenu. Le nombre de combinaisons à tester est souvent très important et ne donne en général pas de meilleurs résultats que ceux des méthodes Intégrées.

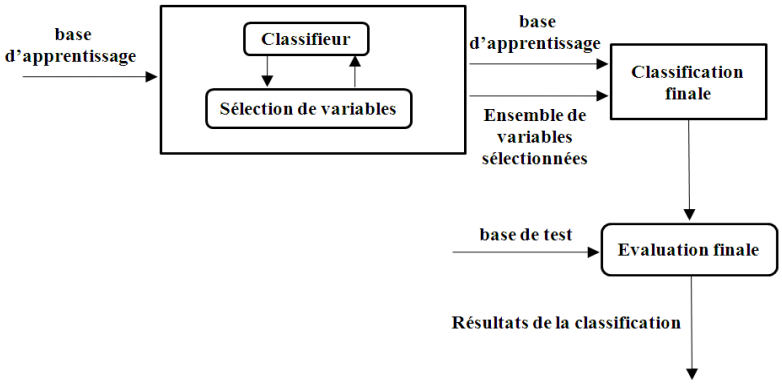


Figure 3*Le principe général d'une méthode de sélection de type wrapper* **[18]**

1.4.2.3 Approche intégrée (Embedded)

Ce type de méthodes choisit le groupe de variables durant l'apprentissage du classifieur. La recherche du meilleur sous-ensemble est guidée par l'apprentissage avec par exemple la mise à jour de la fonction objective (Figure 4). Des techniques déjà évoquées **[22]** peuvent être utilisées pour supprimer les variables non pertinentes. Elles font partie intégrante des méthodes intégrées.

Le problème des méthodes Intégrées est lié au fait qu'il faut adopter une stratégie en adéquation avec le type du classifieur, Optimal Brain Damage **[22]** ne s'applique qu'aux

Perceptrons multicouches.

Les méthodes connexionnistes représentent une grande majorité parmi les méthodes de type Intégrées. En général, elles se basent sur des critères heuristiques permettant d'estimer l'importance d'une ou de plusieurs variables sur la performance globale du système.

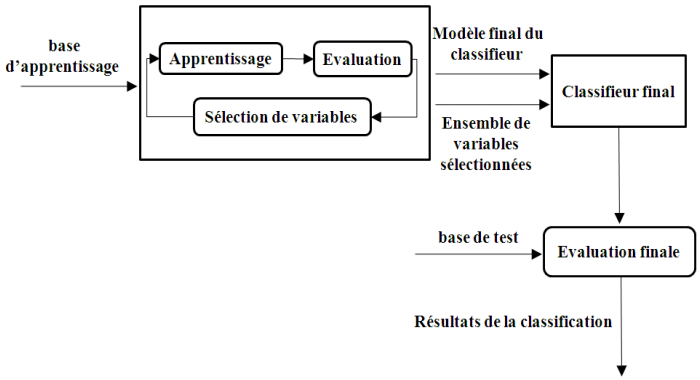


Figure 4 Le principe général d'une méthode de sélection de type Embedded [18]

1.4.3 Fonction d'évaluation

La fonction d'évaluation est utilisée pour mesurer la capacité d'une variable, ou d'un ensemble de variables, à discriminer les classes de la partition impliquée par la variable endogène. L'optimalité d'un sous-ensemble est relative à la fonction d'évaluation utilisée. Dash et Liu **[10]** considèrent que ces fonctions peuvent être regroupées en cinq catégories qui sont les suivantes : les mesures de divergence, les mesures d'information, les mesures de dépendance, les mesures de consistance et les mesures de précision.

- L'information : fonction quantifiant l'information apportée par une variable sur la variable à prédire. La variable ayant le gain d'information le plus élevé, est préférée aux autres variables.
- La distance : fonction s'intéressant au pouvoir discriminant d'une variable. Elle permet d'évaluer la séparabilité des classes en se basant sur les distributions de probabilités des classes. Une variable est préférée à une autre si elle induit une plus grande séparabilité.
- La dépendance : fonction mesurant la corrélation ou l'association. Elle permet de calculer le degré avec lequel une variable exogène est associée à une variable endogène.
- La consistance : fonction liée au biais des variables minimum. Elle permet de

rechercher le plus petit ensemble de variables qui satisfait un pourcentage d'inconsistance minimum défini par l'utilisateur. (Deux objets sont dits inconsistants si leurs modalités sont identiques et s'ils appartiennent à deux classes différentes.) Ces mesures peuvent permettre de détecter les variables redondantes.

- La précision : Elle utilise le classifieur comme fonction d'évaluation. Le classifieur choisit parmi tous les sous-ensembles de variables, celui qui est à l'origine de la meilleure précision prédictive.

La mesure de dépendance est toujours considérée comme une mesure d'information ou de distance **[24]**. Si l'on écarte la mesure de l'erreur de classification qui est un critère d'évaluation de l'approche wrapper, les mesures utilisées comme critères d'évaluation peuvent être répartis en trois catégories: mesure de consistance, mesure de distance et mesure de l'information. Il s'agit de mesures intrinsèques aux variables candidates, elles sont indépendantes de la phase d'apprentissage et sont très utilisées comme critère d'évaluation pour l'approche filtre.

1.4.4 Critère d'arrêt

Le critère d'arrêt permet à la procédure de sélection de variables de s'arrêter. En effet, la plupart des fonctions d'évaluations rencontrées dans la littérature sont monotones **[8]**.
Le critère d'arrêt peut être lié à la stratégie de recherche ou bien à la mesure d'évaluation. Dans le premier cas, le critère d'arrêt est la taille prédéfinie du sous-ensemble à sélectionner ou un nombre fixe d'itérations de l'algorithme de sélection de variables. Dans le deuxième cas, un critère d'arrêt lié à la mesure d'évaluation est La différence de qualité entre deux ensembles non significative (l'ajout ou la suppression d'une variable n'améliore pas la qualité du sous-ensemble) ou un seuil pour la fonction d'évaluation à atteindre. Si la distribution empirique de la mesure d'évaluation est connue, un bon critère d'arrêt est alors l'invraisemblance de la valeur de l'évaluation. Cette invraisemblance est mesurée grâce à un test statistique **[10].**

1.5 Revue de quelques méthodes de sélection

1.5.1 SFS, SBS et BDS

SFS (Sequential Forward Selection) ou (sélection séquentielle croissante) est la première méthode proposée pour la sélection de caractéristiques. Cette méthode a été proposée en 1963 par Marill et Green **[25]**. Une approche heuristique de recherche est utilisée dans cette

méthode, en commençant par un ensemble vide de variables. A chaque itération, la meilleure caractéristique parmi celles qui restent sera sélectionnée, supprimée de l'ensemble de départ et ajoutée au sous-ensemble des variables sélectionnées (Algorithme 2.1). Le processus de sélection continue jusqu'à un critère d'arrêt. En 1971, Whitney **[26]** a proposé une méthode similaire au SFS appelée SBS (Sequential Backward Selection) ou (sélection séquentielle arrière). Cette méthode commence par l'ensemble de toutes les variables et à chaque itération, la caractéristique la plus mauvaise sera supprimée (Algorithme 2.2).

Bien que les deux méthodes SFS et SBS semblent similaires, Aha et Bankert **[27]** ont montré que la méthode SBS est plus performante parce qu'elle prend en considération l'interaction d'une caractéristique avec un ensemble de caractéristiques plus large. A l'inverse de SFS qui ne prend en considération que l'interaction de cette variable avec le sous-ensemble déjà sélectionné. Par ailleurs, l'évaluation des sous-ensembles de grande taille avec la méthode SBS pose un problème au niveau du temps de calcul.

Algo 1.1 Algorithme *SFS*	**Algo 1.2** Algorithme *SBS*
Entrées: $\boldsymbol{F} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \dots \dots \boldsymbol{f_N}\}$ M : taille de l'ensemble final **Sorties:** $E = \{\boldsymbol{f_{s1}}, \boldsymbol{f_{s2}}, \dots \dots \boldsymbol{f_{sM}}\}$ $E = \emptyset$ **Pour** i = 1 à M **Faire** **Pour** $j = 1$ *à* $\lvert F \rvert$ **Faire** Évaluer $f_j \cup E$ **Fin Pour** $fmax$= meilleure f_j $E = E \cup fmax, F = F \backslash fmax$ **Fin Pour** **Retourner** E	**Entrées:** $\boldsymbol{F} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \dots \dots \boldsymbol{f_N}\}$ M : taille de l'ensemble final **Sorties:** $E = \{\boldsymbol{f_{s1}}, \boldsymbol{f_{s2}}, \dots \dots \boldsymbol{f_{sM}}\}$ $E = F$ **Pour** i = 1 à N-M **Faire** **Pour** $j = 1$ *à* $\lvert E \rvert$ **Faire** Évaluer $E \setminus f_j$ **Fin Pour** f_{jmin}= la plus mauvaise f_j $E = E \setminus f_{jmin}$ **Fin Pour** **Retourner** E

En 1978, des généralisations des méthodes SBS et SFS appelées GSFS et GSBS, sont proposées par Kittler. Dans ces méthodes, l'auteur propose d'inclure (ou d'exclure) un sous ensemble de variables à chaque itération. Ces méthodes ont montré une meilleure performance par rapport aux méthodes initiales, mais elles conservent les mêmes problèmes que les méthodes de base **[28]**.

Deux autres méthodes de la famille (FS, BS) qui limitent les inconvénients des méthodes décrites plus haut, appelées SFFS (Sequential Floating Forward Selection) et SFBS (Sequential Floating Backward Selection) ont été proposées en 1994 par Pudil et al **[29]**. Ces

méthodes consistent à utiliser ***l*** fois l'algorithme SFS de manière à ajouter ***l*** variables, puis à utiliser **r** fois l'algorithme SBS afin d'en supprimer **r**. Ces étapes sont alors réitérées jusqu'à l'obtention du critère d'arrêt. La dimension du sous-ensemble à chaque étape sera alors dépendante des valeurs de ***l***et **r**. Les valeurs optimales de ces paramètres ne pouvant pas être déterminées théoriquement, les auteurs proposent de les laisser flottants au cours du processus de sélection afin de se rapprocher au maximum de la solution optimale.

BDS (Bi-Birectional Selection) ou recherche bidirectionnelle consiste est une mise en œuvre parallèle de SFS et SBS **[50]**:

- SFS est exécuté à partir de l'ensemble vide
- SBS est implémenté à partir de l'ensemble de toutes les variables

Pour garantir que SFS et SBS convergent vers la même solution, nous devons nous assurer que

- Les variables déjà sélectionnés par SFS ne seront pas éliminés par SBS
- Les variables déjà retirées par SBS ne seront pas sélectionnés par SFS

Par exemple, avant que SFS tente d'ajouter une nouvelle variable, il vérifie si elle a été supprimée par SBS et, si il a tenté d'ajouter la deuxième meilleure option, et ainsi de suite. SBS fonctionne de la même façon (Voir Algorithme 1.3).

Algo 1.3 :Algorithme BDS

1. Démarrer SFS avec l'ensemble vide $Y_F=\{\emptyset\}$
2. Démarrer SBS avec l'ensemble complet $Y_B=X$
3. Sélectionner la plus meilleure variable

$$X^+ = \arg\max_{\substack{x\notin Y_{F_K} \\ x\in Y_{B_K}}} [J(Y_{F_K} + X)]$$

$$Y_{F_{K+1}} = Y_{F_K} + X^+$$

4. Retirer la mauvaise variable

$$X^- = \arg\max_{\substack{x\notin Y_{B_K} \\ x\in Y_{F_{K+1}}}} [J(Y_{B_K} - X)]$$

$$Y_{B_{K+1}} = Y_{B_K} - X^-;$$

K=K+1

5. Aller à 2

1.5.2 Branch and Bound

Cette méthode est liée à la modélisation du problème de recherche du meilleur sous-ensemble sous forme de graphe. La méthode *"Branch and Bound"* (BB) appelé aussi "Séparation et Evaluation", consiste à énumérer un ensemble de solutions d'une manière intelligente en

utilisant certaines propriétés du problème en question, cette technique arrive à éliminer des solutions partielles qui ne mènent pas à la solution que l'on recherche. Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour les exclure ou les maintenir comme des solutions potentielles. La performance de cette méthode dépend de la qualité de la fonction d'évaluation partielle. Cette technique a été appliquée pour résoudre des problèmes de sélection de variables en 1977 par Narendra et Fukunaga **[30]**. Son principe est de construire un arbre de recherche où la racine représente l'ensemble des variables et les autres nœuds représentent des sous-ensembles de variables. En parcourant l'arbre de la racine jusqu'aux feuilles, l'algorithme supprime successivement la plus mauvaise variable du sous ensemble courant (nœud courant) qui ne satisfait pas le critère de sélection. Une fois que la valeur attribuée à un nœud est plus petite qu'un seuil (bound), les sous-arbres de ce nœud sont supprimés. Cette technique garantit de trouver un sous-ensemble optimal de variables à condition d'utiliser une fonction d'évaluation monotone. L'inconvénient de cette méthode est son temps de calcul qui croît rapidement avec l'augmentation du nombre de variables et qui devient impraticable à partir d'un certain nombre (30 variables). Une amélioration de cette méthode en utilisant d'autres techniques de recherche dans l'arbre afin d'accélérer le processus de sélection a été proposé dans **[31, 32].**

L'algorithme ABB de Liu and H.Motoda, 1998, est une version automatique de l'algorithme « Branch and bound » (Algorithme 7). On parle d'automatique car le seuil est déterminé automatiquement et non prédéfini. L'algorithme débute avec l'ensemble complet des variables. On enlève une variable à la fois en utilisant un parcours en profondeur d'abord jusqu'à ce qu'aucune des variables ne puisse plus être supprimée puisque le critère d'inconsistance est satisfait.

Algo 1.4 L'algorithme ABB, Automatic Branch and Bound (Liu and H.Motoda, 1998)

Entrées:

$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes

U: le taux d'inconsistance pour mesure de pertinence

S_1, S_2: des sous-ensembles de

Q : une pile vide

L : une liste pour stocker les ensembles satisfaisant le taux d'inconsistance

Sorties:

$S \subset \aleph$: le plus petit ensemble de L qui satisfait le taux d'inconsistance

$L \leftarrow \aleph$

$\delta \leftarrow \boldsymbol{U}(\aleph)$
Pour chaque variable $\boldsymbol{X_i}$ de $\aleph$ **faire**
$\boldsymbol{S_1} \leftarrow \aleph - \boldsymbol{X_i}$
Empiler $\boldsymbol{S_1}$ dans $\boldsymbol{Q}$
fin pour
Tant que $\boldsymbol{Q}$ n'est pas vide **faire**
Dépiler $\boldsymbol{Q}$ dans $\boldsymbol{S_2}$
Si $U(S_2) \leq \delta$ **alors**
$\boldsymbol{L} \leftarrow \boldsymbol{L} + \boldsymbol{S_2}$
$ABB(S_{2)}$
finsi
fin tant que
$\boldsymbol{S} \leftarrow$ le plus petit ensemble de $\boldsymbol{L}$
Renvoyer $\boldsymbol{S}$

1.5.3 FOCUS

FOCUS, méthode de filtrage pour la sélection de variables a été proposé par Almuallim et Dietterich en 1991**[33].** Cette méthode repose sur une recherche exhaustive sur l'ensemble initial de variables pour trouver le sous-ensemble le plus performant de taille minimale. L'algorithme FOCUS (algorithme 2.3) commence par générer et évaluer tous les sous-ensembles de taille *T* (initialement un), puis tous les couples de variables, les triplets et ainsi de suite jusqu'à ce que le critère d'arrêt soit satisfait.

Les inconvénients de cette méthode sont :

- la sensibilité de sa méthode d'évaluation au bruit;
-le temps de calcul qui devient énorme avec l'augmentation de la taille de l'ensemble des variables et du nombre d'exemples de la base.

Ces mêmes auteurs ont proposé FOCUS2 comme une amélioration de leur méthode initiale **[34]**. FOCUS2 est beaucoup plus rapide que FOCUS, mais elle est toujours sensible au bruit.

Algo 1.5 Algorithme de sélection *FOCUS*

Entrées: Une base d'apprentissage A = {X_1, X_2, .., X_M} où Xi = {$\boldsymbol{x_{i1}, x_{i2}, \ldots . x_{iN}}$}

T : Taille maximale de l'ensemble final et un seuil ∈

Sorties: S : ensemble final des variables

$S = \emptyset$

Pour i = 1 à T **Faire**

Pour chaque sous-ensemble (S_1) de taille (i) **Faire**

Cons = Inconsistance(A, S_1)

Si *Cons* < *e* **alors** *S* = S_1

Retourner *S*

Fin Si

Fin Pour

Fin Pour

1.5.4 Relief

Relief est une méthode de filtrage très connue et très utilisée pour la sélection de variables. Cette méthode fut proposée en 1992 par Kira et Rendell **[35]**. Son principe est de calculer une mesure globale de la pertinence des variables en accumulant la différence des distances entre des exemples d'apprentissage choisis aléatoirement et leurs plus proches voisins de la même classe et de l'autre classe, algorithme 2.4. Les avantages de cette méthode sont :

- La simplicité, la facilité de la mise en œuvre;
- La précision même sur des données bruitées.

En revanche, sa technique aléatoire ne peut pas garantir la cohérence des résultats lorsqu'on applique plusieurs fois la méthode sur les mêmes données. Cette méthode ne prend pas en compte la corrélation entre les variables. Afin d'éviter le caractère aléatoire de l'algorithme, John et al. **[6]** ont proposé une version déterministe appelée ReliefD. D'autres variantes de cette méthode ont été proposées pour améliorer sa performance **[7, 3]**

Algo 1.6 Algorithme de sélection de *Relief*

Entrées: Une base d'apprentissage A = {X_1, X_2, .., X_M} où chaque exemple

Xi = {$\boldsymbol{x_{i1}, x_{i2}, \ldots . x_{iN}}$}, Nombre d'itérations T

Sorties: W[N] : vecteur de poids des caractéristiques ($\boldsymbol{f_i}$), -1≤ W[i] ≤ 1, ∀i, W[i] = 0;

Pour *t = 1 à* T **Faire**

Choisir aléatoirement un exemple $\boldsymbol{X_k}$

Chercher deux plus proches voisins (un dans sa classe ($\boldsymbol{X_a}$) et un deuxième dans l'autre, classe$\boldsymbol{(X_b))}$

Pour i = 1 à N **Faire**

$$\boldsymbol{W[i] = W[i] + \frac{|x_{ki} - x_{bi}|}{M \times T} - \frac{|x_{ki} - x_{ai}|}{M \times T}}$$

Fin Pour
Fin Pour
Retourner W

1.5.5 LVW et LVF

Liu et Setiono ont proposé en 1996 la méthode de sélection de variables LVW (Las Vegas Wrapper) **[36]**. Elle consiste à générer aléatoirement et à chaque itération un sous-ensemble de variables et à l'évaluer avec un classificateur.

Algo 1.7 Algorithme *LVW*	**Algo 1.8** Algorithme *LVF*
Entrées: Une base d'apprentissage A Une base de caractéristiques S Nombre d'itérations T **Sorties:** S : Ensemble sélectionné Err = Classificateur (A,S) k = 0, N = |S| **Répéter** S1 = Générer_Al(), N1 = |S1| Err1 = Classificateur(A,S1) **Si** (Err1 <Err) ou (Err = Err1 et N1 < N) alors k = 0,N = N1,S = S1,Err = Err1 **Fin** Si k = k + 1 Jusqu''à k=T Retourner S	**Entrées:** Une base d'apprentissage A Une base de caractéristiques S Nombre d'itérations T et un seuil ε **Sorties:** S : Ensemble sélectionné N = |S| Pour i=1 à T Faire S1 = Générer_Al() N1 = |S1| **Si** Inconsistance (A,S1) <ε et (N1 < N) alors N = N1, S = S1 **Fin** Si Fin Pour Retourner S

Au départ, l'ensemble de base est supposé comme le meilleur sous-ensemble, ce sous-ensemble devient le meilleur sous-ensemble courant. Ce processus est répété jusqu'à ce que T essais consécutifs soient infructueux pour l'amélioration. Cette méthode présente l'inconvénient de ne pas garantir l'optimalité de la solution finale ainsi qu'un temps de calcul très élevé. Deux ans plus tard par les mêmes auteurs ont proposé LVF (Las Vegas Filter) comme nouvelle méthode de filtrage pour la sélection de variables, elle est similaire à la méthode LVW mais l'évaluation des sous-ensembles se fait par le calcul d'une mesure appelée "taux d'incohérence" ou "taux d'inconsistance"**[36]**. L'inconsistance pour un sous-ensemble de variables est définie par le rapport entre le nombre d'exemples inconsistants de la base de données et le nombre total d'exemples. Un exemple est dit inconsistant s'il existe un autre exemple qui a la même représentation dans l'espace des variables du sous-ensemble de variables étudié (appelé exemple équivalent), mais qui appartient à une autre classe. Cette méthode présente les mêmes inconvénients que la méthode FOCUS. Elle est donc très sensible au bruit et comme toutes les méthodes de recherche exhaustive, elle est très coûteuse en temps de calcul.

1.5.6 SAC

SAC (Sélection Adaptative de Caractéristiques) est une méthode de sélection de descripteurs proposée par Kachouri et al. en 2010 **[37]**. L'idée générale de cette méthode est de construire un ensemble de classificateurs SVM appris sur chacun des descripteurs et de sélectionner les meilleurs par discrimination linéaire de Fisher (FLD). Ils proposent de considérer la performance de l'apprentissage des modèles correspondant à ces descripteurs pour l'identification d'une meilleure discrimination de Fisher, algorithme 1.9.

Algo 1.9 Algorithme de sélection de *SAC*
Entrées: Une base d'apprentissage A = {X_1, X_2, .., X_M} où chaque exemple $Xk=\{desc_{k1}, desc_{k2}, ...desc_{kN}\}$, k = 1..m et $X^i = \{desc_{1i}, desc_{2i}, ...desc_{Mi}\}$, i = 1..N **Sorties:** *Ms :*les classificateurs retenus **Pour i = 1** *à N* **Faire** M_i = *Apprentissage SVM*(X^i) *Pr(Mi)* = taux de classification en utilisant le modèle M_i **Fin Pour** $\mathcal{L}$= Trier *(Pr(Mi))* par ordre décroissant ∀i ∈ {1, 2, .., N} *k = FLD*($\mathcal{L}$) **Retourner** M_S = (M_{s1},M_{s2},....M_{sk})

Après avoir construit la base d'apprentissage **M**= $(M_1, M_2, \ldots M_k)$ où N représente le nombre total de descripteurs et ***Mi*** est le modèle construit sur le i^{ème} descripteur en utilisant un classificateur SVM, les auteurs proposent une suite $\mathcal{L}$ qui représente la performance des modèles ***Mi*** triés par ordre décroissant sur lequel le score de Fisher sera calculé **[37]**

$\mathcal{L}$**= {Pr(M$_{s1}$),** ***Pr(M$_{s2}$), .., Pr(Ms$_N$)}).***

Pour calculer ce score, ces auteurs proposent de calculer deux valeurs **m$_1$(i)** et **m$_2$(i),**
i = 1..N, ces valeurs représentent les deux moyennes de performances d'apprentissage
du modèle **M$_i$** ***(Pr(M,$_i$)).***

$$m_1(i) = \frac{1}{i}\sum_{j=1}^{i} \mathbf{Pr}\,(M_{sj}),\quad m_2(i) = \frac{1}{N-i}\sum_{j=i+1}^{N} \mathbf{Pr}\,(M_{sj}) \textbf{(1.1)}$$

Deux variances sont calculées (équation 1.7.2).

$$v_1^2(i) = \frac{1}{i}\sum_{j=1}^{i} |\mathbf{Pr}\,(M_{sj}) - m_1(i)|^2,\quad v_2^2(i) = \frac{1}{N-i}\sum_{j=i+1}^{N} |\mathbf{Pr}\,(M_{sj}) - m_2(i)|^2 \textbf{(1.2)}$$

Finalement le sous-ensemble sélectionné est celui qui maximise le discriminant de Fisher *P(i))* calculé en fonction de m$_1$(i), m$_2$(i), v^2$_1$(i) et v^2 $_2$(i) :

$$P(i) = \frac{|m_1(i) - m_2(i)|}{v_1^2(i) + v_2^2(i)} \qquad \textbf{(1.3)}$$

1.5.7 Max-relevance, Min-Redundancy (mRMR)

Peng et al ont proposé en 2005,*"Max-relevance, Min-Redundancy"* (mRMR) comme méthode de filtrage pour la sélection de variables **[38]**. Cette méthode est fondée sur des mesures statistiques classiques comme l'information mutuelle, la corrélation, etc. L'idée principale est de profiter de ces mesures pour essayer de minimiser la redondance (mR) entre les variables et de maximiser la pertinence (MR). Deux variantes de cette méthode ont été présentées par cette même équipe: la première pour des données discrètes et la seconde pour des données continues.

Pour les données discrètes, les auteurs utilisent l'information mutuelle pour calculer les deux facteurs **mR** et **MR.** Le calcul de la redondance et de la pertinence d'une variable est donné par l'équation 1.4

$$\boldsymbol{Redondance(i) = \frac{1}{|F|^2}\sum_{i,j\in F} I(i,j)\,,\; Pertinance(i) = \frac{1}{|F|^2}\sum_{i,j\in F} I(i,Y)} \quad (1.4)$$

où ***F*** **et** **|*F*|** représentent, respectivement, l'ensemble des variables et sa taille. ***I(i,j)*** est l'information mutuelle entre la $i^{ème}$ **et la** $j^{ème}$ variable et finalement **I(i,Y)** est l'information mutuelle entre la $i^{ème}$ variable et l'ensemble des étiquettes de classes **(Y).** Le score d'une variable est la combinaison de ces deux facteurs tel que :

$$\boldsymbol{Score(i) = \frac{Pertinence\ (i)}{Redondance(i)}\ ou\ Score(i) = Pertinence(i) - Redondance(i)} \quad (1.5)$$

Pour les données continues, les auteurs ont remplacé l'information mutuelle par d'autres mesures. Pour la redondance ils ont utilisé la mesure de corrélation, par contre, la mesure F-statistique est utilisée pour calculer la pertinence.

Après cette évaluation individuelle des variables, une technique de recherche avant séquentielle est utilisée avec un classificateur pour sélectionner le sous-ensemble final de variables. Un classificateur est utilisé pour évaluer les sous-ensembles en commençant par la variable qui a le meilleur score, les deux meilleures, etc., jusqu'à trouver le sous-ensemble qui minimise l'erreur de classification.

1.5.8 Algorithme du MIFS.

L'algorithme MIFS (Mutual Information based Feature Selection) a été proposé par Battiti en 1994 **[51]**. Comme son nom l'indique, l'algorithme est basé sur l'utilisation d'information mutuelle afin d'évaluer la qualité d'un sous-ensemble de variables. L'idée derrière l'utilisation d'information mutuelle est de mesurer la quantité de réduction d'incertitude au sujet des résultats d'une variable **Y** fourni par la connaissance des résultats d'un ensemble de variables **X**. L'idée est alors d'évaluer l'information mutuelle entre les sous-ensembles de variables explicatives et la variable dépendante afin de trouver le sous-ensemble qui fournit les informations les plus élevées sur la variable dépendante. Cependant, l'information mutuelle ne diminue pas si une variable non pertinente ou redondante est ajoutée à un sous-ensemble variable. Ceci peut être nuisible à l'exécution des modèles de prévisions. La méthode la plus utilisé doit commencer par un sous-ensemble de cardinalité basse et ajouter progressivement des variables jusqu'à l'ajout des variables n'augmente pas de manière significative l'information mutuelle entre les sous-ensembles de variables explicatives et la variable

dépendante. Par conséquent, un algorithme vers l'avant (forward) avide (greedy) de sélection peut être employé pour rechercher un bon sous-ensemble de variables en utilisant l'information mutuelle comme mesure de la qualité du sous-ensemble de variables.

L'algorithme **Algo 1.10** calcule l'information mutuelle entre deux variables. Il est particulier parce qu'il renvoie un sous-ensemble de variables de cardinalité k défini par l'utilisateur.

Algo 1.10 MIFS :

-Mettre $\boldsymbol{k}$ *= le nombre de variables du sous-ensemble de variables sélectionnées;*

-Définir ensemble $\boldsymbol{F} = \{\boldsymbol{X_1}, \dots \dots \dots \boldsymbol{X_n}\}$ */ L'ensembles des variables explicatives* $\boldsymbol{X_i}$ *;*

-Définir ensemble $\boldsymbol{S} = \emptyset$*. / Ensemble contenant toutes les variables sélectionnées;*

-Trouver une variables $\boldsymbol{X_i^*}$*, tel que* $\boldsymbol{I}(\boldsymbol{X_i^*}, \boldsymbol{Y}) = \boldsymbol{max}_{\boldsymbol{X_i} \in \boldsymbol{F}} \boldsymbol{I}(\boldsymbol{X_i}, \boldsymbol{Y})$ */* $\boldsymbol{I}(\boldsymbol{X_i}, \boldsymbol{Y})$ *dénote l'information mutuelle entre la variable explicative* $\boldsymbol{X_i}$ *et la variable dépendante* $\boldsymbol{Y}$*.*

$\boldsymbol{F} \leftarrow \boldsymbol{F} \setminus \{\boldsymbol{X_i^*}\}$

$\boldsymbol{S} \leftarrow \{\boldsymbol{X_i^*}\}$

Répéter

Trouver $\boldsymbol{X_i} \in \boldsymbol{F}$ *qui maximise* $\boldsymbol{I}(\boldsymbol{X_i}, \boldsymbol{Y}) - \boldsymbol{\beta} \sum_{\boldsymbol{X_j} \in \boldsymbol{S}} \boldsymbol{I}(\boldsymbol{X_i}, \boldsymbol{X_j})$

$\boldsymbol{F} \leftarrow \boldsymbol{F} \setminus \{\boldsymbol{X_i}\}$

$\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{\boldsymbol{X_i}\}$

Jusqu'à $|\boldsymbol{S}| = \boldsymbol{k}$

Retourner $\boldsymbol{S}$

Dans l'algorithme MIFS, la variable sélectionnée est celle qui maximise la fonction :

$$\mathbf{I}(\mathbf{X_i}, \mathbf{Y}) - \boldsymbol{\beta} \sum_{\mathbf{X_j} \in \mathbf{S}} \mathbf{I}(\mathbf{X_i}, \mathbf{X_j})$$

Le facteur $\boldsymbol{\beta}$ permet de contrôler la pénalisation du terme de la redondance et il a une grande influence sur l'algorithme de sélection. Battiti suggère des valeurs de $\boldsymbol{\beta}$ entre 0.5 et 1.

Kwak et Choi **[54]** indiquent que ce choix de β ne donne pas des résultats satisfaisants, ils justifient cela par le fait que la sélection du premier paramètre, qui a le maximum d'information mutuelle avec la classe Y, influe considérablement sur la sélection du second paramètre via le terme de la redondance. En effet, si $\boldsymbol{\beta} = 0$, l'algorithme de sélection ne tient pas compte de la redondance des paramètres sélectionnés. Si $\beta = 1$, l'algorithme de sélection donne plus d'importance au terme de la redondance au détriment du terme de la pertinence.

Plusieurs auteurs **[51, 54, 55]** utilisent des valeurs différentes de $\boldsymbol{\beta}$ dans l'intervalle [0,1] sans aucune justification. La valeur de β est souvent déterminée expérimentalement et dépond des données utilisées.

1.5.9 Algorithme du CMIM.

Fleuret, propose en 2004, un algorithme basée sur le critère de maximisation de l'information mutuelle conditionnelle (CMIM : Conditional Mutual Information Maximization Criterion) **[52]** Il propose de choisir la variable $X_i \in X_R$ dont la pertinence conditionnelle minimale $I(X_i, Y|X_j)$ pour les variables déjà choisis $X_j \in X_s$ est maximale. Cela nécessite le calcul de l'information mutuelle de X_i par rapport à Y, conditionnellement à chaque $X_j \in X_s$ précédemment choisis. Puis, la valeur minimale est retenue et la variable dont la pertinence minimale est maximale est choisie. La sélection de variables redondantes est ainsi évitée.
La formule de la variable retournée par le critère CMIM est donnée comme suit :

$$\mathbf{X_i^{CMIM}} = \mathbf{argmax_{X_i \in X_R}}[\mathbf{min_{X_j \in X_S}}\, \mathbf{I(X_i; Y|X_j)}]$$

Autrement dit, l'algorithme de Fleuret propose une approche itérative par ajout de variables. La particularité de cet algorithme est la prise en compte des variables déjà sélectionnées. Une variable est considérée comme bonne si elle apporte suffisamment d'information sur la variable à expliquer et si cette information n'est apportée par aucune des variables déjà choisies. Plus formellement, une variable X_i est bonne si l'information mutuelle entre X_i et Y sachant X_j est suffisamment grande pour chaque variable X_j déjà choisie. La sélection de variables redondantes est ainsi évitée. En revanche, les interactions entre plus de deux variables ne sont pas étudiées.

Algo 1.11 CMIM ;

Entrées:

$\aleph$ = *{X1, . . . ,Xm} : l'ensemble des m variables booléennes potentiellement discriminantes*

Y *: la variable à expliquer*

Sorties:

$T \subset \aleph$: un sous-ensemble de $\aleph$ de taille K

$T \leftarrow \emptyset$

$T \leftarrow X_a$ tel que $a = argmax_n\,(I\,(Y; X_n))$

Pour k = 2, . . . ,K faire

$T \leftarrow T + X_a$ tel que $a = argmax_n\,(min_{l<k}\, I\,(Y; X_n|X_l))$

Fin pour

Renvoyer T

1.5.10 Algorithme FCBF (A Fast Correlation-Based Filter)

Yu et Liu **[53]** ont proposé cette algorithme ***(Algo 1.12 FCBF)*** qui s'appuie sur l'utilisation de l'incertitude symétrique ($\boldsymbol{SU}$) en tant que mesure de qualité , pour ce faire une procédure de sélection de bonnes variables pour la classification basée sur l'analyse de corrélation des caractéristiques (y compris la classe) est développée. Cela implique deux aspects :

- Comment décider si une variable est pertinente pour la classe ou non ?
- Comment décider si une telle variable pertinente est redondante ou non lors de l'examen avec d'autres variables pertinentes ?

La réponse à la première question consiste en l'utilisation d'une valeur de seuil $\boldsymbol{SU}$ choisi par l'utilisateur, comme la méthode utilisée par de nombreux autres algorithmes de pondération de variables. Plus précisément, supposons un ensemble de données $\boldsymbol{S}$ qui contient $\boldsymbol{N}$ variables et une classe $\boldsymbol{C}$. Soit $\boldsymbol{SU_{i,c}}$ désignant la valeur $\boldsymbol{SU}$ qui mesure la corrélation entre une variable $\boldsymbol{f_i}$ et la classe $\boldsymbol{C}$ (noté C - corrélation), puis un sous-ensemble $\boldsymbol{S'}$ de variables pertinentes peut être décidée par une valeur seuil $\boldsymbol{SU}$ égale à $\boldsymbol{\delta}$, tel que $\forall\, \boldsymbol{f_i} \in \boldsymbol{S'}$,$1 \leq i \leq N,\ \boldsymbol{SU_{i,c}} \geq \boldsymbol{\delta}$.
La réponse à la deuxième question est plus complexe, car il peut s'agir de l'analyse des corrélations par paires entre toutes les variables (appelées $\boldsymbol{f}$ - corrélation). Pour résoudre ce problème, la méthode ci-dessous est proposée :
Comme $\boldsymbol{f}$ - corrélations sont également capturés par les valeurs de $\boldsymbol{SU}$, afin de décider si une variable pertinente est redondante ou non, il y'a lieu de trouver ainsi un moyen raisonnable de décider du niveau de seuil pour $\boldsymbol{f}$ - corrélations . En d'autres termes , nous devons décider si le niveau de corrélation entre deux variables de $\boldsymbol{S'}$ est suffisamment élevée pour provoquer la redondance de telle sorte que l'une d'eux peut être retirer de $\boldsymbol{S'}$. Pour une variable $\boldsymbol{f_i}$ de $\boldsymbol{S'}$, la valeur de $\boldsymbol{SU_{i,c}}$ quantifie dans quelle mesure $\boldsymbol{f_i}$ est corrélé à (ou prédictive de) la classe $\boldsymbol{C}$. Si l'on examine la valeur de $\boldsymbol{SU_{j,i}}$ pour $\forall\, \boldsymbol{f_i} \in \boldsymbol{S'}$, avec ($j \neq i$) nous allons également obtenir des estimations quantifiés sur la mesure dans laquelle $\boldsymbol{f_i}$ est corrélé au (ou prédite par) reste des variables pertinentes dans $\boldsymbol{S'}$. Par conséquent, il est possible d'identifier des variables très corrélées à $\boldsymbol{f_i}$ de la même manière que nous décidons pour $\boldsymbol{S'}$, en utilisant une valeur de seuil égale $\boldsymbol{SU}$ ou semblable à $\boldsymbol{\delta}$. Nous pouvons le faire pour toutes les fonctions de $\boldsymbol{S'}$. Cependant, cette méthode semble tout à fait raisonnable quand nous essayons de déterminer les variables hautement corrélés à un concept tout en ne tenant pas compte d'une autre théorie. Dans le contexte d'un ensemble de variables pertinentes $\boldsymbol{S'}$ déjà identifiés pour le concept de classe, quand nous essayons de déterminer les variables fortement corrélées pour une fonction

donnée f_i dans S' , il est plus raisonnable d'utiliser le niveau "*C* - corrélation" entre f_i et le concept de classe $SU_{i,c}$, comme une référence. La raison se trouve sur le phénomène commun - une caractéristique qui est en corrélation avec un seul concept (par exemple, la classe) à un certain niveau peut également être corrélée avec d'autres concepts (caractéristiques) au même niveau ou d'un niveau encore plus élevé. Par conséquent, même la corrélation entre cette fonction et le concept de classe est supérieure à un certain seuil de δ et celui-ci rendant cette variable pertinente à la notion de classe, cette corrélation est loin d'être prédominante.

Algo 1.12 FCBF *(Fast Correlation-Based Filter).*

Entrées:

$S(f_1, f_2, f_3, \ldots\ldots, f_N, C)$ // Dataset d'apprentissage

δ // Seuil prédéfini

Sorties:

S_{best} // Sous ensemble optimal

Début

Pour i = 1 à N faire

Début

calculer $SU_{i,c}$ *pour* f_i ;

Si ($SU_{i,c}$ > = δ)

ajouter f_i à $\mathbf{S'}_{\text{list}}$;

Fin;

Classer $\mathbf{S'}_{\text{list}}$ par valeur descendante de $SU_{i,c}$;

f_p = *getFirstElement*($\mathbf{S'}_{\text{list}}$);

Faire début

f_q = *getNextElement*($\mathbf{S'}_{\text{list}}$, *fp*);

Si(*fq* <> NULL)

Faire début

$f'_q = f_q$;

Si ($SU_{p,q} > SU_{q,c}$)

Supprimer f_q de S'_{list};

f_q = *getNextElement*($\mathbf{S'}_{\text{list}}$, *f 'q*);

Sinon f_q = *getNextElement*($\mathbf{S'}_{\text{list}}$, *fq*);;

Fin jusqu'à (f_q == NULL);

f_p = *getNextElement* ($\mathbf{S'}_{\text{list}}$, *fp*);;

Fin jusqu'à (f_p = = NULL);

$S_{best} = S'_{list}$;

Fin;

1.5.11 **Algorithme Fisher (FISHER SCORE)**

L'algorithme Fisher repose sur l'analyse discriminante linéaire de Fisher (AFD) [88]. De façon générale, un problème de discrimination linéaire, à deux classes, revient à séparer l'espace des données en deux espaces grâce à un hyperplan. Choisir la classe (à valeur dans -1 et 1) d'une donnée consiste alors à déterminer de quel côté de l'hyperplan elle se situe, ce qui se traduit par la formule suivante :

$$g(D_j) = sign(\omega . D_j + b) \qquad (3.48)$$

où $D_j \in \mathfrak{R}^N$, **j** = 1 **. . . M,** $\omega \in \mathfrak{R}^N$. b est un biais d'estimation et $g(D_j)$ la classe prédite pour la donné D_j

L'objectif de l'AFD est de trouver la droite sur laquelle les données projetées sont mieux séparées. Cela dit, celle qui maximise le rapport des variances inter et intra classes des projections. Lorsque l'on fait l'hypothèse que les attributs sont non corrélés et suivent sur chaque classe (+**,** −) une distribution gaussienne $N(\mu^+, \sigma^+)$ *et* μ^-, σ^-, l'AFD permet de distinguer les deux classes en mesurant le chevauchement de leurs fonctions de densité de probabilité. Ainsi, les scores des attributs seront estimés par :

$$\omega_i = \frac{(\mu_i^+ - \mu_i^+)}{(\sigma_i^+)^2 + (\sigma_i^-)^2}$$

La formule peut être étendue au cas multi-classe en considérant qu'il s'agit d'un ensemble de problèmes à deux classes du type un contre tous (one-all).

1.5.12 Autres méthodes de sélection de variables basés sur l'information mutuelle

L'information mutuelle est largement utilisée pour la sélection des attributs. En général, elle mesure la quantité d'information d'une variable contenue dans une seconde. Ainsi, lorsque cette valeur est maximale, ces deux variables sont dites « identiques ». Sélectionner la variable étant le plus lié à la classe *C* peut donc se faire en maximisant leur information mutuelle. Généralement, cette information est basée sur la notion d'entropie.

En (1948) Shannon avait proposé initialement le concept d'entropie qui est une mesure de l'incertitude d'une variable aléatoire **[56] [57]**.

1.6 La sélection de variables dans la littérature

Dans ce tableau nous exposons quelques travaux concernant la résolution du problème de grande dimension, nous présentons les méthodes permettant d'y remédier et leurs applications dans les différents domaines **(Tableau 1.1).**

AUTEURS	ARTICLES	APPROCHES	EXPÉRIENCES	RÉSULTATS
Tian Lan, Deniz Erdogmus, Andre Adami, Michael Pavel, (2005)	***"Feature selection by ICA and MIM in EEG Signal Classification".***	Le schéma de sélection de variables en utilisant l'analyse linéaire en composante indépendante et l'information mutuelle(MI). Son principe est de maximiser l'information. L'évaluation du taux de classification a été faite avec le classifieur K-NN.	Utilisation du signal EEG.	Plusieurs tests ont été réalisés avec différents nombre de variables sélectionnées à partir : -20 variables le taux est 82%, -30 variables,un taux de 87%, -Après la sélection de 35 variables il a été remarquée une chute du taux de classification de 2%.
Yuhang Wing, Fillia Makedon, (2004)	***"Application of ReliefF to selecting informative genes for cancer classification using microarray data".***	Implémentation de la méthode de sélection ReliefF pour sélectionner les gènes les plus pertinents des différentes bases de données avec les classifieurs SVM et K-NN.	Les bases de données sont : - ALL leukemia, - MLL leukemia	Après la sélection de 150 gènes pour chaque base, les taux de classification sont : **SVM :** - ALL : 99% - MLL :97% **K-NN :** - ALL : 100%-MLL : 98%

AUTEURS	ARTICLES	APPROCHES	EXPÉRIENCES	RÉSULTATS
Shousken Li, Rui Xia, Ching quing Zong, Chui Ran Hueing, (2009).	***"A framework of Feature selection methods for text Categorization".***	Classification des textes, il se base sur la sélection des termes et leur classification, il compare six méthodes : DF (document frequency), MI(mutual information, IG(information gain), CHI-2(X2- test), BNS (binormal separation) et WLLR (weighted log likelihood ratio). Ces méthodes ont été implémentées pour mesurer le score entre les termes et leurs catégories.	Le corpus de Reuters-21578 dénommé R2 et 20 NG est une collection d'environ 20000 termes de 20 documents.	— DF score=0,004 — MI score =0,870 Ce qui montre que MI score a exprimé une bonne information sur la catégorie.
Yi Zhang, Chris Ding, Tao Li, (2008).	***"Gene selection algorithm by combining ReliefF and MRMR".***	Combinaison de deux méthodes de sélection ReliefF et MRMR ou la première consiste à trouver un ensemble de gènes et la seconde est appliquée explicitement pour réduire la redondance ; afin d'avoir un ensemble de gènes compacte et efficace. La classification a été réalisé avecSVM et Naive bayes.	ALL (acute lymphoblastic leukemia), ARR (Arrhythmia), GCM, HBC,MLL (leukemia)	Les taux de classification sont évalués après la sélection de 30 Gènes pour chaque base.

AUTEURS	ARTICLES	APPROCHES	EXPÉRIENCES	RÉSULTATS
Pablo A.Estévez, Michel Tesmer, Clandio A.Perez, Jacek M.Zurada, (2009)	***"Normalized mutual information Feature Selection".***	Proposition d'une normalisation de la méthode de sélection MI en NMIFS et GAMIFS. Le premier est la normalisation de l'information mutuelle pour la sélection des variables, le second est une hybridation entre les algorithmes génétiques et l'information mutuelle.	Bases de données artificielles : Sonar, Breiman, Spam base, Madelon, Arcene.	- NMIFS : Nombre de variables sélectionnées est 11 avec un taux de classification de86,36%. -GAMIFS: : Nombre de variables sélectionnées est 11 avec un taux classification de 90,96%.
B.Chandra, Manish Gupta, (2010)	***"An efficient statistical feature selection approach for Classification of gene expression data".***	Introduction de la méthode ERGS (Effective Range based Gene Selection). Son principe est que le meilleur poids est donné à la variable qui discrimine beaucoup plus la classe. L'évaluation a été faite avec Naive bayes et SVM.	La performance de l'algorithme de sélection a été évaluée sur six BDD connus des ensembles de données d'expression de gènes à savoir ALL_AML , du Tumeur du Colon, lymphome (DLBCL) [2], le cancer du poumon [21], (MLL) [4] et de la prostate [47].	COLON : NB: 83,87%, NG :100 SVM : 83,87%, NG :100

Tableau 1.1 Quelques travaux sur la Sélection de Variables

Nous concluons cette partie par un tableau récapitulatif qui met en relief la comparaison des différentes méthodes de sélection de variables les plus utilisées. Cette comparaison met en valeur les différentes caractéristiques de chaque méthode: le type, la stratégie de recherche, la complexité et la sensibilité aux bruits.

Méthode	Type	Stratégie de recherche	Non élimination de la redondance	Non prise en compte des interactions	Complexité	Dépendance à la fonction d'évaluation	Sensibilité aux bruits
SFS	Filter	Heuristique	**X**	**X**			
SBS	Filter	Heuristique	**X**		**X**		
BDS	Filter	Heuristique	**X**		**X**		
B and B	Filter ou Wrapper	Heuristique			**X**	**X**	
Focus	Filter	Exhaustive		**X**			**X**
Relief	Filter	Aléatoire	**X**	**X**			
LVW	Wrapper	Aléatoire			**X**	**X**	
LVF	Filter	Aléatoire	**X**		**X**		**X**
SAC	Hybride	Heuristique	**X**	**X**		**X**	
mRMR	Filter	Heuristique	**X**	**X**			
MIFS	Filter	Aléatoire			**X**	**X**	
CMIM	Filter	Heuristique			**X**		
FCBF	Filter	Heuristique			**X**		

Tableau 2 Résumé des méthodes de sélection présentées

1.7 Contribution

La sélection de variables est un domaine de recherche qui donne lieu à de nombreuses études et à de nouvelles approches. Les différents travaux réalisés durant ce mémoire apportent certaines contributions concernant la sélection de variables pour des problèmes de classification supervisée.

La première contribution développée dans ces travaux traite de l'application et la comparaison entre plusieurs méthodes de sélection pour la classification de variables. Notre expérimentation se focalise sur l'approche *"Filter"*, qui se révèle être meilleur dans beaucoup de domaines, comme la bioinformatique, la reconnaissance de forme et la catégorisation de textes, et ce, par rapport aux autres approches *"wrapper», "embedded"* qui nécessitent à chaque évaluation l'ajustement du modèle, ce qui se révèle être très coûteux en temps. Ces approches sont beaucoup plus adaptées à la phase de modélisation qu'à la phase de prétraitement (préparation d'une analyse). Par contre, les critères des techniques Filtre sont fondés uniquement sur des données, elles permettent à l'utilisateur d'accéder visuellement aux connaissances implicites représentées par un ensemble d'observations et de juger la pertinence des variables responsables d'un tel évènement et d'entamer une analyse plus fine de ces données en augmentant la transparence du modèle.

Nous nous sommes focalisés uniquement sur l'extraction de variables réelles et non pas sur la construction de nouvelles variables artificielles pour réduire la dimension. De là nous justifions le choix de ces différentes méthodes de sélection, nous utilisons les méthodes basés sur l'information mutuelle (MI) pour la sélection de variables , à l'instar de MIFS qui se dénote à la base par l'entropie de Shannon, la méthode MRMR (Minimum Redundancy, Maximum Relevance) qui est une extension de la méthode précédente, passant à une méthode de sélection Relief qui prend en compte les dépendances entre les variables et d'autres méthodes tel que SFS, SBSS et BDS qui s'avère être des méthodes très intéressante qui sont caractérisées par leur simplicité de calcul durant le processus de sélection.

Pour la validation des variables sélectionnés dans nos différentes bases de données, nous testons leurs performances et leurs taux de classification avec les classifieurs SVM et Naïves Bayésien.

Nous avons aussi, dans notre expérimentation appliquer un nouveau principe qui consiste à prendre les variables pertinentes *COMMUNES* sélectionnées en exécutant les différentes méthodes de sélection et de classification et leurs appliquer de nouveau le processus des sélection

de variables et de classification pour essayer de prédire le comportement de ses variables une nouvelle fois en se posant plusieurs questions:

- Est ce que les variables pertinentes communes sélectionnés initialement seront de nouveaux sélectionnées par ces algorithmes ou nous aurons de nouvelles variables?

- Est ce que les taux de classification (par opposition les taux d'erreurs) vont augmenter ou au contraire, les taux de classification vont diminués?

CHAPITRE 2.

Classification et classificateur

2.1 Classification

La classification est l'une des techniques les plus anciennes d'analyse et de traitement de données. Plusieurs définitions ont été proposées par les spécialistes du domaine :

- Pour Mari et Napoli **[39]**: "Effectuer une classification, c'est mettre en évidence des relations entre des objets, et entre ces derniers et leurs paramètres".

- Un problème de classification selon Henriet **[40]** "consiste à affecter des objets, des candidats, des actions potentielles à des catégories ou des classes prédéfinies".

- Michie et al. **[41]**ont un point de vue axé sur l'apprentissage, ils définissent la classification par : " l'action de regrouper en différentes catégories des objets ayant certains points communs ou faisant partie d'un même concept, sans avoir connaissance de la forme ni de la nature des classes au préalable, on parle alors de problème d'apprentissage non supervisé ou de classification automatique, ou l'action d'affecter des objets à des classes prédéfinies, on parle dans ce cas d'apprentissage supervisé ou de problème d'affectation" .

2.2 Types de classification

La classification repose sur des objets à classer. La résolution d'un problème de classification consiste à trouver une application de l'ensemble des objets à classer, décrits par les variables descriptives choisies, dans l'ensemble des classes. Les objets sont localisés dans un espace de variables. Ce problème n'a de sens que si on pose l'existence d'une correspondance entre ces deux espaces. L'algorithme ou la procédure qui réalise cette application est appelé classifieur.

2.2.1 La classification supervisée

L'objectif de la classification supervisée est d'apprendre, à l'aide d'un ensemble d'entraînement, une procédure de classification qui permet de prédire l'appartenance d'un nouvel exemple à une classe. En d'autre terme, l'objectif est d'identifier les classes auxquelles appartiennent des objets à partir de leurs variables descriptives **[1]**.

2.2.2 La classification non supervisée

Les méthodes de classification non supervisée ou automatique regroupent les objets en un nombre restreint de classes homogènes et séparées. Les éléments d'une classe sont les plus proches possible les uns des autres et éparées veut dire qu'il existe un écart entre les classes. La

proximité et l'écart ne sont pas nécessairement au sens de distance. L'homogénéité et la séparation entrent dans le cadre des principes de cohésion et d'isolation de Cormack **[42]**.

Les méthodes de classification automatique déterminent leurs classes à l'aide d'algorithmes formalisés. Cormack distingue entre trois familles de méthodes : la classification hiérarchique, le partitionnement et le groupement. D'autres auteurs rajoutent trois autres catégories à la taxonomie de Cormack **[43]**: la classification automatique sous contraintes, la classification automatique floue et les méthodes géométriques.

Hansen et Jaumard définissent deux autres types d'algorithmes de classification : les sous-ensembles et le Packing **[44]**.

Les méthodes de classification hiérarchique et les méthodes de partitionnement sont les plus utilisées. La classification hiérarchique peut être ascendante ou descendante, le nombre de classes n'est pas fixé au préalable. Le partitionnement est une classification non hiérarchique en un nombre fixe de classes.

2.2.3 La classification semi supervisée

Classer des données manuellement ou expérimentalement peut être une tâche longue, complexe, coûteuse, parfois impossible. Avec peu de données déjà classées, il est difficile d'obtenir un classifieur ayant toute la capacité de généralisation qu'on en attend. Pourtant, il arrive souvent que beaucoup de données non étiquetées soient disponibles comme par exemple des documents textuels, des pages html ou encore des séquences protéiques. Ces données apportent une information sur la distribution des exemples qui doit pouvoir être utilisée lors de la phase d'apprentissage.

On appelle classification semi-supervisée la problématique qui consiste à utiliser des données non étiquetées, en plus de données classées, pour le calcul d'un classifieur $\boldsymbol{f}$, afin d'influencer la procédure d'un algorithme d'apprentissage et améliorer sa performance. L'objectif est toujours d'apprendre un classifieur $\boldsymbol{f}$ qui minimise le risque $R(\boldsymbol{f})$, pour cela, nous disposons de deux ensembles de données d'apprentissage l'ensemble S_{lab} (ensemble de données labélisées) et un ensemble de données non étiquetées S_{unlab}. Généralement, la taille de S_{unlab} est beaucoup plus grande que celle de $S_{lab.}$ **[45]**

2.3 La classification supervisée

2.3.1 Formalisation mathématique

Dans le cadre de la classification supervisée, les classes sont connues et l'on dispose d'exemples (ou individus) de chaque classe.

Un exemple est un couple $(\mathbf{x}, \mathbf{y})$ ou $\mathbf{x} \in \boldsymbol{X}$ est la description ou la représentation del'objet et $\mathbf{y} \in \mathbf{Y}$ représente la supervision de **x**.

Dans un problème de classification, **y** s'appelle la classe de **x**. Pour la classification binaire nous utilisons typiquement **X** pour dénoter l'espace d'entrées tel que $X \subseteq \mathbb{R}$ et Y l'espace de sortie tel que **Y = {-1,1}** .

Soit un ensemble d'exemples de données étiquetées : $\mathbf{S = \{(x_1,y_1)... (x_n, y_n)\}}$. Chaque donnée $\mathbf{x}_i$ est caractérisée par **P** variables et par sa classe $\mathbf{y}_i$. On cherche une hypothèse telle que :

- $\boldsymbol{h}$ satisfait les échantillons $\forall \boldsymbol{i} \in \{\mathbf{1}, \dots, \boldsymbol{n}\}$, $\boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{y_i}$
- $\boldsymbol{h}$ possède de bonnes propriétés de généralisation.

Le problème de la classification consiste à prédire la classe de toute nouvelle donnée .

2.3.2 Le problème de la généralisation

L'objectif de la classification est de fournir une procédure ayant un bon pouvoir prédictif garantissant des prédictions fiables sur les nouveaux exemples qui seront soumis au système. La qualité prédictive d'un modèle peut être évaluée par le risque réel ou espérance du risque, qui mesure la probabilité de mauvaise classification d'une hypothèse **[46].**

2.3.2.1 Risque réel

Soit $\boldsymbol{h}$ une hypothèse apprise à partir d'un échantillon $\boldsymbol{S}$ d'exemples. Le risque réel de $\boldsymbol{h}$ est définit par :

$$\boldsymbol{R(h)} = \int_{\boldsymbol{x \in X \times Y}} \boldsymbol{l}\,[\boldsymbol{h(x_i)}, \boldsymbol{y_i}]\; \boldsymbol{dF(x,y)} \qquad (\mathbf{1.6})$$

où $\boldsymbol{1}$ est une fonction de perte ou de coût associé aux mauvaises classifications et où l'intégrale prend en compte la distribution $\boldsymbol{F}$ de l'ensemble des exemples sur le produit cartésien $\boldsymbol{X \times Y}$**.** La fonction de perte la plus simple utilisée en classification est définie par :

$$\boldsymbol{l}\,[\boldsymbol{h}(\boldsymbol{x_i}), \boldsymbol{y_i}] \;=\; \begin{cases} \mathbf{0} & \boldsymbol{si y_i = h(x_i)} \\ \mathbf{1} & \boldsymbol{si y_i \neq h(x_i)} \end{cases} \qquad (1.7)$$

La distribution des exemples est inconnue, ce qui rend impossible le calcul du risque réel. Le système d'apprentissage n'a en fait accès qu'à l'erreur apparente (ou empirique) qui est mesurée sur l'échantillon d'apprentissage.

2.3.2.2 Risque empirique

Soit un ensemble d'apprentissage **S = {(x_1,y_1)... (x_n, y_n)}** de taille **n** et une hypothèse ***h***. Le risque empirique de ***h*** calculé est défini par :

$$\boldsymbol{R_{emp}} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{l}\,[\boldsymbol{h}(\boldsymbol{x_i}), \boldsymbol{y_i}] \qquad \mathbf{(1.8)}$$

Le risque empirique ou apparent est simplement le nombre moyen d'exemples mal classés. On peut montrer que, lorsque la taille de l'échantillon tend vers l'infini, le risque apparent converge en probabilité vers le risque réel, si les éléments de **S** sont tirés aléatoirement. Malheureusement on ne dispose que d'un échantillon limité d'exemples ; le risque empirique est très optimiste et n'est pas un bon indicateur des performances prédictives de l'hypothèse ***h.***

2.3.2.3 Évaluation d'une hypothèse de classification

L'algorithme de validation croisée k-blocs (k-fold cross-validation) consiste à découper l'ensemble initial d'exemples D en k-blocs. On répète alors k phases d'apprentissage- évaluation où une hypothèse h est obtenue par apprentissage sur k-1blocsdedonnées et testée sur le bloc restant. La moyenne des erreurs empiriques ainsi obtenues constitue l'erreur estimée **[8]**. L'algorithme est le suivant:

1- Partitionner l'ensemble d'exemples D en k sous-ensembles disjoints : $\boldsymbol{D_1}, \ldots . \boldsymbol{D_k}$,

2- Pour tout ***i*** de ***1 à k***

- Appliquer l'algorithme sur le jeu d'apprentissage $\boldsymbol{D_1}, \ldots . \boldsymbol{D_k}$,pour obtenir une hypothèse.
- Calculer $\boldsymbol{R_i}$,l'erreur $\boldsymbol{R_i}$, sur $\boldsymbol{D_i}$

3- Retourner $\boldsymbol{R} = \dfrac{\sum_{1 \leq i \leq k} \boldsymbol{R_i}}{\boldsymbol{k}}$ ***comme estimation de l'erreur***

L'usage montre que l'évaluation par validation croisée fournit de bons résultats pour ***k=10***. Il faut noter que lorsque le nombre d'échantillons dont on dispose est limité on peut également appliquer le processus appelé Leave-One-Out Cross Validation(LOOCV) où la validation croisée est appliquée avec ***k=n*** le nombre d'échantillons **[8]**.

2.3.3 Les techniques de la classification supervisée

Pour présenter les techniques de la classification supervisée, nous avons repris la répartition formulée par Weiss et Kulikowski **[47]** qui sépare ces techniques en deux catégories:

- Les techniques statistiques;
- Les techniques d'apprentissage automatique.

Les techniques statistiques regroupent une collection de méthodes qui sont les techniques basées sur l'apprentissage Bayésien, l'analyse discriminante et la méthode des k plus proches voisins (KNN). En apprentissage automatique, nous présentons les réseaux de neurones, les arbres de décision, et les Séparateurs à Vaste Marge SVM.

2.3.3.1 L'apprentissage Bayésien

L'apprentissage Bayésien est basé sur le théorème de Bayes. Le problème de classification peut se traduire par la minimisation du taux d'erreur, ce qui peut être formulé mathématiquement en utilisant la règle de Bayes. L e classificateur Bayésien est basé sur une approche probabiliste employant la règle de Bayes. Notons $\boldsymbol{P(C_i)}$la probabilité a *priori* d'une classe $\boldsymbol{C_i}$, $\boldsymbol{P(x)}$ la probabilité d'observer un vecteur caractéristique $\boldsymbol{x}$ et $\boldsymbol{P(x|C_i)}$ la probabilité d'observer le vecteur $\boldsymbol{x}$ sachant que la classe est $\boldsymbol{C_i}$. La règle de Bayes permet alors de calculer la probabilité a *posteriori* de la classe $\boldsymbol{C_i}$ quand $\boldsymbol{x}$ est observé **[8]**:

$$\boldsymbol{P(C_i|x)} = \frac{\boldsymbol{P(x|C_i)\,P(C_i)}}{\boldsymbol{P(x)}}$$

où $\boldsymbol{P(x)}$ est le facteur de normalisation équivalent à $\sum_j \boldsymbol{P(x|C_j)\,P(C_j)}$. Ainsi, selon la règle de décision, l'observation $\boldsymbol{x}$ est affectée à la classe $\boldsymbol{C_i}$, où la probabilité a posteriori $\boldsymbol{P(C_i|x)}$ est maximum.

$$\boldsymbol{P(C_i|x)} = \frac{\boldsymbol{P(x|C_i)\,P(C_i)}}{\sum_j \boldsymbol{P(x|C_j)\,P(C_j)}}$$

Les probabilités $\boldsymbol{P(C_i)}$ de chaque classe ainsi que les distributions $\boldsymbol{P(x|C_i)}$ doivent être préalablement estimées à partir d'un échantillon d'apprentissage. Le vecteur $\boldsymbol{x}$ est assigné à la classe $\boldsymbol{C_i}$ si :

$$\forall\, \boldsymbol{j \neq i}, \boldsymbol{P(C_i|x)} > P(\boldsymbol{C_j}|\boldsymbol{x})$$

Ce principe est certainement très séduisant, mais ses performances sont fortement dépendantes de la qualité de l'échantillon qui est extrait de la population à étudier. En effet, l'estimation des probabilités a posteriori est correcte si les variables obéissent aux mêmes vraisemblances de l'échantillon utilisé pour les estimer.

C'est pourquoi, il est nécessaire de les estimer à partir des observations de l'échantillon. Les probabilités a priori des classes s'obtiennent naturellement en faisant le rapport entre le nombre d'observations de chaque classe sur le nombre total d'observations. En revanche, la difficulté majeure du théorème de Bayes demeure dans l'estimation des densités de probabilité. En effet, leur estimation est susceptible de se heurter au problème connu sous le nom de la "malédiction de la dimensionnalité".

2.3.3.2 *k* plus proches voisins:

La méthode des ***k*** plus proches voisins ***(Knn*** k-nearest neighbor en anglais) repose sur une comparaison directe entre les vecteurs caractéristiques représentant des entités de référence et le vecteur caractéristique représentant l'entité à classer. La comparaison consiste en un calcul de distances entre ces entités. L'entité à classer est assignée à la classe majoritaire parmi les classes des *k* entités les plus proches au sens de la distance utilisée **[48]**.

Notons par $X_p = (x_{p1}, x_{p2}, \dots x_{pN})$ le vecteur caractéristique de l'entité ***p***, avec ***N*** le nombre de caractéristiques et par ***p*** et ***q*** deux entités à comparer.

Les distances suivantes sont usuellement employées par les classificateurs ***Knn*** *:*

$$\textbf{Distance Euclidienne}: D(X_p, X_q) = \sqrt{\sum_{i=1}^{N}(x_{pi} - x_{qi})^2} \qquad \textbf{(1.9)}$$

$$\textbf{Distance de Manhattan}: D(X_p, X_q) = \sum_{i=1}^{N}(|x_{pi} - x_{qi|}) \qquad \textbf{(1.10)}$$

$$\textbf{Distance Minkowski}: D(X_p, X_q) = \left(\sum_{i=1}^{N}(x_{pi} - x_{qi})^r\right)^{1/r} \qquad \textbf{(1.11)}$$

$$\textbf{Distance de Tchebychev}: D(X_p, X_q) = max_{i=1}^{N}(|x_{pi} - x_{qi|}) \qquad \textbf{(1.12)}$$

Exemple de classification avec les KNN :

Dans la figure 5, à gauche, la classification est simple quel que soit le nombre de voisins choisis : le nouvel objet est noir. A droite, tout dépend du nombre de voisins choisis et de l'heuristique de classification. Pour **k = 1**, le nouvel objet est gris. Pour **k = 3**, si les trois voisins ont le même poids, alors le nouvel objet est noir. Par contre, si le poids est pondéré par l'inverse de la distance, le nouvel objet peut être gris. Cela revient à pondérer l'affectation de la classe avec la distance : plus un voisin est éloigné, plus son influence est faible.

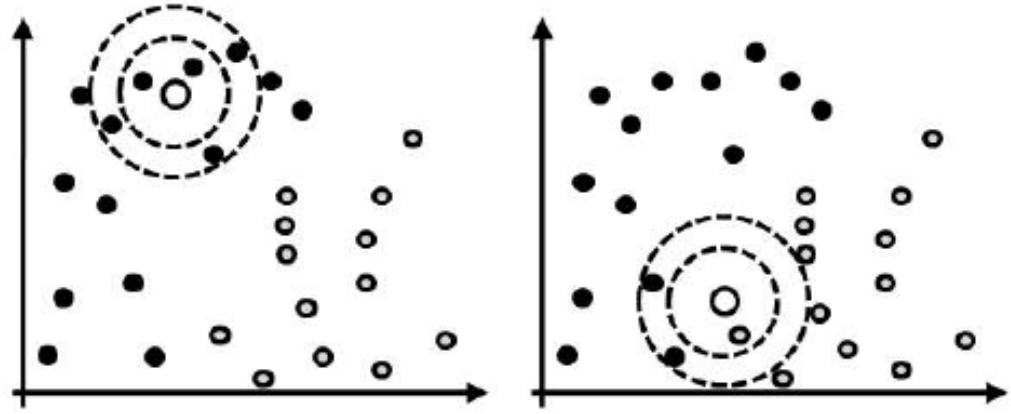


Figure 5*Exemple de classification avec les KNN* **[8]**

Les principaux inconvénients de cette méthode sont le nombre d'opérations nécessaires pour classer une entité dans le cas d'une grande base de référence ainsi que sa sensibilité au bruit présent dans les données d'apprentissage.

2.3.3.3 Les arbres de décision

Les arbres de décision ont pour objectif la classification et la prédiction. Leur fonctionnement est basé sur un enchaînement hiérarchique de règles exprimées en langage courant. Un arbre de décision est composé d'un nœud racine par lequel entrent les données, de nœuds feuilles qui correspondent à un classement de questions et de réponses qui conditionnent la question suivante. La mise en place d'un arbre de décision consiste en premier lieu à préparer les données puis à créer et valider l'arborescence. La définition de la nature, du format des variables et leur méthode de traitement reste une priorité. Ces variables peuvent être non ordonnées ou encore continues. Dans le cas de l'existence d'une base de règles simple et limitée, la construction de l'arbre se fait en interaction avec le décideur, en validant les arborescences une à la fois jusqu'à la détermination de l'affectation. C'est un processus interactif d'induction de règles qui permet

d’aboutir à une affectation bien justifiée. La création et la validation de l’arborescence se passe selon l’algorithme de calcul choisi. Différents algorithmes ont été développés: CART, C4.5 et CHAID **[1]**.

Les avantages fournis par les arbres de décision sont :

-la rapidité et la facilité quant à l’interprétation des règles de décision;

-la facilité du dialogue homme-machine grâce à la clarté des règles de décision;

-le traitement de l’ensemble d’apprentissage avec des données manquantes car ce sont des méthodes non paramétriques qui ne font aucune hypothèse sur les données.

Les arbres de décision présentent des inconvénients au niveau de la performance et le coût de l’apprentissage. Ils deviennent peu performants et très complexes lorsque le nombre de variables et de classes augmente. En effet, ils risquent de devenir trop détaillés, ce qui leur fait perdre un peu de leur lisibilité ou encore d’aboutir à de mauvais classements et d’augmenter le coût de l’apprentissage.

La figure 6 illustre un exemple de classification sur des données continues en deux dimensions en utilisant les arbres de décision.

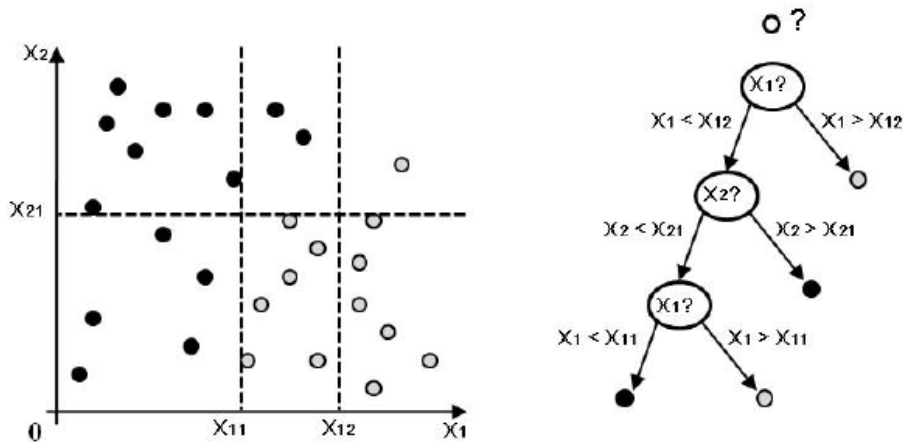


Figure 6*Exemple de classification avec les Arbres de Décision* **[8]**

2.3.3.4 Réseaux de neurones

Les réseaux de neurones sont issus de la structure neurophysiologique du cerveau. Un neurone formel est l'unité élémentaire d'un système modélisé par un réseau de neurones artificiels. A la réception de signaux provenant d'autres neurones du réseau, un neurone formel r produit un signal de sortie qui sera transmis à d'autres neurones du réseau. Le signal reçu est une somme pondérée

des signaux provenant de différents neurones. Le signal de sortie est une fonction de cette somme pondérée :

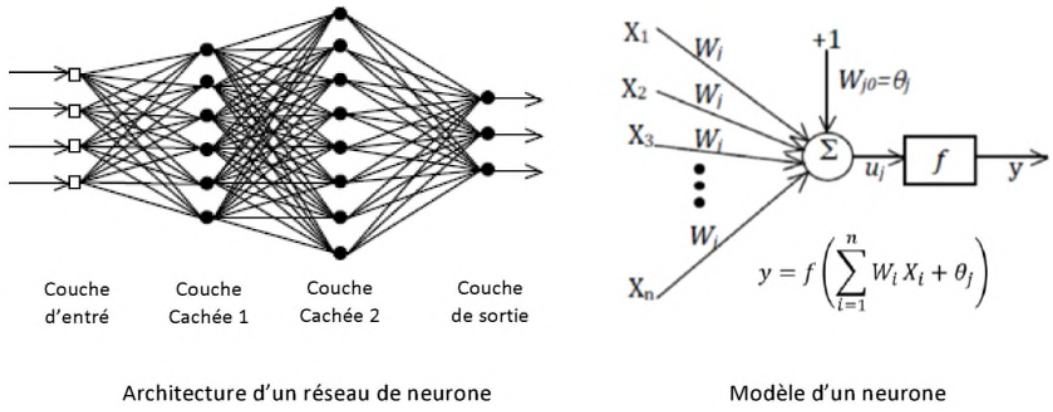


Figure 7*Représentation d'un réseau de neurones Multicouches* **[8]**

$$y_i = f(\sum_{i=1}^{N} w_{ij} x_i)$$

y_j la sortie du neurone formel j, x_i les signaux reçus par le neurone j de la part des neurones i, et w_{ij} les poids des interconnexions entre les neurones **i et *j***. Selon l'application, la fonction f, appelée *fonction d'activation,* est le plus souvent une fonction identité, sigmoïde, tangente hyperbolique ou une fonction linéaire par morceaux. En classification, les réseaux de neurones **[63]** ont permis d'introduire la non-linéarité dans la séparation entre les classes grâce au choix de la fonction d'activation. Les neurones sont ainsi organisés en trois couches ou plus. L'apprentissage du classificateur consiste à faire évoluer les poids w_{ij} par des méthodes d'optimisation non linéaires pour minimiser une fonction de coût qui constitue la mesure de l'écart entre les réponses obtenues du réseau et les réponses désirées.

2.3.3.5 Séparateurs à vastes marges

En 1998, V. Vapnik a défini un critère d'optimalité basé sur la "marge" pour séparer des classes linéairement séparables et l'a généralisé à des frontières non linéaires grâce à un changement d'espace [**46**].

L'hyperplan séparateur a pour équation : $x^T\beta + \beta_0 = 0$, avec x un vecteur caractéristique et β un vecteur de coefficients. Maximiser la plus petite distance séparant un point de l'espace des

observations à l'hyperplan séparateur, revient maximiser la marge, ce qui peut s'écrire sous la forme du problème d'optimisation suivant :

$$\begin{cases} \boldsymbol{min_{\beta,\beta 0}} ||\boldsymbol{\beta}||^2 \\ \forall \boldsymbol{j}, \boldsymbol{y^j}\left(\boldsymbol{x^{j^T}}\ \boldsymbol{\beta} + \boldsymbol{\beta 0}\right)\ \geq\ \mathbf{1} \end{cases}$$

$\boldsymbol{x^j}$ est le vecteur caractéristique de la $\boldsymbol{j^{ème}}$ observation et $\boldsymbol{y^j}$ est la classe correspondante. Ce problème admet une solution unique qui ne dépend que des points situés sur la marge (les points supports) les plus difficiles à classer. La fonction de décision des séparateurs à vastes marges (SVM) s'écrit ainsi :

$$\boldsymbol{f(x)}\ =\ \boldsymbol{\beta 0}\ +\ \sum_{\boldsymbol{support}} \boldsymbol{\alpha_i\, y_i x_i^T x}$$

$\boldsymbol{\alpha_i}$ les multiplicateurs de Lagrange. Dans le cas de données non séparables linéairement dans leur espace d'origine, un changement d'espace, généralement de dimension plus grande, peut les rendre séparables. A une frontière linéaire dans l'espace transformé, correspond une frontière non linéaire dans l'espace d'origine. L'hyperplan optimal dans l'espace transformé s'écrit :

$$\boldsymbol{f(x)} = \boldsymbol{\beta 0} + \sum_{\boldsymbol{support}} \boldsymbol{\alpha_i\, y_i} \langle \boldsymbol{\Phi}\ (\mathbf{x_i}) \mid \boldsymbol{\Phi}\ (\mathbf{x}) \rangle = \mathbf{0} \text{ avec } \boldsymbol{\Phi}\ (\mathbf{x})$$

La transformé de $\boldsymbol{x}$ dans le nouvel espace. Pour éviter de calculer explicitement les transformés des points dans le nouvel espace, on choisit une transformation qui permet le calcul du produit scalaire $\langle \boldsymbol{\Phi}\ (\mathbf{x_i}) \mid \boldsymbol{\Phi}\ (\mathbf{x}) \rangle$ en fonction de $\mathbf{x_i}$ et de $\boldsymbol{x}$ $\boldsymbol{(\langle \Phi\ (x_i) \mid \Phi\ (x) \rangle = K\ (x_i, x))}$.

K est appelé *fonction noyau* (ou *kernel).*

Les SVMs présentent également de bonnes performances en termes de généralisation. La généralisation est la faculté d'un classificateur à prédire correctement les classes de nouvelles observations et non pas seulement les classes des observations d'apprentissage [**49]**. Pour un échantillon d'apprentissage de taille *M* et une probabilité d'erreur **δ**, V.Vapnik a vérifié l'inégalité suivante :

$$\boldsymbol{R} \leq \boldsymbol{R_{emp}}\ +\ \boldsymbol{\epsilon(M, \delta)}$$

$\boldsymbol{R}$ est le risque réel, $\boldsymbol{R_{emp}}$ est le risque empirique (sur l'ensemble d'apprentissage). Cette inégalité donne une borne pour le risque à partir du risque empirique dépendant de la taille de l'échantillon

d'apprentissage et des propriétés intrinsèques au classificateur (la VC-dimension), mais pas de la distribution des observations d'apprentissage, ce qui permet d'assurer une bonne généralisation du classificateur sans avoir besoin de connaître a *priori* la distribution des observations.

Cette inégalité peut être également utilisée pour affiner le classificateur en choisissant les bons paramètres du modèle. En effet, plus le modèle est complexe, plus la VC-dimension est élevée. Le risque empirique $\boldsymbol{R_{emp}}$ décroît en fonction de la VC-dimension tandis que le terme croît en fonction de la VC-dimension. Il s'agit donc de trouver le modèle qui assure le meilleur compromis entre ajustement et généralisation.

2.3.3.6 Les algorithmes génétiques

Les algorithmes génétiques ont montré avec succès leur grande capacité à résoudre des problèmes d'optimisation. Ils ont aussi été utilisés dans le domaine de la sélection de caractéristiques. De nombreuses études rapportées dans la littérature ont montré que les méthodes qui utilisent les algorithmes génétiques comme technique de recherche ont donné des meilleurs résultats en comparaison avec les autres méthodes de sélection **[38]**

Les algorithmes génétiques sont des algorithmes itératifs basés sur la reproduction et l'évolution naturelle des individus en utilisant les principes de la survie des individus considérés comme les plus forts ou les mieux adaptés à l'environnement. Il s'agit alors de combiner les points forts de chaque individu pour en créer de nouveaux de manière à ce que leur efficacité soit meilleure. Avec les algorithmes génétiques on cherche à optimiser une fonction ***(objectif)*** donnée dans un espace de recherche, celui des individus. Le fonctionnement d'un algorithme génétique est basé généralement sur les phases suivantes **(Figure** 8**) :**

1. **Initialisation :** Générer aléatoirement une population initiale de taille N chromosomes.
2. **Evaluation :** Évaluer chaque individu de la population par la fonction d'évaluation appropriée au problème.
3. **Reproduction :** Créer une nouvelle population de N chromosomes par l'utilisation d'une méthode de sélection appropriée et l'application d'opérateurs génétiques (croisement et mutation) sur certains chromosomes au sein de la population courante.
4. **Retour** à la phase 2 tant que la condition d'arrêt du problème n'est pas satisfaite.

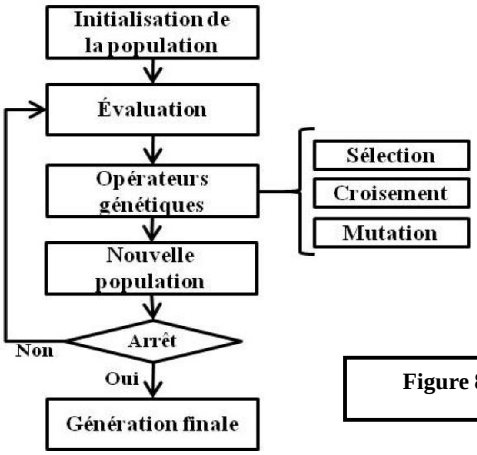


Figure 8 ***Architecture générale d'un algorithme génétique*** **[1]**

Algorithme 1.13 Principe d'un algorithme génétique.
Début génération aléatoire d'une population de *N* individus **Répéter** - reproduction et donc production de *M* descendants par croisement et mutation de la population - évaluation de chaque individu avec une fonction d'adaptation au problème - remplacement de tout ou partie de la population par les *N* meilleurs individus parmi les *N* + *M* disponibles **Jusqu'à** convergence d'un critère de qualité ou nombre maximal de génération atteint **Fin**

2.4 Conclusion

Dans ce chapitre nous avons défini le concept de classification et les techniques utilisées, nous avons présenté les concepts de la classification non supervisée, semi-supervisée et détaillé le concept de classification supervisée qui présente un intérêt pour notre étude.

CHAPITRE 3.

Expérimentation

3.1 Introduction

Dans ce chapitre, nous présentons en premier lieu la plate forme de développement et format de fichier de données utilisés pour la réalisation du logiciel, qui sert pour l'expérimentation de différentes méthodes de sélection de variables sélectionnées dans ce projet. Nous abordons, par la suite, la partie de modélisation de notre application. Nous poursuivons par l'expérimentation réalisée et les résultats obtenus par l'expérimentation et les résultats obtenus et leurs interprétations. Enfin nous faisons une comparaison avec les résultats de certains travaux déjà réalisés afin de situer les résultats obtenus par rapport à ces travaux et mettre en évidence nos contributions.

3.2 Plateforme de développement

La plate-forme de développement utilisée dans notre projet, permet de dérouler et d'exécuter des programmes écrits en langage JAVA, indépendamment de toute architecture matérielle et de tout type système d'exploitation. Ces applications logicielles sont portables et flexibles.

Pour des besoins de chargement rapides de données (bases de données ou Datasets) lors de l'exécution des différentes méthodes de sélection et/ou algorithmes de classification, nous utilisons des fichiers plats, contenant l'ensemble de la structure -simple- et des données exploités lors de l'exécution du programme applicatif.

Ces bases de données sont dotés de l'extension (*.arff) ou (*.data), légères et ne nécessitant pas de conversion ou de moulinette (ODBC: Open Datbases Converter) pour les adapter pour nos traitements, ce qui augmenterait considérablement le temps de traitement plus que celui nécessaire à l'exécution des différents programmes des méthodes de sélection de variables.

3.2.1 Le langage JAVA:

Java est un langage de programmation récent (les premières versions datent de 1995) développé par Sun MicroSystems. Il est fortement inspiré des langages C et C++.

Comme C++, Java fait partie de la « grande famille » des **langages orientés objets**. Il répond donc aux trois principes fondamentaux de l'approche orientée objet (POO) : **l'encapsulation**, **le polymorphisme** et **l'héritage.**

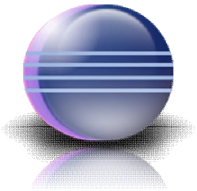
Les principales raisons du succès de Java :

Java a rapidement intéressé les développeurs pour quatre raisons principales :

- C'est un langage orienté objet dérivé du C, mais plus simple à utiliser et plus « pur » que le C++. On entend par « pur » le fait qu'en Java, on ne peut faire que de la programmation orienté objet contrairement au C++ qui reste un langage hybride, c'est-à-dire autorisant plusieurs styles de programmation. C++ est hybride pour assurer une compatibilité avec le C ;
- Il est doté, en standard, de bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boites de dialogue, contrôles, menus, graphisme), la programmation multi-threads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau … L'utilisation de ces bibliothèques facilitent grandement la tâche du programmeur lors de la construction d'applications complexes ;
- Il est doté, en standard, d'un mécanisme de gestions des erreurs (les exceptions) très utile et très performant. Ce mécanisme, inexistant en C, existe en C++ sous la forme d'une extension au langage beaucoup moins simple à utiliser qu'en Java ;
- Il est multi plates-formes : les programmes tournent sans modification sur tous les environnements où Java existe (Windows, Unix, Linux,…) ;

Pour écrire et exécuter nos programmes en JAVA, nous utilisons, une des plates formes existantes en accès libre, permettant de réaliser facilement différentes types d'applicatifs sous JAVA, c'est la plate forme de développement "Eclipse".

L'environnement de développement Eclipse :



Eclipse est un environnement de développement intégré, libre, extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

"Eclipse" possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug-ins.
- Plusieurs versions d'un même plug-in peuvent cohabiter sur une même plate-forme.
- Un support multi langage grâce à des plug-ins dédiés : Cobol, C, PHP, C#, ...
- Support de plusieurs plates formes d'exécution : Windows, Linux, Mac OS X, ...
- Malgré son écriture en Java, Eclipse est très rapide à l'exécution grâce à l'utilisation de la bibliothèque SWT;
- Les nombreuses fonctionnalités de développement proposées par le JDT (refactoring très puissant, complétion de code, nombreux assistants, ...);
- Une ergonomie entièrement configurable qui propose selon les activités à réaliser différentes « perspectives »;
- La construction incrémentale des projets Java grâce à son propre compilateur qui permet en plus de compiler le code même avec des erreurs, de générer des messages d'erreurs personnalisés, de sélectionner la cible (java 1.3 ou 1.4) et de mettre en œuvre le scrapbook (permet des tests de code à la volée);
- Une exécution des applications dans une JVM dédiée sélectionnable avec possibilité d'utiliser un débogueur complet (points d'arrêts conditionnels, visualiser et modifier des variables, évaluation d'expression dans le contexte d'exécution, changement du code à chaud avec l'utilisation d'une JVM 1.4, ...)
- Propose le nécessaire pour développer de nouveaux plug-ins
- Possibilité d'utiliser des outils open source : CVS, Ant, Junit

La plate-forme est entièrement internationalisée dans une dizaine de langue sous la forme d'un plug-in téléchargeable séparément

3.3 Diagramme et exploitation de l'application

L'utilisateur de notre application est un expert en datamining, ou un cogniticien, il a besoin d'un outil logiciel pour l'aider à effectuer le déroulement et l'exécution des méthodes

de sélection de variables et/ou de classification sur des données disposées dans des bases de données et pouvoir interpréter ces résultats. De ce fait, l'application doit répondre à certaines caractéristiques :

- Une interface intuitive et simple à utiliser
- Rapidité d'exécution;
- Une bonne performance des résultats.
- Un minimum d'erreur.
- Facilement Interprétable

Nous présentons ci-après un diagramme (Figure 9) explicatif sur le fonctionnement et l'exploitation de l'application développée pour expérimenter les différentes méthodes de sélection de variables.

Ce diagramme présente un "Expert" pour notre cas, l'exploitant de cet outil logiciel pour faire l'ensemble des tests et essais au niveau de l'application. L'exploitant - l'Expert- commence par charger ses données, puis il a le choix entre lancer la classification avec l'un des algorithmes SVM ou NB sur ces données uniquement, ou choisir une méthode de sélection et un algorithme de classification.
Dans les deux cas, les résultats affichés sont : Taux d'erreur, temps d'exécution et la liste de variables sélectionnées. L'utilisateur pourra par la suite interpréter les résultats obtenus.

Dans la partie annexe (Annexe2), un ensemble de masque de saisie relative à l'application développée et spécifiant l'interface en détail est présentée.

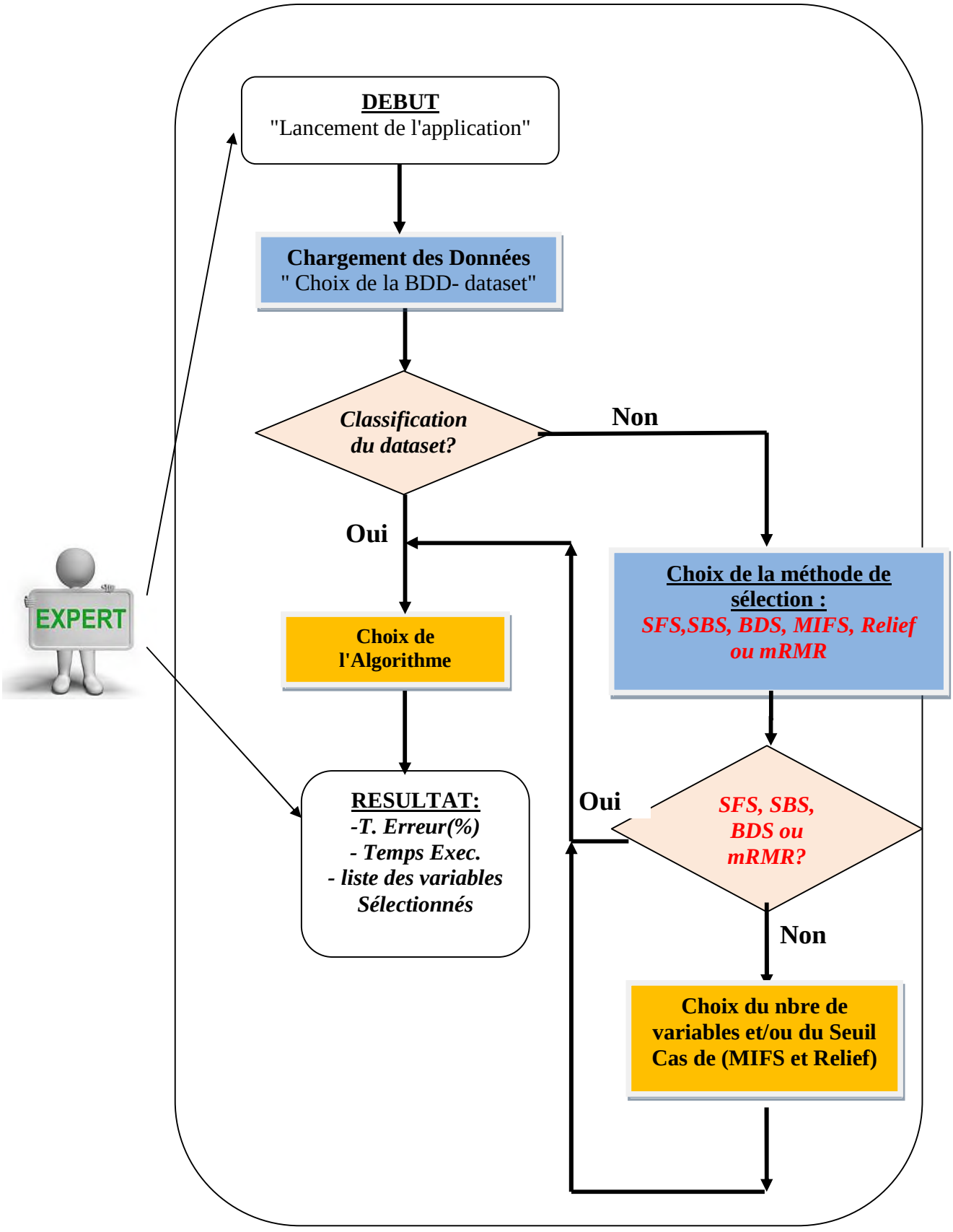


Figure 9 *Diagramme simplifié pour l'exploitation de l'application*

3.4 Les jeux de données

Pour nos expérimentations, et en vue, d'analyser les performances des différentes méthodes de sélection utilisées, nous avons sélectionné un ensemble de bases sélectionnées parmi celles de l'UCI **[64]**. Ces données diffèrent principalement par leurs nombres d'instances et de variables.

Chaque dataset dispose d'un nombre d'instances qui ont été répartis, à raison de 70% pour la phase d'apprentissage et 30% pour la phase de test. Le dataset test est utilisé pour évaluer les performances des algorithmes utilisées. Cette approche permet d'éviter le sur-apprentissage.

Les caractéristiques de ces bases sont présentées en annexe. Toutes les variables ont été discrétisées afin de pouvoir comparer les différentes méthodes entre elles.

Le tableau ci-dessous (Tableau 3-1) résume les caractéristiques des jeux de données utilisés et qui concernent différents domaines, tel que les problèmes; de diabètes, de reconnaissance de cancers ou de prévision de diagnostic en oncologie.

Nous préciserons aussi, que toutes les expérimentations ont été effectuées sur un micr-ordinateur doté d'un processeur pentium i5-2,5Ghz et d'une mémoire Ram de 08 GO.

Jeux de données	***Nombre de Classe***	***Nombre d'instances***	***Nombre de variables***
Iris	**3**	**150**	**4**
Pima Diabètes	**2**	**768**	**8**
Breast cancer	**2**	**699**	**9**
Leukemia	**2**	**72**	**7129**
Lung Cancer	**2**	**181**	**12533**

Tableau 3. 1 *Caractéristiques des jeux de données*

3.4.1 Les résultats obtenus et comparaisons

Les objectifs des expérimentations effectuées sur les cinq jeux de données sont :

1)- *Tester l'effet de la non sélection de variables en consignant les résultats obtenus: taux de classification, temps d'exécution et taux d'erreur ; Cela permettra de déduire lequel des deux algorithmes d'apprentissage (SVM ou NB) offre un meilleur taux de classification par rapport aux datasets sélectionnés, d'une part et d'autre parts, les comparer aux autres résultats issus de l'application des algorithmes de sélection de variables.*

2)- *Montrer les performances (TC, temps d'exécution) de chaque algorithme de sélection de variables sur différents jeux de données utilisés dans notre expérimentation sans classification;*

3)- *Comparer, analyser et interpréter les résultats des différents algorithmes de sélection de variables sans classification pour chaque data-set.*

4)- *Refaire les opérations 2) et 3) en plus de l'exécution des algorithmes d'apprentissage et interpréter les résultats dans le détail.*

5)- *retirer les variables pertinentes communes issus de l'opération 4 des différents algorithmes pour chaque data-set et refaire les opérations 2) et 3) et interpréter les résultats obtenus.*

Nous précisons que les algorithmes de classification utilisés sont SVM et BN et l'évaluation du taux de classification et faite par une validation croisée (Cross Validation) pour les data-sets dont le nombre d'instance est supérieur à 100 et de type LOOCV (Leave One Out Cross validation) pour ceux dont le nombre d'instance est inférieur à 100 exemples.

On utilisera ainsi :

- 10-CV pour Breast Cancer, Iris, Pima Diabetes, et Lung Cancer Data Sets;
- LOOVC pour Leukemia et Lung Cancer

Le tableau (Tableau 3-2) montre les taux de classification pour les cinq jeux de données pour chacun des algorithmes SVM et NB.

	Classifieur Utilisé							
	SVM				NB			
	Training Data (70%)		Test Data (30%)		Training Data (70%)		Test Data (30%)	
Dataset	T. C (%)	T.E.(sec)	T. C (%)	T.E.(sec)	T. C (%)	T.E.(sec)	T. C (%)	T.E.(sec)
Iris	94,89	0.017	95,93	0.013	94,78	0.017	95,83	0.015
PIMA	75,24	0.032	78,58	0.025	74,23	0.038	79,10	0.026
Breast Cancer	95,41	0.023	97,90	0.020	95,21	0.026	97,31	0.020
Leukemia	99,96	0.038	99,98	0,035	99,91	0,028	99,94	0,025
Lung Cancer	99,95	1,99	99,99	0,85	97,96	0,80	99,98	0,06

*NOTA: **T.C(%)** = Taux de classification ; **T.E.(sec)** = Temps d'exécution en secondes*

Tableau 3. 2 Taux de classification (%) sans sélection de variables

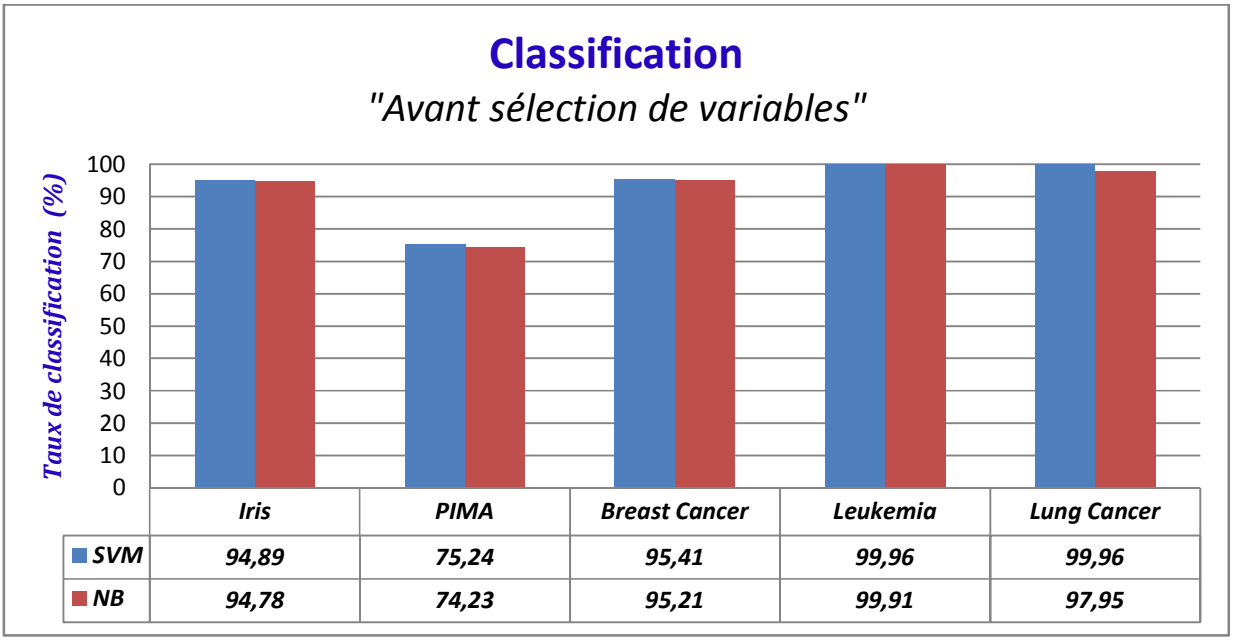


***Figure 10** Graphique des taux de classifications des datasets par les classifieurs SVM et NB*

*Avant sélection de variables **(Training Data)***

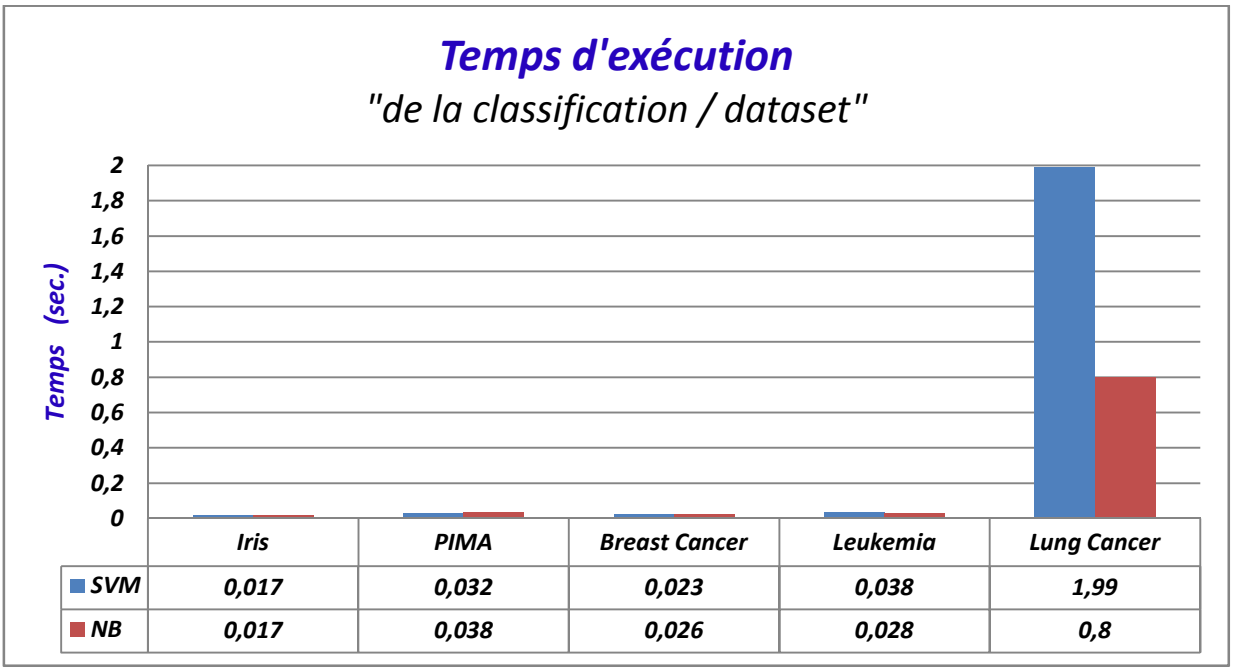


Figure 12 *Graphique des temps d'exécution (sec.) des datasets* ***(Training Data)*** *par les classifieurs SVM et NB*

3.4.2 Discussions sur l'utilisation des classifieurs

- Les résultats du tableau 3-2 montrent que les taux de classification sont satisfaisants pour certaines bases de données à l'instar de l'instar de Iris, Leukemia, Breast cancer et moyenne pour d'autres Pima Diabète;

- Les Classifieur SVM donne de meilleurs résultats que Naïves Bayésiens, Les performances dégradées du classifieur NB sont dues essentiellement à la sensibilité de ce classifieur aux variables corrélées.

3.4.3 Analyse de l'évaluation expérimentale avec méthode de sélection.

Dans la suite, nous comparons les performances de six (06) méthodes de sélection de variables à savoir les méthodes filtre MRMR, SFS, SBS, BDS, Relief et MIFS. Pour ces méthodes nous avons utilisé les classifieurs SVM et NB pour estimer la fonction d'aptitude par la validation croisée de type K- Fold CV ou LOOCV.

Nous avons testé et comparé les différentes méthodes de sélection de variables et nous avons résumé les résultats de nos expérimentations dans les tableaux détaillés 3-1 à 3.14.

Ces résultats sont également synthétisés de manière graphique dans les figures 13 et 14

Les tableaux permettent d' évaluer le comportement général des diverses méthodes d'apprentissage testées lorsqu' elles sont associées à une méthode de sélection de variables. En effet, ils présentent la valeur moyenne du rapport « taux de classification correct » de chaque méthode de sélection pour l' ensemble des algorithmes d'apprentissage utilisés -SVM et NB -et l'évaluation de la méthode est réalisée *avec K-Cross-Validation(pour les datasets ayant un nombre d'instances > 100), ou avec LOOCV dans le cas contraire* **.** Les taux de classification sont obtenus sur l'ensemble des cinq jeux de données considérés ont été réalisées avec K=10, soit 10-Cross-Validation. Les résultats permettent de conclure que, de manière générale, l' ensemble des méthodes de sélection de variables impliquent l' obtention d' un taux d' erreur quasi-équivalent lorsque les variables fournies par ces méthodes ou l' ensemble total des variables sont utilisées. De ce fait, quelle que soit la méthode d'apprentissage utilisée, les taux d'erreurs sont corrects et quasiment similaires. On peut toutefois remarquer qu' il existe un léger déficit au niveau de la qualité d' apprentissage, pour l' ensemble des méthodes de sélection, lorsqu' elles sont associées à la méthode d' apprentissage NB ou SVM. Les méthodes de sélection ***SBS, Relief et mRMR*** donnent lorsqu' elles sont associées aux classifieurs SVM une nette amélioration de la qualité d'apprentissage par rapport au Naïves Bayésien.

a) Experimentation Data set IRIS (UCI)

		Méthode de sélection + Classifieur SVM				Classifieur SVM		
		Training Data (70%)				Test Data (30%)		
Dataset	Méthode Selec.	T.C. (%)	T.Exe (sec)	N.V.S (unité)	V.S (liste)	T.C. (%)	T.E. (Sec)	V.S. (liste)
Iris	SFS	94,78	0.017	2	0;2	95,77	0.01	0;2
	SBS	94,88	0.016	2	2;3	96,91	0.015	2;3
	BDS	94,82	0.015	3	1;2;3	95,86	0.013	1;2;3
	MIFS	94,33	0.018	3	0;1;2	95,35	0.016	0;1;2
	Relief	94,87	0.017	2	1;2	95,90	0.011	1;2
	mRMR	94,33	0.018	3	0;1;2	95,35	0.016	0;1;2

NOTA: ***T.C(%)*** *= Taux de classification ;* ***T.E.(sec)*** *= Temps d'exécution en secondes*
N.V.S *= Nombre de variables sélectionnées;* ***V.S****= Variables sélectionnées*

Tableau 3. 3 Taux de classification (%) avec sélection de variables pour jeux *de données* ***IRIS*** *avec le classifieur* ***SVM***

		Méthode de sélection + Classifieur NB				Classifieur NB		
		Training Data (70%)				Test Data (30%)		
Dataset	Méthode Selec.	T.C. (%)	T.Exe (sec)	N.V.S (unité)	V.S (liste)	T.C. (%)	T.E. (Sec)	V.S. (liste)
Iris	SFS	94,18	0.016	2	0;2	95,64	0.015	0;2
	SBS	94,73	0.017	2	2;3	95,81	0.016	2;3
	BDS	94,62	0.015	3	1;2;3	95,78	0.017	1;2;3
	MIFS	93,22	0.018	3	0;1;2	94,67	0.018	0;1;2
	Relief	94,68	0.017	2	1;2	95,80	0.016	1;2
	mRMR	93,22	0.018	3	0;1;2	94,67	0.018	0;1;2

NOTA: ***T.C(%)*** *= Taux de classification ;* ***T.E.(sec)*** *= Temps d'exécution en secondes*
N.V.S *= Nombre de variables sélectionnées;* ***V.S****= Variables sélectionnées*

Tableau 3. 4 Taux de classification (%) avec sélection de variables pour jeux *de données* ***IRIS*** *avec le classifieur* ***NB***

b) Experimentation Data set PIMA Diabetes (UCI)

		Méthode de sélection + Classifieur SVM				***Classifieur SVM***		
		Training Data (70%)				***Test Data (30%)***		
Dataset	***Méthode Selec.***	***T.C. (%)***	***T.Exe (sec)***	***N.V.S (unité)***	***V.S (liste)***	***T.C. (%)***	***T.E. (Sec)***	***V.S. (liste)***
PIMA Diabetes	**SFS**	*73,68*	*0.024*	*6*	*0;1;2;3;4;6*	*78,49*	*0.017*	*0;1;2;3;4;6*
	SBS	*75,57*	*0.026*	*6*	*0;1;2;4;5;6*	*80,00*	*0.018*	*0;1;2;4;5;6*
	BDS	*75,48*	*0.025*	*7*	*0;1;2;3;4;5;7*	*80,33*	*0.016*	*0;1;2;3;4;5;7*
	MIFS	*74,37*	*0.027*	*5*	*1;2;4;5;7*	*77,88*	*0.018*	*1;2;4;5;7*
	Relief	*75,78*	*0.024*	*5*	*1;2;3;4;5*	*78,90*	*0.017*	*1;2;3;4;5*
	mRMR	*75,88*	*0.025*	*5*	*0;1;2;3;4;*	*78,78*	*0.018*	*0;1;2;3;4*

NOTA: ***T.C(%)*** *= Taux de classification ;* ***T.E.(sec)*** *= Temps d'exécution en secondes*
N.V.S *= Nombre de variables sélectionnées;* ***V.S****= Variables sélectionnées*

Tableau 3. 5 *Taux de classification (%) avec sélection de variables pour jeux de données* ***PIMA Diabètes*** *avec le classifieur* ***SVM***

		Méthode de sélection + Classifieur NB				***Classifieur NB***		
		Training Data (70%)				***Test Data (30%)***		
Dataset	***Méthode Selec.***	***T.C. (%)***	***T.Exe (sec)***	***N.V.S (unité)***	***V.S (liste)***	***T.C. (%)***	***T.E. (Sec)***	***V.S. (liste)***
PIMA Diabetes	**SFS**	*73,86*	*0.026*	*6*	*0;1;2;3;4;6*	*76,37*	*0.019*	*0;1;2;3;4;6*
	SBS	*75,06*	*0.030*	*6*	*0;1;2;4;5;6*	*80,04*	*0.021*	*0;1;2;4;5;6*
	BDS	*75,43*	*0.024*	*7*	*0;1;2;3;4;5;7*	*80,43*	*0.019*	*0;1;2;3;4;5;7*
	MIFS	*75,04*	*0.025*	*5*	*1;2;4;5;7*	*79,04*	*0.021*	*1;2;4;5;7*
	Relief	*76,21*	*0.026*	*5*	*1;2;3;4;5*	*79,15*	*0.019*	*1;2;3;4;5*
	mRMR	*73,47*	*0.026*	*5*	*0;1;2;3;4;*	*77,54*	*0.019*	*0;1;2;3;4;*

NOTA: ***T.C(%)*** *= Taux de classification ;* ***T.E.(sec)*** *= Temps d'exécution en secondes*
N.V.S *= Nombre de variables sélectionnées;* ***V.S****= Variables sélectionnées*

Tableau 3. 6 *Taux de classification (%) avec sélection de variables pour jeux de données* ***PIMA Diabètes*** *avec le classifieur* ***NB***

c) Experimentation Data set Breast Cancer (UCI)

		Méthode de sélection + Classifieur SVM				Classifieur SVM		
		Training Data (70%)				Test Data (30%)		
Dataset	Méthode Selec.	T.C. (%)	T.Exe (sec)	N.V.S (unité)	V.S (liste)	T.C. (%)	T.E. (Sec)	V.S. (liste)
Breast Cancer	SFS	94,08	0.027	6	0;2;3;4;5;6	97,98	0.016	0;2;3;4;5;6
	SBS	96,24	0.028	5	0;2;4;5;7	98,81	0.017	0;2;4;5;7
	BDS	95,25	0.03	6	1;2;4;5;6;8	97,81	0.02	1;2;4;5;6;8
	MIFS	95,77	0.031	7	0;1;2;3;4;5;7;8	98,82	0.019	0;1;2;3;4;5;7;8
	Relief	95,85	0.029	5	0;1;2;6;7	98,81	0.021	0;1;2;6;7
	mRMR	95,55	0,029	5	0;1;2;3;4	97,98	0,020	0;1;2;3;4

NOTA: T.C(%) = Taux de classification ; T.E.(sec) = Temps d'exécution en secondes
N.V.S = Nombre de variables sélectionnées; V.S= Variables sélectionnées

Tableau 3. 7 *Taux de classification (%) avec sélection de variables pour jeux de données* ***Breast Cancer*** *avec le classifieur* ***SVM***

		Méthode de sélection + Classifieur NB				Classifieur NB		
		Training Data (70%)				Test Data (30%)		
Dataset	Méthode Selec.	T.C. (%)	T.Exe (sec)	N.V.S (unité)	V.S (liste)	T.C. (%)	T.E. (Sec)	V.S. (liste)
Breast Cancer	SFS	95,48	0.030	6	0;2;3;4;5;6	97,90	0.023	0;2;3;4;5;6
	SBS	95,90	0.033	5	0;2;4;5;7	98,00	0.021	0;2;4;5;7
	BDS	94,13	0.032	6	1;2;4;5;6;8	97,08	0.025	1;2;4;5;6;8
	MIFS	95,97	0.028	7	0;1;2;3;4;5;7;8	97,96	0.026	0;1;2;3;4;5;7;8
	Relief	95,35	0.029	5	0;1;2;6;7	98,00	0.027	0;1;2;6;7
	mRMR	95,32	0,030	5	0;1;2;3;4	97,96	0,027	0;1;2;3;4

NOTA: T.C(%) = Taux de classification ; T.E.(sec) = Temps d'exécution en secondes
N.V.S = Nombre de variables sélectionnées; V.S= Variables sélectionnées

Tableau 3. 8 *Taux de classification (%) avec sélection de variables pour jeux de données* ***Breast Cancer*** *avec le classifieur* ***NB***

d) Expérimentation Leukemia** (Leucémie) **(UCI Dataset)

			Méthode de sélection + Classifieur SVM				***Classifieur SVM***		
			Training Data (70%)				***Test Data (30%)***		
Dataset	***Méthode Selec.***	***N.V. Util.***	***T.C. (%)***	***T.Exe (sec)***	***N.V.S (unité)***	***V.S (liste)***	***T.C. (%)***	***T.E. (Sec)***	***V.S. (liste)***
Leukemia	**SFS**	-	-	-	-	-	-	-	-
	SBS	-	-	-	-	-	-	-	-
	BDS	-	-	-	-	-	-	-	-
	MIFS	-	-	-	-	-	-	-	-
	Relief	*20*	*95,00*	*0,035*	*20*	*(*)*	*95,11*	*0,034*	*(*)*
		50	*98,72*	*0,036*	*50*	*(*)*	*99,00*	*0,035*	*(*)*
		100	*99,90*	*0,035*	*100*	*(*)*	*99,98*	*0,035*	*(*)*
	mRMR	*20*	*96,01*	*0,66*	*20*	*(*)*	*96,97*	*0,035*	*(*)*
		50	*99,93*	*0,68*	*50*	*(*)*	*99,97*	*0,034*	*(*)*
		100	*99,98*	*0,72*	*100*	*(*)*	*99,99*	*0,033*	*(*)*

NOTA: ***N.V.Util.** = Nombre de variables utilisées; **T.C(%)** = Taux de classification ;**V.S**= Variables sélectionnées;*
***T.E.(sec)** = Temps d'exécution en secondes; **N.V.S** = Nombre de variables sélectionnées;*

Tableau 3. 9 *Taux de classification (%) avec sélection de variables pour jeux de données **Leukemia** avec le classifieur **SVM***

			Méthode de sélection + Classifieur NB				*Classifieur NB*		
			Training Data (70%)				*Test Data (30%)*		
Dataset	***Méthode Selec.***	***N.V. Util.***	***T.C. (%)***	***T.Exe (sec)***	***N.V.S (unité)***	***V.S (liste)***	***T.C. (%)***	***T.E. (Sec)***	***V.S. (liste)***
Leukemia	**SFS**	-	-	-	-	-	-	-	-
	SBS	-	-	-	-	-	-	-	-
	BDS	-	-	-	-	-	-	-	-
	MIFS	-	-	-	-	-	-	-	-
	Relief	*20*	*94,85*	*0,027*	*20*	*(*)*	*95,12*	*0,025*	*(*)*
		50	*98,00*	*0,028*	*50*	*(*)*	*98,13*	*0,026*	*(*)*
		100	*99,01*	*0,027*	*100*	*(*)*	*99,90*	*0,025*	*(*)*
	mRMR	*20*	*95,98*	*0,03*	*20*	*(*)*	*96,33*	*0,025*	*(*)*
		50	*99,90*	*0,028*	*50*	*(*)*	*99,93*	*0,026*	*(*)*
		100	*99,97*	*0,027*	*100*	*(*)*	*99,98*	*0,026*	*(*)*

NOTA: ***N.V.Util.*** *= Nombre de variables utilisées;* ***T.C(%)*** *= Taux de classification ;* ***V.S****= Variables sélectionnées;* ***T.E.(sec)*** *= Temps d'exécution en secondes;* ***N.V.S*** *= Nombre de variables sélectionnées;*

Tableau 3. 10 *Taux de classification (%) avec sélection de variables pour jeux de données* ***Leukemia*** *avec le classifieur* ***NB***

REMARQUE : sachant que le nombre de variables est de 7129 (nombre important), le programme a bloqué lors de l'exécution des programmes pour les méthodes de sélection SFS, SBS, BDS et MIFS, par saturation de la mémoire du poste PC.

(*) : La liste des variables est trop longue pour la caser au niveau dans ce tableau , d'une part , et d'autre part, les variables ont été choisis aléatoirement lors de l'expérimentation.

e) Experimentation Lung Cancer (UCI Dataset)

			Méthode de sélection + Classifieur SVM				*Classifieur SVM*		
			Training Data (70%)				***Test Data (30%)***		
Dataset	***Méthode Selec.***	***N.V. Util.***	***T.C. (%)***	***T.Exe (sec)***	***N.V.S (unité)***	***V.S (liste)***	***T.C. (%)***	***T.E. (Sec)***	***V.S. (liste)***
Lung Cancer	**SFS**	-	-	-	-	-	-	-	-
	SBS	-	-	-	-	-	-	-	-
	BDS	-	-	-	-	-	-	-	-
	MIFS	-	-	-	-	-	-	-	-
	Relief	*50*	*98,98*	*0,65*	*50*	*(*)*	*99,98*	*0,29*	*(*)*
		100	*99,95*	*0,68*	*100*	*(*)*	*99,99*	*0,31*	*(*)*
		200	*99,77*	*0,72*	*200*	*(*)*	*100*	*0,32*	*(*)*
	mRMR	*50*	*97,00*	*0,66*	*50*	*(*)*	*98,88*	*0,37*	*(*)*
		100	*98,98*	*0,68*	*100*	*(*)*	*99,75*	*0,41*	*(*)*
		200	*99,77*	*0,72*	*200*	*(*)*	*99,98*	*0,43*	*(*)*

NOTA: ***N.V.Util.*** *= Nombre de variables utilisées;* ***T.C(%)*** *= Taux de classification ;* ***V.S****= Variables sélectionnées;* ***T.E.(sec)*** *= Temps d'exécution en secondes;* ***N.V.S*** *= Nombre de variables sélectionnées;*

Tableau 3. 11 *Taux de classification (%) avec sélection de variables pour jeux de données* ***Lung Cancer*** *avec le classifieur* ***SVM***

			Méthode de sélection + Classifieur NB				*Classifieur NB*		
			Training Data (70%)				*Test Data (30%)*		
Dataset	***Méthode Selec.***	***N.V. Util.***	***T.C. (%)***	***T.Exe (sec)***	***N.V.S (unité)***	***V.S (liste)***	***T.C. (%)***	***T.E. (Sec)***	***V.S. (liste)***
Lung cancer	**SFS**	-	-	-	-	-	-	-	-
	SBS	-	-	-	-	-	-	-	-
	BDS	-	-	-	-	-	-	-	-
	MIFS	-	-	-	-	-	-	-	-
	Relief	*50*	*95,00*	*0,45*	*50*	*(*)*	*96,31*	*0,35*	*(*)*
		100	*98,45*	*0,50*	*100*	*(*)*	*99,98*	*0,42*	*(*)*
		200	*99,78*	*0,58*	*200*	*(*)*	*99,99*	*0,47*	*(*)*
	mRMR	*50*	*93,44*	*0,45*	*50*	*(*)*	*95,00*	*0,38*	*(*)*
		100	*96,45*	*0,50*	*100*	*(*)*	*98,78*	*0,40*	*(*)*
		200	*98,78*	*0,58*	*200*	*(*)*	*99,99*	*0,43*	*(*)*

NOTA: ***N.V.Util.*** *= Nombre de variables utilisées;* ***T.C(%)*** *= Taux de classification ;* ***V.S****= Variables sélectionnées;*
T.E.(sec) *= Temps d'exécution en secondes;* ***N.V.S*** *= Nombre de variables sélectionnées;*

Tableau 3. 12 *Taux de classification (%) avec sélection de variables pour jeux de données* ***Lung Cancer*** *avec le classifieur* ***NB***

REMARQUE : sachant que le nombre de variables est de 12533 (nombre important), le programme a bloqué lors de l'exécution des programmes pour les méthodes de sélection SFS, SBS, BDS et MIFS, par saturation de la mémoire du poste PC.

(*) : La liste des variables est trop longue pour la caser au niveau dans ce tableau , d'une part , et d'autre part, les variables ont été choisis aléatoirement lors de l'expérimentation.

Jeu de données	Classifieur SVM													
	Avant sélection		SFS		SBS		BDS		MIFS		Relief		mRMR	
	TC (%)	Nbr. Var.	TC (%)	Nbr. Var.	TC (%)	Nbr. Var.	TC (%)	Nbr. Var.	TC (%)	Nbr. Var.	TC (%)	Nbr. Var.	TC (%)	Nbr. Var.
Iris	*94,89*	*4*	*94,78*	*2*	*94,88*	*2*	*94,82*	*3*	*94,33*	*3*	*94,87*	*2*	*94,33*	*3*
Pima	*75,24*	*8*	*73,68*	*6*	*75,57*	*6*	*75,48*	*7*	*74,37*	*5*	*75,78*	*5*	*75,80*	*5*
Breast Can.	*95,41*	*10*	*94,08*	*6*	*96,24*	*5*	*95,25*	*6*	*95,77*	*7*	*95,85*	*5*	*95,55*	*5*
Leukemia *(1)*	*99,96*	*7129*	-	-	-	-	-	-	-	-	*99,90*	*100*	*99,98*	*100*
Lung Can. *(2)*	*99,95*	*12533*	-	-	-	-	-	-	-	-	*99,77*	*200*	*99,77*	*200*

Tableau 3. 13 *Evaluation des méthodes de sélection avec algorithme* ***SVM*** ***(Training data)***

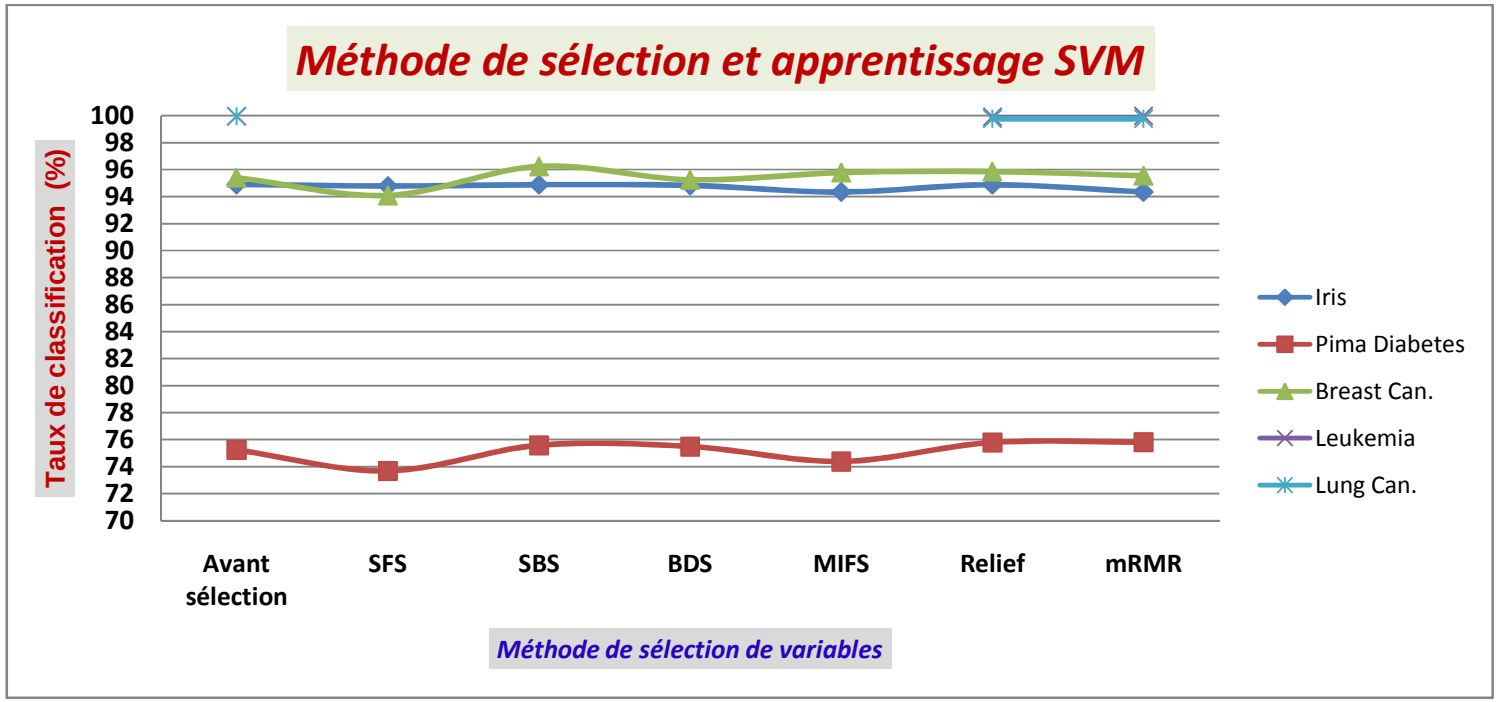


Figure 13 *Taux de classification moyen (%) des méthodes testées avec le classifieur* ***SVM*** ***(Training Data)***

Jeu de données	Classifieur NB													
	Avant sélection		SFS		SBS		BDS		MIFS		Relief		mRMR	
	TC (%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.
Iris	*94,78*	*4*	*94,18*	*2*	*94,73*	*2*	*94,62*	*3*	*93,22*	*3*	*94,68*	*2*	*93,22*	*3*
Pima	*74,23*	*8*	*73,86*	*6*	*75,06*	*6*	*75,43*	*7*	*75,04*	*5*	*76,21*	*5*	*73,47*	*5*
Breast	*95,21*	*10*	*95,48*	*6*	*95,90*	*5*	*94,13*	*6*	*95,97*	*7*	*95,35*	*5*	*95,32*	*5*
Leukemia(1)	*99,91*	*7129*	-	-	-	-	-	-	-	-	*99,01*	*100*	*99,97*	*100*
Lung Can.(2)	*97,95*	*12533*	-	-	-	-	-	-	-	-	*99,78*	*200*	*98,78*	*200*

Tableau 3. 14 *Evaluation des méthodes de sélection avec algorithme* ***NB*** ***(Training data)***

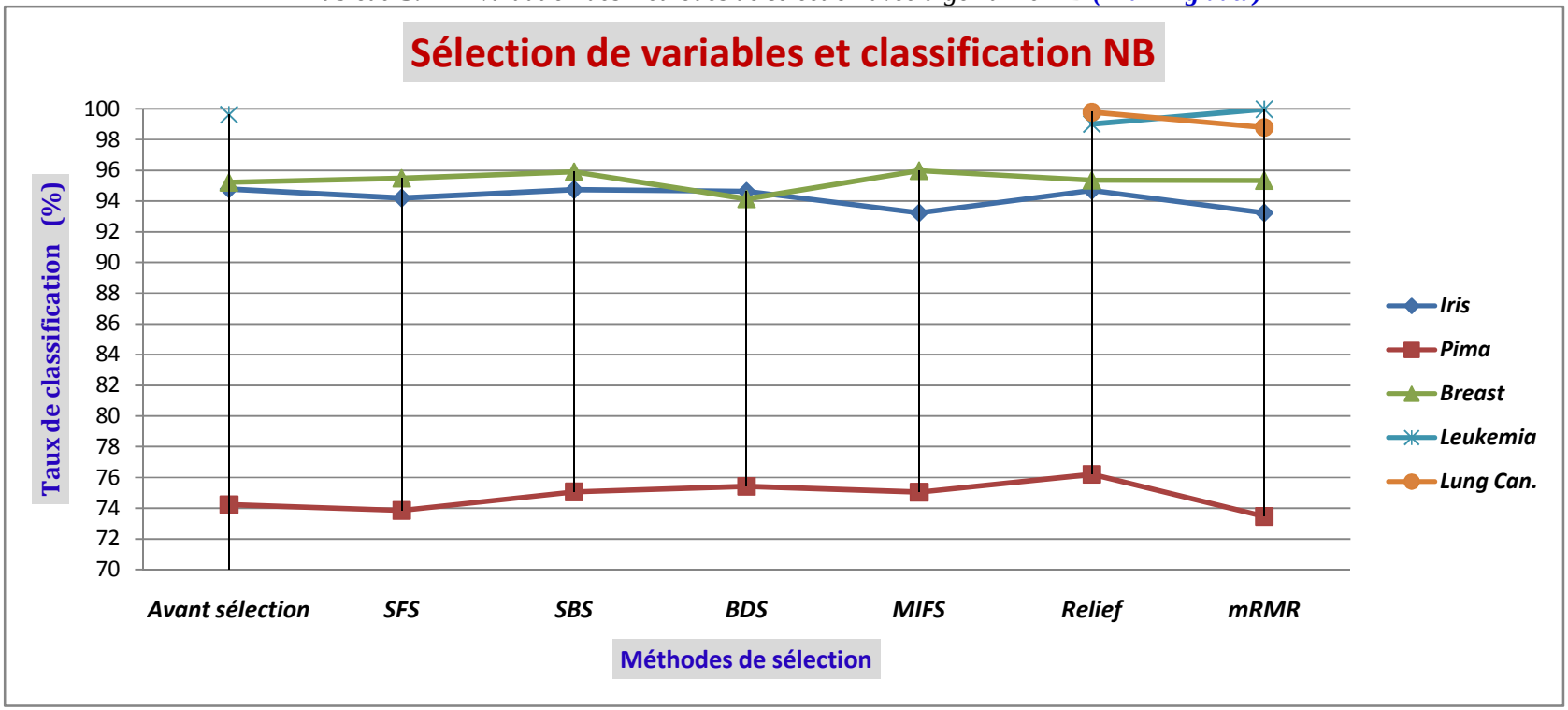


Figure 14 *Taux de classification moyen (%) des méthodes testées avec le* ***classifieur NB*** ***(Training data)***

Jeu de données	Classifieur SVM													
	Avant sélection		SFS		SBS		BDS		MIFS		Relief		mRMR	
	TC (%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.
Iris	95,93	4	95,77	2	96,91	2	95,86	3	95,35	3	95,90	2	95,35	3
Pima	78,58	8	79,49	6	80,00	6	80,33	7	78,88	5	77,90	5	78,78	5
Breast Can.	97,90	10	97,98	6	98,81	5	97,81	6	98,82	7	98,81	5	97,98	5
Leukemia	99,98	7129	-	-	-	-	-	-	-	-	99,98	100	99,99	100
Lung Can.	99,99	12533	-	-	-	-	-	-	-	-	100	200	99,98	200

Tableau 3. 15 *Evaluation des méthodes de sélection avec algorithme* ***SVM*** ***(Test data)***

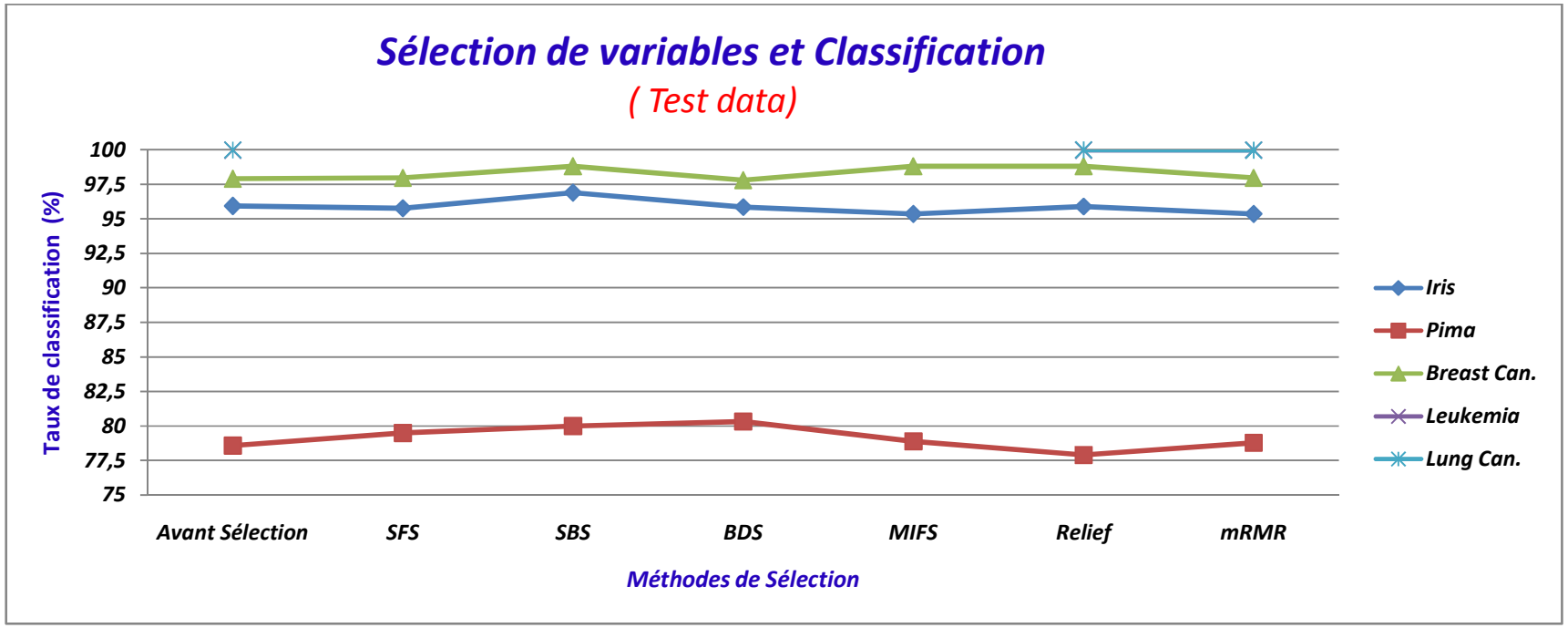


Figure 15 *Taux de classification moyen (%) des méthodes testées avec le* ***classifieur SVM*** ***(Test data)***

Jeu de données	Classifieur NB													
	Avant sélection		SFS		SBS		BDS		MIFS		Relief		mRMR	
	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.	TC(%)	Nbr. Var.
Iris	*95,83*	*4*	*95,64*	*2*	*95,81*	*2*	*95,78*	*3*	*94,67*	*3*	*95,80*	*2*	*94,67*	*3*
Pima	*79,10*	*8*	*76,37*	*6*	*80,04*	*6*	*80,43*	*7*	*79,04*	*5*	*79,15*	*5*	*77,54*	*5*
Breast	*97,31*	*10*	*97,90*	*6*	*98,00*	*5*	*97,08*	*6*	*97,96*	*7*	*98,00*	*5*	*97,96*	*5*
Leukemia	*99,94*	*7129*	-	-	-	-	-	-	-	-	*99,90*	*100*	*99,98*	*100*
Lung Can.	*99,98*	*12533*	-	-	-	-	-	-	-	-	*99,99*	*200*	*99,99*	*200*

Tableau 3. 16 *Evaluation des méthodes de sélection avec algorithme* ***NB*** ***(Test data)***

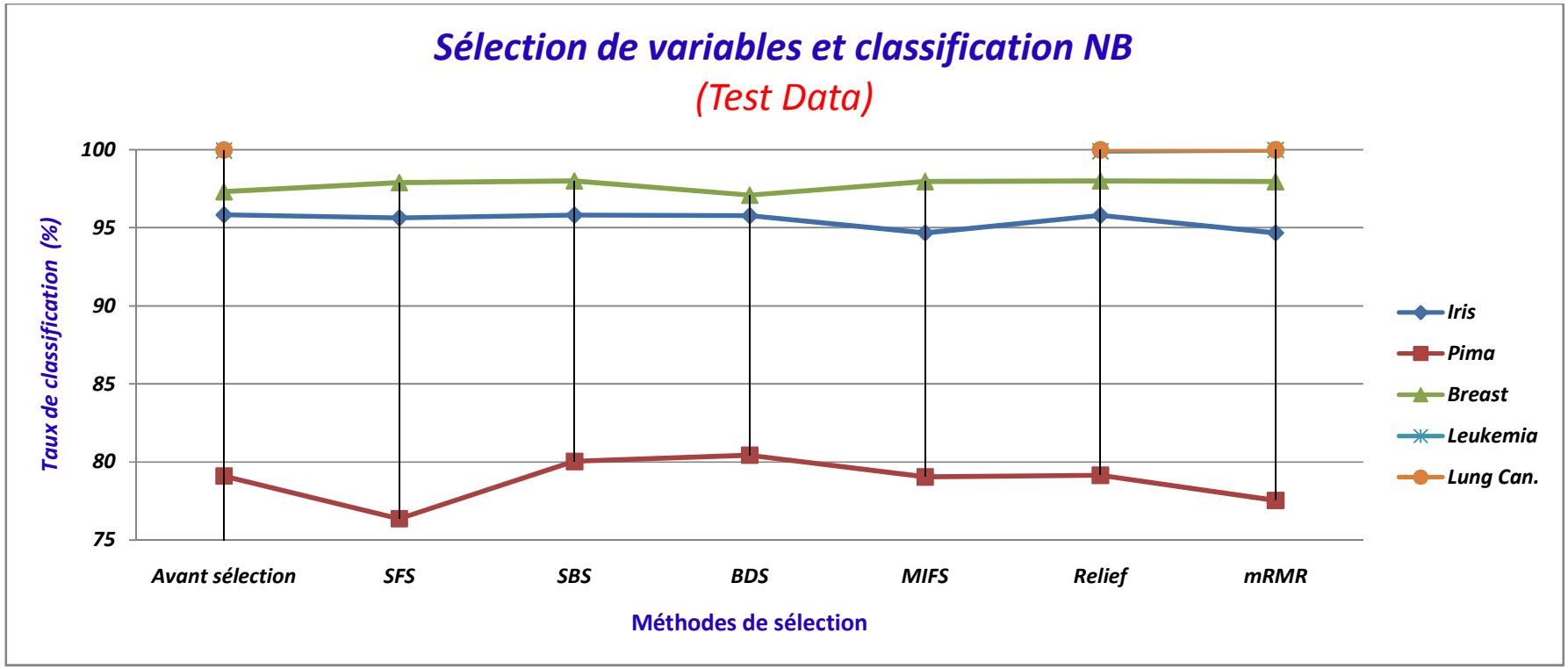


Figure 16 *Taux de classification moyen (%) des méthodes testées avec le classifieur* ***NB*** ***(Test data)***

3.5 Interprétation des résultats

3.5.1 Expérimentation avec le classifieur SVM

Les résultats expérimentaux présentés sur le **tableau 3.13** permettent d'analyser le comportement du taux de succès ***(de la méthode de classification SVM)*** avant et après exploitation des différentes méthodes de sélection :

a) Pour le ***dataset « Iris »,*** vu que la base ne contient pas beaucoup de variables exogènes, la sélection n'a pas amélioré le taux de succès mais on remarque que, le nombre de variables est réduit de 50% avec la méthode de sélection ***« SBS »*** et le temps d'exécution après sélection (0,016 sec.) est meilleur qu'avant la sélection de variables (0,017 sec.). Nous pouvons clairement dire que la sélection de variables et la classification ont amélioré les performances de traitement.

b) Pour le ***dataset « Pima Diabetes »,*** composée initialement de huit (08) variables exogènes, le nombre de variables est réduit par quatre méthodes (SBS, BDS, Relief et mRMR) avec une réduction à cinq (05) avec une amélioration des taux de succès de ***0,54% et de 0,56%*** respectivement pour les méthodes sélection ***« Relief » et « mRMR »***. On voit bien que la méthode de sélection qui l'emporte est ***« mRMR ».***

c) Pour le dataset ***« Breast Cancer »*** , composé initialement de dix (10) variables exogènes , le nombre de variables a été réduit par quatre méthodes de sélection ***(SBS, MIFS, Relief et mRMR)*** avec une réduction à cinq (05), soit une réduction de ***50%*** par rapport au nombre de variables pour ***"SBS"***. Seul la méthode ***« SBS»*** l'emporte avec un taux de succès de ***0,83%*** ;

d) Pour le dataset ***« Leukemia »*** , nous avons échantillonné à partir du dataset original comprenant 7129 variables, trois datasets contenant respectivement 20,50 et 100 variables. Nous avons opéré à une expérimentation avec ces trois fichiers, arrivée au fichier comptant 100 variables , le taux de succès était identique , voir meilleur avec une réduction conséquente pour du nombre de variables (passaage de 7139 à 100 variables). Cette expérimentation a été faite en s'inspirant de la littérature pour le même dataset , sachant que ces deux méthodes ***Relief et mRMR*** nous permettent de sélection des variables lors de l'expérimentation, à l'inverse des autres méthodes SBS, SFS, BDS…., pour lesquels la machine (PC) à bloqué par overflow de la mémoire .

e) Pour le dataset ***« Lung cancer »*** , comprenant 12533 variables, nous avons procédé à l'identique du processus d'expérimentation utilisé pour le dataset Leukemia, en

effectuant l'expérimentation sur trois datasets issus du dataset original et comprenant respectivement 50,100 et 200 variables. Le taux de succès était identique , voir meilleur avec une réduction conséquente pour du nombre de variables (passage de 12533 à 200 variables). Cette expérimentation a été faite en s'inspirant de la littérature pour le même dataset , sachant que ces deux méthodes ***Relief et mRMR*** nous permettent de sélection le nombre de variables, à l'inverse des autres méthodes SBS, SFS, BDS…., pour lesquels la machine (PC) à bloqué par overflow de la mémoire .

3.5.2 Expérimentation avec le classifieur NB :

Les résultats expérimentaux présentés sur le **tableau 3.14** permettent d'analyser le comportement du taux de succès ***(de la méthode de classification NB)*** avant et après utilisation des différentes méthodes de sélection :

a) Pour le ***dataset « Iris »,*** la sélection n'a pas apportée une amélioration du taux de succès, par contre, elle a réduit du nombre de variables de moitié avec la méthode de sélection du ***SBS*** **.**Le taux de classification est identique par rapport à l'avant sélection. Nous pouvons que le taux de succès reste appréciable sachant que le nombre de variables correctes a été réduit de moitié, par rapport au nombres de variables avant sélection.

b) Pour le ***dataset « Pima Diabetes »,*** nous nous somme focalisé d'abord sur la réduction du nombre de variables, puis sur l'amélioration du taux de succès. Ce Dataset est composée initialement de huit (08) variables exogènes, le nombre de variables est réduit par quatre méthodes de sélection, en l'occurrence, ***SBS, BDS, MIFS et Relief***. ***Relief*** a fournir les meilleurs résultats avec une réduction de variables à cinq (05) et une amélioration des taux de succès de ***0,98%*** . On voit bien que la méthode de sélection qui l'emporte est ***« Relief».***

c) Pour le dataset ***« Breast Cancer »*** , composé initialement de Dix (10) variables exogènes , le nombre de variables a été réduit à cinq (05), soit une réduction de 50% du nombre de variables avant application de la méthode de sélection ***« SBS»*** *qui l'emporte sur les autres méthodes* ***SFS, MIFS, relief et mRMR*** *qui ont-elles aussi réduit le nombre de variables et amélioré le taux de succès par rapport à l'avant sélection*. La méthode ***« SBS »*** l'emporte avec un taux de succès de ***0,83%*** .

f) Pour le dataset ***« Leukemia »*** , nous avons échantillonné à partir du dataset original comprenant 7129 variables, trois datasets contenant respectivement 20,50 et 100

variables. Nous avons opéré à une expérimentation avec ces trois fichiers, arrivée au fichier comptant 100 variables , le taux de succès était identique , voir meilleur avec une réduction conséquente pour du nombre de variables (passage de 7139 à 100 variables). Cette expérimentation a été faite en s'inspirant de la littérature pour le même dataset , sachant que ces deux méthodes ***Relief et mRMR*** nous permettent de sélection des variables lors de l'expérimentation, à l'inverse des autres méthodes SBS, SFS, BDS…., pour lesquels la machine (PC) à bloqué par overflow de la mémoire . Nous précisons que la méthode ***"mRMR"*** a amélioré le taux de succèes para rapport à l'avant sélection et l'emporte sur la méthode ***"Relief"***

d) Pour le dataset ***« Lung cancer »*** , nous gardons la même analyse et interprétation, citée au ***point 3.5.1, alinéa e).***

3.5.3 Comparaison des résultats entre SVM et NB :

Une lecture des résultats des tableaux 3.13 à 3.16, nous permet de dire que :

1- Nous avons obtenu une réduction du nombre de variables avec toutes les méthodes de sélection expérimentées, et ce, par rapport à l'avant sélection (application de l'apprentissage seul) ;

2- Les variables correctes sélectionnées par chaque méthode de sélection de variables avec les deux classifieurs sont identiques. Ceci peut nous amener à dire que la classification à augmenter ou diminuer le taux de classification mais n'intervient pas sur la choix des variables sélectionnées au niveau de chaque méthode de sélection.

3- Les méthodes ***«SBS» « Relief »*** **et** ***« mRMR »*** ont données avec les datasets expérimentés de meilleurs taux de succès, par rapport à l'avant sélection (application de l'apprentissage sans sélection de variables), exception faite pour le dataset « Iris » ou nous avons constaté une légère réduction du taux de succès mais un meilleur taux de variables sélectionnées, par rapport à l'avant sélection. Cela est du au fait que nous sommes en face de larges datasets, à l'exception de "Iris" Dataset.

4- La sélection de variables avec l'algorithme d'apprentissage SVM donne de meilleurs taux de succès que l'apprentissage avec le classifieur Naïves Bayésien, et ce , pour les datasets composés de beaucoup de variables.

5- Les résultats trouvés au niveau des tableaux 3.13 et 3.14 pour la partie Training set, sont confirmés comme étant de bon résultats après expérimentation des testing dataset contenus au niveau des tableaux 3.15 et 3.16 qui sont traduits en graphiques ,respectivement graphique 14 et 15.

3.6 Conclusion

Les méthodes de sélection de variables utilisées ont permis de construire des prédicteurs efficaces pour un problème de classification supervisée de données pouvant servir à différents domaines de la vie quotidienne (Biologie, bioinformatique, catégorisation de textes, imagerie...). Les performances obtenues sont aussi bonnes, que ceux des meilleurs prédicteurs publiés à ce jour pour les mêmes bases de données.

Notre principale contribution est d'obtenir ces performances avec un nombre minimal de variables. Cette caractéristique est importante pour la robustesse de nos prédicteurs avec une condition nécessaire à une possible utilisation à d'autres liées à d'autres domaines (Finances, Marketing, Energie, Bioinformatique, données spatiales …..

CONCLUSION GENERALE

La littérature abondante depuis plusieurs décennies sur le problème de sélection de variables (features selection) témoigne non seulement sur son importance mais aussi sur ces difficultés; Le choix des caractéristiques pertinentes pour une application donnée n'est pas aisé.

Notre démarche de sélection de variables a consisté dans un premier temps à comparer les performances de plusieurs méthodes de sélection, afin de mettre en évidence la transparence de notre système, avec un objectif d'extraire les variables les plus pertinentes et les plus informatives. Les expérimentations réalisées ont permis d'évaluer les performances des résultats avec les différents classifieurs. .

Bien que les résultats obtenus soient intéressants et encourageants, beaucoup de points sont susceptibles d'être étudiés dans le cadre de travaux futurs, tel que :

- L'utilisation d'autres mesures de sélection de variables pour mettre en valeur les différentes relations entre les variables.

-D'après l'étude des avantages et des inconvénients des méthodes de sélection utilisées dans ce travail, une hybridation entre les techniques est envisageable ou la fusion entre les points forts de ces méthodes.

- Refaire d'autres expérimentations avec les autres approches 'Wrapper et/ou Embedded, pour comparer les résultats aux notres;

- Faire appel aux méthodes de d'évaluation de l'erreur, tel que « le boosting » pour améliorer encore le taux de classification,

- Disposer d'autres Datasets, tel la catégorisation de texte, ou la reconnaissance de formes pour connaitre le comportement des méthodes de sélection de variables et leurs performances;

Ce domaine de recherche restera toujours actif tant qu'il est motivé par l'évolution des systèmes de collecte et de stockage des données d'une part et par les exigences d'autre part. La meilleure approche pour juger cette sélection est de collaboré avec des experts de différents domaines pour une interprétation des résultats et mettre en évidence de nouvelles performances.

Cette collaboration avec les experts permet de nous orienter vers la manière d'utiliser ces données fondamentales en pratique et leurs influences sur la prise de décision car ce domaine de recherche est majeur dans la prédiction de l'évolution future des événements.

BIBLIOGRAPHIE

BIBLIOGRAPHIE:

[1] Ali El Akadi, *"Contribution à la sélection des variables pertinentes en classification supervisé"*, thèse de doctorat, (2012).

[2] Le petit Robert , 60.000 mots, (2013).

[3] Liu, H., Motoda, H., *"Feature extraction construction and selection"*, Boston: Kluwer Academic,(1998).

[4] Bennani,Y., *"Systèmes d'apprentissage connexionnistes : sélection de variables"*, Revue d'Intelligence Artificielle. Hermes Science Publications, Paris, France, vol. 15, pp. 3–4, (2001).

[5] Kohavi & John,. *"Wrappers for feature selection"*. Artificial Intelligence, 97(1-2), 273-324. ,(1997).

[6] John et al,. *"Irrelevant features and the subset selection problem in machine learning"* : Proceedings of the Eleventh International, pages 121-129. Morgan Kaufmann., (1994)

[7] Koller, D., & Sahami, M. *"Toward optimal feature selection"*. 13th Internattional Conference on Machine Learning, (pp. 1-15), (1996)

[8] Chouaib, H., *"Sélection de caractéristiques: méthodes et applications"*, thèse de doctorat, (2011).

[9] Dash et Liu,.*"Hybride search of feature subsets"*. Dans Springer (Edition)., (2006).

[10] Dash et Liu,. *"Feature selection for classification"*. Intelligent Data Analysis, 1:131-156., (1997).

[11] Siedlecki, W., &Sklansky, J. *"On automatic feature selection"*. International Journal of Pattern Recognition and Artificial Intelligence, 2, pp. 197-220.,(1998).

[12] Guyon, I., &Elisseeff, A. *"An introduction to variable and feature selection"*. Journal of Machine Learning Research, 3, pp. 1157-1182., (2003)

[13] Le Petit Larousse illustré , (2013)

[14] I. R. et L. Personaz, *"Mlps (mono-layer polynomials and multi-layer perceptrons) for nonlinear modeling,"* Journal of Machine Learning Research, vol. 03, pp. 1383–1398, (2003).

[15] R. D. H. Stoppiglia, G. Dreyfus and Y. Oussar, *"Ranking a random feature for variable and feature selection,"* Journal of Machine Learning Research, vol. 03, pp. 1399–1414, (2003).

[16] M. A. Hall and L. A. . Smith., *"Feature subset selection : a correlation based filter approach,"* International Conference on Neural Information, vol. 1,No 4, pp. 855–858, (1997).

[17] L. Y. et H. Liu, *"Feature selection for high-dimensional data : a fast correlation based filter solution,"* International Conference on Machine Leaning, vol. 01,No 12, pp. 856–863, (2003).

[18] Sancho Salcedo-Sanz, Gustavo Camps-Valls and C. Bousono-Calzon, *"Enhancing genetic feature selection through restricted search and walsh analysis,"* IEEE transactions on systems, man, and cybernetics : applications and reviews, vol. 34, no 4, (2004).

[19] Christophe Nicolas Mangnan, *"Apprentissage a partir de données diversement étiquetées pour l'étude de rôle de l'environnement local dans les interaction entre acides amines"*. Thèse doctorat, (2007).

[22] Le Cun and S. Solla, *"Optimal brain damage,"* Advances in Neural Information Processing Systems, vol. 02, pp. 598–605, (1990).

[23] V. N. Vapnik, "The nature of statistical learning theory," Springer Verlag, vol. 01, 1995.

[24] Cherit, M., Kharma, N., Cheng-Lin, L., &Suen, C. *"Character recognition system a guide for students and practitioners"*. John Wiley., (2007).

[25] Marill, T. et Green, D. M. *"On the effectiveness of receptors in recognition systems."* IEEE transactions on Information Theory, 9:11-17., (1963)

[26] Whitney, A. W.. *"A direct method of nonparametric measurement selection"*. IEEE Trans. Comput., 20:1100-1103., (1971)

[27] Aha, D. W. et Bankert, R. L.. *"A comparative evaluation of sequential feature selection algorithms"*. In 5th International Workshop on Artificial Intelligence and Statistics, pages 1-7. Ft. Lauderdale, USA., (1995)

[28] Kittler, J.. *"Feature set search algorithms. Pattern Recognition and Signal Processing"*, pages 41-60., (1978)

[29] Pudil, P., Novovicova, J. et Kittler, J. *"Floating search methods in feature selection. Pattern Recogn. Lett"*., 15:1119-1125., (1994)

[30] Narendra, P. M. et Fukunaga, K.). *"A branch and bound algorithm for feature subset selection"*. IEEE Trans. Comput., 26:917-922., (1977)

[31] Chen, X.-w.. *"An improved branch and bound algorithm for feature selection. Pattern Recognition Letters"*, 24(12):1925 - 1933. ,(2003)

[32] Somol, P., Pudil, P. et Kittler, J. *"Fast branch & bound algorithms for optimal feature selection"*. IEEE Pattern Analysis and Machine Intelligence, 26:900-912., (2004)

[33] Almuallim, H. et Dietterich, T. G., *"Learning with many irrelevant features"*. In Proceedings of the Ninth National Conference on Artificial Intelligence, pages 547-552. AAAI Press., (1991)

[34] Almuallim, H. et Dietterich, T. G.). *"Efficient algorithms for identifying relevant features"*. In In Proceedings of the Ninth Canadian Conference on Articial Intelligence, pages 38-45. Morgan Kaufmann., (1992)

[35] Kira, K. et Rendell, L. A. *"The feature selection problem : Traditional methods and a new algorithm"*. In AAAI, pages 129-134, Cambridge, MA, USA. AAAI Press and MIT Press., (1992).

[36] Liu, H. et Setiono, R. *"Feature selection and classification probabilistic wrapper approach"*. In in Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES, pages 419-424., (1996)

[37] Kachouri, R., Djemal, K. et Maaref, H. *"Adaptive feature selection for heterogeneous image databases"*. In Djemal, K. et Deriche, M., éditeurs : Second IEEE International Conference on Image Processing Theory, Tools 38 ; Applications, 10, Paris, France. , (2010)

[38] Peng, H., Long, F. et Ding, C. *"Feature selection based on mutual information : criteria of max-dependency, max-relevance, and min-redundancy"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:1226-1238., . (2005).

[39] Mari, J., & Napoli, A. *"Aspects de la classification"*. Rapport technique 2909, INRIA., (1996).

[40] Henriet, L. *"Système d'évaluation et de classification multicritères pour l'aide à la décision, construction de modles et procédures d'affectation"*. Thèse de doctorat en science. Université Paris Dauphine., (2000).

[41] Michie, D., Spiegelhalter, D., & C.C. *"Machine learning, neural and statistical classification"*. New York: Ellis Horwood., (1994)

[42] Cormack, R. *"A review of Classification"*. Journal of the Royal Statistical Society, A(134), pp. 321-367., (1971).

[43] Gordon, G., Jensen, R., Hsiao, L., Gullans, S., Blumenstock, J., & S. Ramaswamy, W. R. *"Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma"*. Cancer Research, 6, pp. 4963–4967., (2002).

[44] Hansen, P., & Jaumard, B. *"Cluster analysis and mathematical programming. Mathematic Programming"*, 79, pp. 191-215., (1997).

[45] Magnan C . "*Apprentissage à partir des données diversement étiquetées",* Thèse de doctorat., . (2007)

[46] Vapnik, V. *"Statistical learning theory"*. New York: Wiley., (1998).

[47] Weiss, S., &Kulikowski, C. *Computer systems that learn, classification ans prediction methods from statistics, neural nets, machine learning and experts systems.* San Mateo: California Morgan Kaufman Publishers., (1991).

[48] Indyk, P. et Motwani, R. *"Approximate nearest neighbors : Towards removing the curse of dimensionality"*. pages 604{613., (1998).

[49] Burges, C. J. C. *"A tutorial on support vector machines for pattern recognition"*. Data Min. Knowl. Discov., 2:121-167., (1998).

[50] J. Doak. *"An evaluation of feature selection methods and their application to computer security"*. Technical report, Davis CA: University of California, Department of Computer Science, (1992).

[51] R. Battiti. *"Using mutual information for selecting features in supervised neural net learning"*. IEEE Transactions on Neural Networks, 5(4):537–550, (1994).

[52] F. Fleuret. *"Fast binary feature selection with conditional mutual information"*. Journal of Machine Learning Research, 5:1531–1555, (2004).

[53] L. Yu and H. Liu. *"Efficient feature selection via analysis of relevance and redundancy"*. Journal of Machine Learning Research, 5:1205–1224, (2004).

[54] N Kwak and C.H Choi, *"Input Feature Selection for Classification Problems,"* IEEE Transactions on Neural Networks, vol. 13, no. 1, pp. 143–159,(January 2002)

[55] Bollacker, K.D., Ghosh, J. "*Mutual Information Feature Extractors for Neural Classifiers"*. IEEE International Conference on Neural Networks. 3, 1528–1533, (1996)

[56] T.M Cover and J.A Thomas, *"Elements of information theory"*, 2nd ed.: Wiley Series in telecommunications and Signal Processing, (2006).

[57] C.E. Shannon, *"A mathematical theory of communication,"* Bell Systems Technical Journal, vol. 27, pp. 379-423, July (1948).

[58] M. Vidal-Naquet and S. Ullman, *"Object recognition with informative features and linear classification"*. IEEE on Computer Vision and Pattern Recognition, vol. 1, Nice, France, pp. 281-288., (2003)

[59] D. Lin and X. Tang, *"Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion,"* in Proc. European Conference on Computer Vision, vol. Part I, Graz , Autriche, pp. 68-82., (May 2006).

[60] P. E. Meyer, C. Schretter, and G. Bontempi. *"Information-theoretic feature selection in microarray data using variable complementarity"*. IEEE Journal of Selected Topics in Signal Processing, 2(3): 261–274, (2008).

[61] D. D. Lewis. "*Feature selection and feature extraction for text categorization*". In Proceedings of the workshop on Speech and Natural Language, pages 212–217. Association for Computational Linguistics Morristown, NJ, USA, (1992).

[62] W. Duch. *"Feature Extraction: Foundations and Applications"*, chapter 3, pages 89–117. Studies in Fuzziness & Soft Computing. Springer, ISBN 3-540-35487-5.,(2006.)

[63] Dietz, W. E., Kiech, E. L. et Ali, M. *"Classification of data patterns using and auto associative neural network topology"*. In Proceedings of the 2nd international conference on Industrial and engineering applications of artificial intelligence and expert systems - Volume 2, pages 1028{1036, New York, NY, USA. ACM., (1989).

[64] C. L. Blake and C. J. Merz, *"UCI Repository of machine learning databases,"* Irvine, CA: University of California, Department of Information and Computer Science, (1998).

ANNEXES

ANNEXE 1 : Détail des bases de données utilisées

1- Base de données "Iris"

Iris Plant Database disponible sur le site de l'UCI .

A l'origine de cette base de données, les travaux de R.A. Fisher "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936).

Ces données sont souvent utilisées en classification. Il y a 3 classes d'Iris à découvrir : Iris Setosa, Iris Versicolour et Iris Virginica. La base de données contient 150 instances réparties à égalité dans chaque classe (50 par classe).

Il y a quatre attributs numériques : sepal length (longueur du sépal) en cm, sepal width (largeur du sépal) en cm, petal length (longueur du pétal) en cm et petal width (largeur du pétal) en cm.

Le tableau D.1 donne des indications sur les données de cette base.

	Min	Max	Mean	SD Class	Correlation
Sepal length	4.3	7.9	5.84	0.83	0.7826
sepal width	2.0	4.4	3.05	0.43	-0.4194
petal length	1.0	6.9	3.76	1.76	0.9490
petal width	0.1	2.5	1.20	0.76	0.9565

TAB. D.1 – Statistiques descriptives de la base Iris

2- Base de données PIMA " Dibètes"

La base de données Pima Indians Diabetes Database provient du site de l'UCI et a été constituée par l'Institut national des maladies du diabète, de la digestion et du foie. L'objectif est de réaliser un diagnostic du diabète sur une population vivant près de Phoenix, Arizona aux USA.

Il y a 768 patients (donc instances) qui sont toutes des femmes de plus de 21 ans et provenant toutes des tribus indiennes Pima. Les huit attributs numériques observés sont :

1. Nombre de grossesses
2. Concentration du plasma en glucose après un test de tolérance au glucose de 2 heures
3. Pression diastolique du sang (mm Hg)
4. Epaisseur de la peau au niveau du triceps (mm)
5. Taux d'insuline au bout de 2 heures (mu U/ml)
6. Body mass index (poids en kg/(taille in m))
7. Fonction pédigrée du diabète

8. Age (années)

Le tableau D.2 donne des indications sur les données de cette base.

	Mean	Standard Deviation
1.	3.8	3.4
2.	120.9	32.0
3.	69.1	19.4
4.	20.5	16.0
5.	79.8	115.2
6.	32.0	7.9
7.	0.5	0.3
8.	33.2	11.8

TAB. D.2 – Statistiques descriptives de la base Diabete.

3- Base de données " Breast Cancer "

Cette base de données relative au cancer du sein a été mise à disposition par l'université du Wisconsin par Dr. William H. Wolberg.

O.L. Mangasarian and W. H.Wolberg : "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 – 18.

Le jeu de données est composé de 699 instances pour 10 attributs.

Num.	Attribut	Domaine
1	Sample code number	id number
2	Clump Thickness	1 - 10
3	Uniformity of Cell Size	1 - 10
4	Uniformity of Cell Shape	1 - 10
5	Marginal Adhesion	1 - 10
6	Single Epithelial Cell Size	1 - 10
7	Bare Nuclei	1 - 10
8	Bland Chromatin	1 - 10
9	Normal Nucleoli	1 - 10
10	Mitoses	1 - 10

TAB. D.3 – Présentation des attributs et de leur domaine pour le jeu de données Breast cancer.

4- Base de données "Lung cancer"

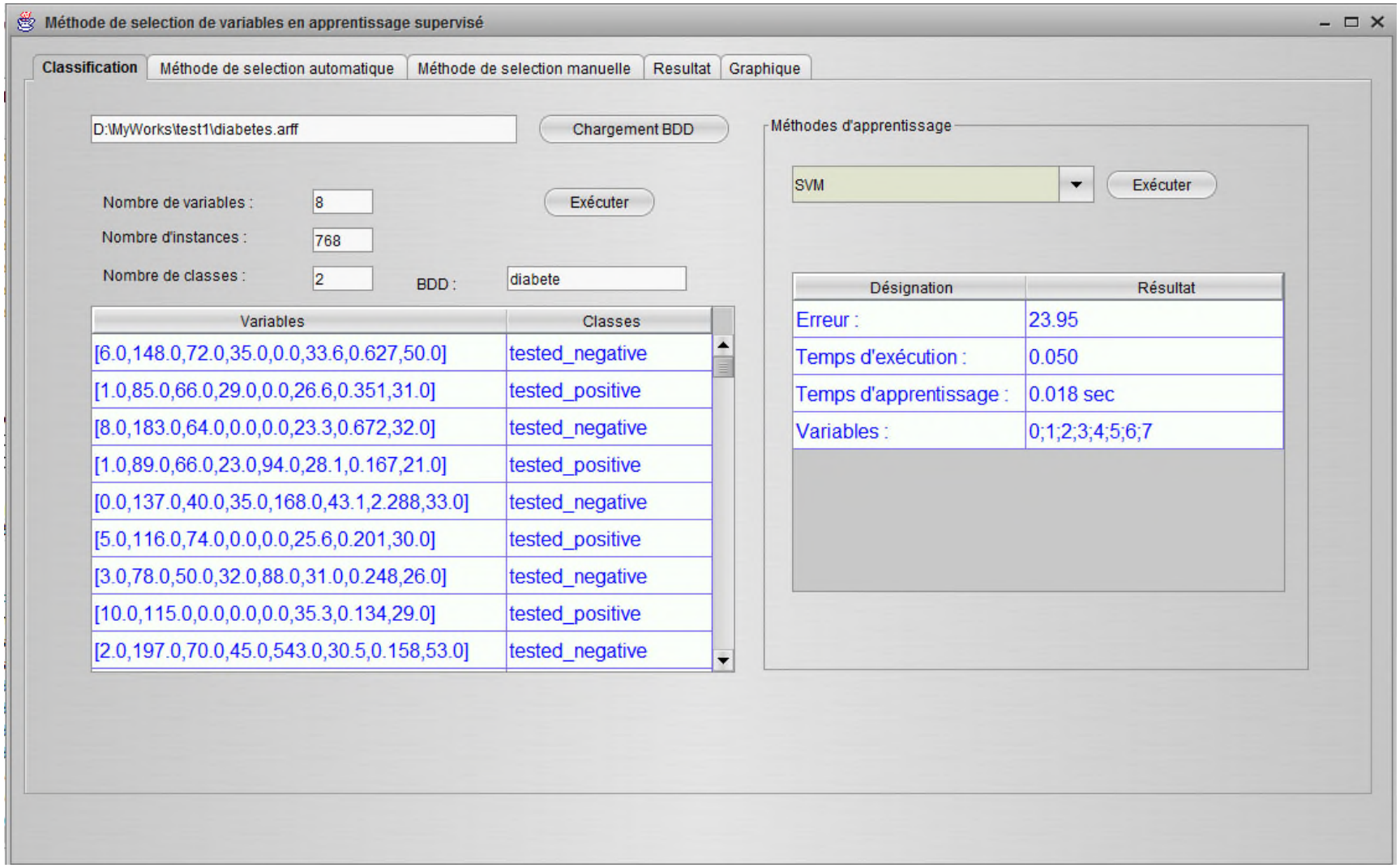
Le jeu de données “Lung cancer” (cancer du poumon) est disponible sur le web et a été publié dans Hong, Z.Q. and Yang, J.Y. "Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane", Pattern Recognition, Vol. 24, No. 4, pp. 317-324,1991.

Ce jeu de données décrit trois types de pathologie du cancer du foie. Il contient des données manquantes pour deux des attributs. Le jeu contient 32 instances pour 56 attributs descriptifs. Tous les attributs sont nominaux et prennent leur valeur parmi les entiers de 0 à 3

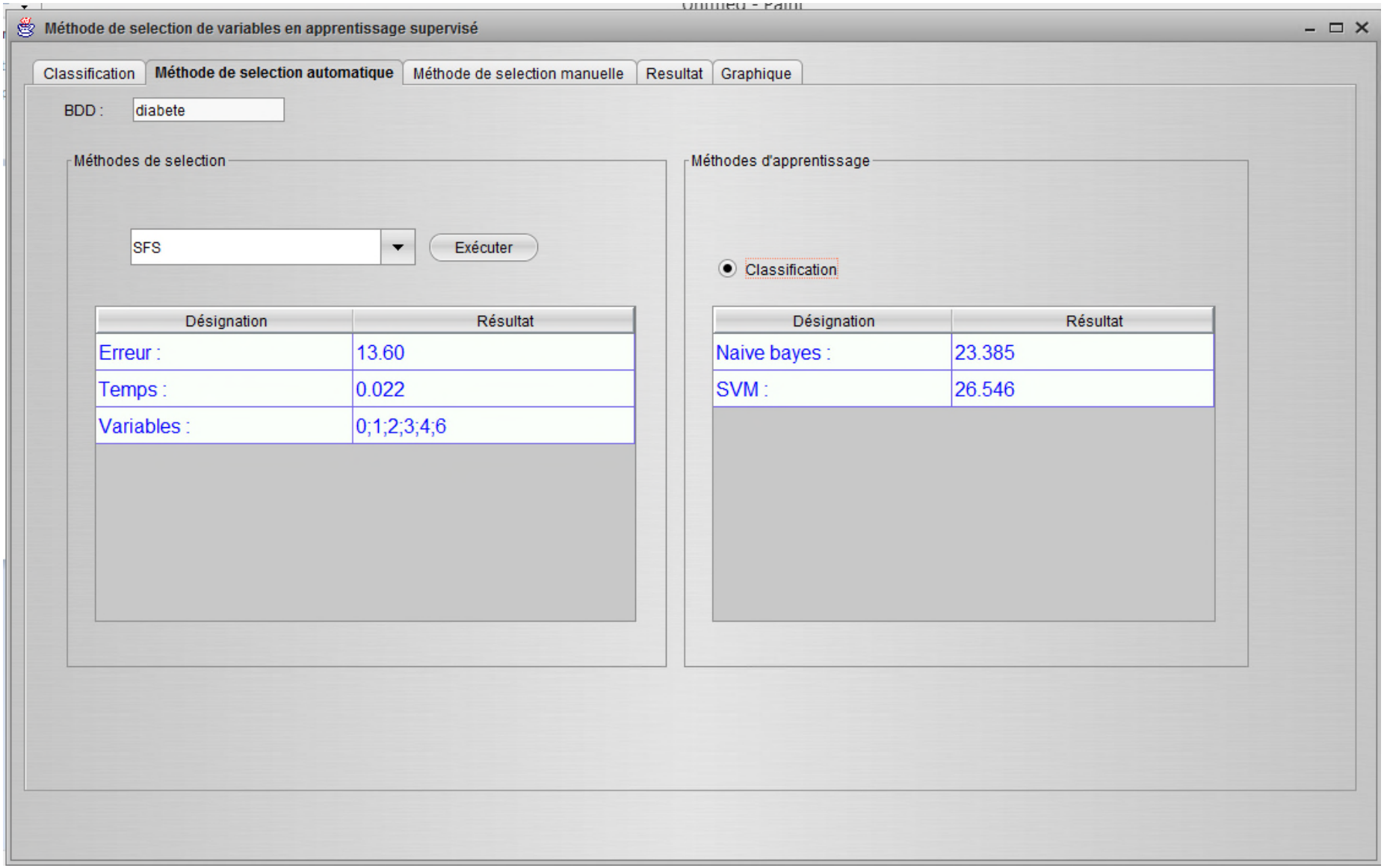
5- Base de données Leukemia (Leucémie) : Ce jeu de données est constitué de 72 échantillons représentant deux types de Leucémie aigüe. 47 tissus sont du type leucémie lymphoblastique aiguë (ALL) et 25 sont du type Leucémie myéloïde aiguë (AML). Pour chaque échantillon, les niveaux d’expression de 7129 gènes ont été relevés (Golub, et al., 1999).

ANNEXE 2: Quelques masques écrans de l'application.

Masque 01: "Résultats l'application des algorithmes de classification SVM sur le Dataset Pima Diabetes"



<u>Masque 02 :</u> "Résultats de la méthodes de Sélection de variables SFS avec les classifieurs SVM et NB"



<u>Masque 03:</u> "Résultats de la méthodes de Sélection de variables Relief avec les classifieurs SVM et NB"