TABLE DES MATIERES

NOTATIONS ET ABREVIATIONS

1. Minuscules latines

a	Vrais positifs
b	Vrais négatifs
c	Faux positifs
c	Classe d'un exemple
d	Faux négatifs
d	Distance
err	Nombre d'exemples mal classés
f	Function d'évaluation de l'activation d'une stimulation reçue
g	Function de la regression logistique
k	Nombres d'échantillons d'apprentissage
$\hat{m}$	Estimateur construit à un étape
p	Position par rapport à un arbre t
x_1	Variable d'entrée
x_2	Variable d'entrée
X_n	Ensemble de signaux d'entrées

2. Majuscules latines

B	Nombre d'estimateurs à agréger
C_{da}	Matrice de confusion de l'analyse discriminante
C_{glm}	Matrice de confusion de la regression logistique
C_{knn}	Matrice de confusion du classifieur plus proche voisins
C_{nb}	Matrice de confusion du classifieur Naîve Bayes
C_{nn}	Matrice de confusion du réseau de neurone
C_{svm}	Matrice de confusion du classifieur Machines à vecteurs de supports
C_t	Matrice de confusion de l'arbre de décisions
C_{tb}	Matrice de confusion du classifieur TreeBagger
C_{Bayes}	Classifieur naîf de Bayes
$\mathbf{C_k}$	Ensemble de procedures de classification
$\mathbf{C_{opt}}$	Une procedure optimale

D	Distance euclidienne
$\boldsymbol{D_n}$	Un n échantillon
E	Fonction d'évaluation d'une stimulation reçue
N	Cardinal d'un ensemble
S	Un exemple
$\boldsymbol{W_{ij}}$	Ensemble de signaux i correspondants aux signaux d'entrées j
X	Variable d'entrée
Y	Variable de sortie

3. Minuscules grecs

$\boldsymbol{\alpha}$	Paramètre d'estimation sur les données d'apprentissages
$\boldsymbol{\beta}$	Parameter à estimer pour la regression linéaire simple
$\boldsymbol{\varepsilon}$	Erreur aléatoire
$\boldsymbol{\theta}$	Seuil du neurone de sortie
$\boldsymbol{\theta_k}$	Échantillon de bootstrap
$\boldsymbol{\pi}$	Probabilité d'observer un événement
$\boldsymbol{\Omega}$	Un échantillon

4. Majuscules grecs

$\boldsymbol{\Delta}$	Fonction interragissant sur l'erreur théorique

5. Abréviations

AdaBoost	Adaptative Boosting
ADN	Acide DésoxyriboNucléique
AUC	Air Under Curve
CRISP-DM	Cross-Industry Standard Process for Data Mining
Eapp	Erreur apparente
ECD	Extraction de Connaissances à partir de Données
EM	Expectation Maximization
ET	Erreur Théorique
FN	Faux Négatifs
FP	Faux Positifs
IHM	Interface Home Machine

IT	Information Technology
NoSQL	Not only Structured Query Language
kNN	K Nearest Neighbors
kppV	K plus proches Voisins
OCR	Optical Character Recognition
PMML	Predictive Model Markup Language
RFP	Ration Faux Positif
RNA	Ribonucleic Acid
ROC	Receiver Operating Characteristic
RVP	Ratio Vrai Positif ou sensitivité
SAP	Systems Applications and Products
SAS	Statistical Analysis System
SEM	Search Engine Marketing
SEMMA	Sample Explore Modify Model and Assess
SPSS	Statistical Package for Social Sciences
SQL	Structured Query Language
SVM	Support Vector Machines
SVR	Support Vector Regression
VC	Validation Croisé
VC	Vapnik-Chervonenkis
VN	Vrais Négatifs
VP	Vrais Positifs

INTRODUCTION GENERALE

Aujourd'hui, de nombreux domaine ne cessent d'évoluer avec la technologie. Les exemples suivantes y font parties : la recherche en Intelligence Artificielle et en théorie de l'apprentissage, les capacités de stockage et de calcul offertes par le matériel et les techniques informatiques modernes, la constitution de giga-bases de données pour les besoins de gestion des entreprises, les logiciels universels, puissants et conviviaux et ainsi que l'intégration du data mining dans les processus de production. Toutes ses phénomènes permettent de traiter de grands volumes de données et font sortir le data mining des laboratoires de recherche pour entrer dans les entreprises.

Le Data mining est un domaine pluridisciplinaire permettant, à partir d'une très importante quantité de données brutes, d'en extraire de façon automatique ou semi-automatique des informations cachées, pertinentes et inconnues auparavant en vue d'une utilisation industrielle ou opérationnelle de ce savoir. Il peut également mettre en avant les associations et les tendances et donc servir d'outil de prévisions au service de l'organe décisionnel.

On distingue le data mining supervisé qui sert essentiellement à la classification des données et le data mining non supervisé qui est utilisé dans la recherche d'associations ou de groupes d'individus. Le champ d'action du data mining s'étend du chargement et du nettoyage des données dans les bases de données, à la mise en forme des résultats, en passant par le plus important : la classification et la mise en relation de différentes données.

Ainsi dit, l'objectif de ce mémoire de fin d'études intitulé : « Implémentation et performance d'un système optimisé d'analyse de donnée par le data mining supervisé» est de comprendre le fonctionnement de l'extraction et fouille de donnée sur un gros volume de données, mais aussi de connaître quelle méthode est efficace vis-à-vis des différentes algorithmes d'apprentissage supervisée utilisés face à un donnée spécifique, ainsi que son préalable optimisation.

Pour cela, notre travail va se regrouper en quatre chapitres dont le premier se focalise sur l'extraction de connaissances. Ensuite, le second chapitre parlera de l'apprentissage automatique ou machine learning. Et après, on abordera par le troisième chapitre, en explicitant le principe de la modélisation des règles et patterns dans un système d'apprentissage supervisée pour la prédiction. Enfin, le dernier chapitre sera consacré à la partie simulation sous Matlab en analysant un type de donnée résultant d'une compagne marketing d'une banque ou marketing directe.

CHAPITRE 1
EXTRACTION DE CONNAISSANCES

1.1 Introduction

L'extraction de connaissances et data mining consiste à donner un sens aux grandes quantités de données, d'un certain domaine, capturées et stockées massivement par les entreprises d'aujourd'hui. En effet, la vraie valeur n'est pas dans l'acquisition et le stockage des données, mais plutôt dans notre capacité d'en extraire des rapports utiles et de trouver des tendances et des corrélations intéressantes pour appuyer les décisions faites par les décideurs d'entreprises et par les scientifiques. Cette extraction fait appel à une panoplie de techniques, méthodes, algorithmes et outils d'origines statistiques, intelligence artificielle, bases de données, etc.

Cependant, l'activité de l'extraction de connaissances et data mining a été rapidement organisée sous forme d'un processus appelé processus d'ECD (Extraction de Connaissances à partir de Données). Ce chapitre consiste à l'explication de l'extraction de connaissances et de data mining. Pour ce faire, on va suivre le plan suivant : premièrement, on va définir l'ECD, expliquer les étapes du processus et ses tâches. Puis on va présenter le format des données d'entrées dans ce processus ainsi que l'entrepôt où l'on stocke ces données avant de lancer le processus. Après on parlera du data mining qui est l'étape le plus importante dans le processus d'extraction de connaissances dans la base de données. Et on terminera ce chapitre par une brève conclusion.

1.2 Définition de l'Extraction de Connaissances

L'ECD (Extraction de Connaissances à partir de Données) est un processus pour la découverte de nouvelles connaissances sur un domaine d'application donné.

L'ECD est également défini comme étant un processus non trivial qui permet d'identifier, dans des données, des patterns ultimement compréhensibles, valides, nouveaux et potentiellement utiles. [1]

1.3 Le processus d'extraction de connaissances

Le processus d'ECD vise à transformer des données (volumineuses, multiformes, stockées sous différents formats sur des supports pouvant être distribués) en connaissances. Ces connaissances peuvent s'exprimer sous forme de concepts généraux qui enrichissent le champ sémantique de l'usager par rapport à une question qui le préoccupe. Elles peuvent prendre la forme d'un rapport ou d'un graphique. Elles peuvent s'exprimer comme un modèle mathématique ou logique pour la

prise de décision. Les connaissances extraites doivent être les plus intelligibles possibles pour l'utilisateur. Elles doivent être validées, mises en forme et agencées. [1] [2]

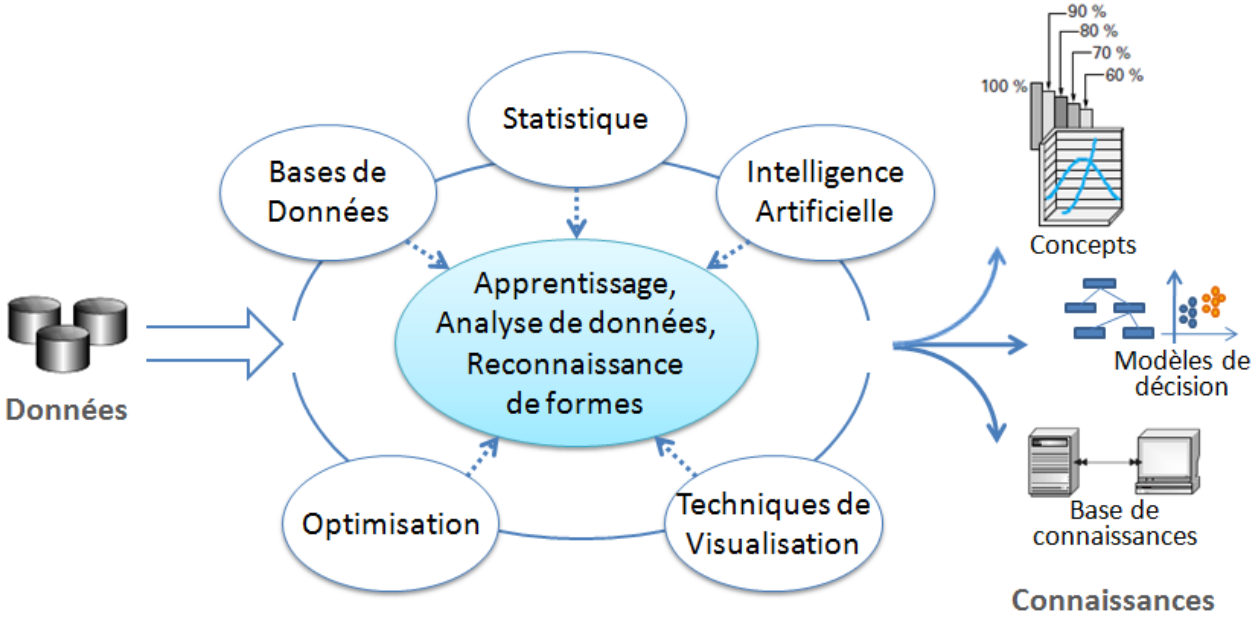


Figure 1.01 : *Techniques et domaines en relation avec le processus d'ECD*

Le processus d'ECD s'effectue sur plusieurs étapes interrompues continuellement par des prises de décision par l'utilisateur expert. Il nécessite sommairement la préparation des données, la recherche de patterns et l'évaluation des connaissances extraites et leur raffinement, toutes répétées dans plusieurs itérations. [1]

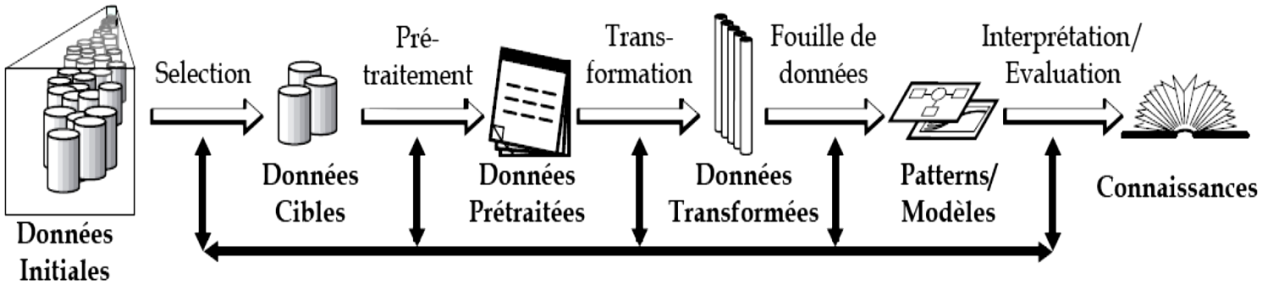


Figure 1.02 : *Etapes du processus d'ECD*

Ces étapes se résument comme suit.

1.3.1 Compréhension du domaine d'application

Ceci consiste à développer une compréhension du domaine d'application et des connaissances pertinentes préalables. Cette étape prépare l'analyste pour comprendre et définir les objectifs opérationnels du processus d'ECD du point de vue des utilisateurs immédiats de ses résultats.

1.3.2 Création d'un jeu de données cibles

L'analyste doit sélectionner les données à utiliser et les attributs pertinents pour la tâche de fouille de données.

1.3.3 Nettoyage des données et prétraitement

Cette étape vise la préparation d'un jeu de données « propre » et bien structuré. Elle comprend des opérations de base telles que : l'élimination des données bruyantes, recueil des informations nécessaires pour modéliser et tenir compte du bruit, choix des stratégies de traitement des valeurs manquantes, ainsi que de décider des questions sur la base de données à utiliser.

1.3.4 Réduction et projection des données

Il s'agit de trouver des attributs utiles pour représenter les données en fonction de l'objectif de la tâche d'extraction, et d'utiliser des méthodes de réduction de dimensionnalité ou de transformation afin de réduire le nombre effectif de variables d'étude et dégager de nouvelles variables plus pertinentes. Cette étape est très importante pour la réussite du projet d'ECD et doit être adaptée en fonction de la base de données et des objectifs opérationnels du projet.

1.3.5 Choix de la tâche de fouille

Cette étape consiste à faire correspondre les objectifs opérationnels du processus d'ECD à une tâche particulière de fouille de données, comme la classification, la régression, le clustering, ou la description et synthèse de données.

1.3.6 Choix des algorithmes de fouille de données appropriés

Il consiste à sélectionner les méthodes à utiliser pour la cherche de patterns dans les données, décider quels sont les modèles et paramètres appropriés, et conclure par le choix d'une méthode particulière de fouille de données en accord avec le critère global du processus d'ECD.

1.3.7 Fouille de données ou data mining

Il s'agit d'exécuter la ou les méthodes choisies avec leurs paramètres afin d'extraire des patterns d'intérêt sous une forme de représentation particulière. Par exemple des règles ou arbres de

classification, des modèles de régression, des clusters, et autres. Parfois, il sera nécessaire d'appliquer la méthode de fouille plusieurs fois pour obtenir le résultat escompté.

1.3.8 Interprétation des patterns extraits

Cette étape comprend l'évaluation et l'interprétation des modèles découverts dans les données. Il peut être nécessaire de retourner à l'une des étapes 1 à 7 pour des itérations éventuelles. Cette étape donne l'occasion de revenir sur les étapes précédentes, mais aussi d'avoir une représentation visuelle des patterns, de supprimer les patterns redondants ou non représentatifs et de transformer le résultat en informations compréhensibles par l'utilisateur final.

1.3.9 Consolidation des connaissances extraites

C'est l'étape de consolidation des connaissances extraites en utilisant directement ces connaissances, en les incorporant dans d'autres systèmes pour des actions ultérieures, ou simplement en les documentant et les rapportant aux utilisateurs concernés. Ceci inclue également la détection et la résolution de tout conflit potentiel avec d'autres connaissances déjà confirmées ou extraites.

1.4 Les différentes tâches de l'Extraction de Connaissances

La tâche représente le but, ou l'objectif, d'un processus d'ECD. On distingue dans la pratique deux tâches primaires de haut niveau : la prédiction et la description. La prédiction consiste à utiliser des variables ou des champs dans la base de données pour prédire des valeurs futures ou inconnues d'autres variables d'intérêt. Alors que la description se concentre sur la recherche de patterns (modèles, schémas ou règles) décrivant les données et interprétables par l'utilisateur. Bien que les limites entre la prédiction et la description ne soient pas nettes, la distinction entre ces deux tâches est utile pour la compréhension de l'objectif global du processus d'ECD. [3]

Les tâches de prédiction et de description peuvent être réalisées en utilisant une grande variété de méthodes de fouille de données à savoir :

- la segmentation ou clustering,
- la classification,
- la régression,
- analyse de dépendances,
- etc.

1.5 Les données d'entrées

Les données qui font l'objet de tâches de fouilles se présentent suivant différents formats. Nous en distinguerons trois principaux : les tableaux utilisés en fouille de données, les textes bruts et les documents semi-structurés. [2]

1.5.1 Les tableaux

Commençons donc par les tableaux exploités en fouille de données. Cette discipline est née notamment dans les milieux des banques, des assurances et de la médecine, domaines qui ont intégré depuis longtemps l'usage des bases de données informatiques. Dans un tableau de données, chaque instance est décrite par un certain nombre d'attributs typés (ou de champs). Les différents types possibles des attributs sont les types élémentaires traditionnels de l'informatique : booléen, caractère, nombre, chaîne de caractères, valeur prise dans une liste finie… La valeur prise par un attribut peut être obligatoire ou facultative.

Notons dès à présent qu'il existe différentes terminologies pour décrire les éléments de telles bases de données tabulaires. Le plus souvent, les lignes du tableau sont appelées les exemples, et les colonnes les attributs. On peut également considérer qu'il s'agit d'objets décrits par des valeurs sur plusieurs dimensions, ou bien de points décrits par leur cordonnées. [3]

Identifiant	**Carburant**	**Cylindres**	**Longueur (cm)**	**Puissance (Chevaux)**	**Prix (Euro)**
1	Diesel	8	186	6000	16000
2	Essence	4	170	5800	9000
3	Diesel	6	172	?	12000
4	Diesel	4	156	5200	6500
5	Essence	12	190	5500	19000
6	?	4	175	5800	9500
7	Ethanol	4	158	6000	8000
8	Diesel	6	188	5200	18000
9	Essence	4	**1680**	5000	7500
10	Diesel	6	170	6000	10500

Tableau 1.01: *Données bruitées et contenant des valeurs manquantes*

Notons aussi que dans certains cas, les données peuvent être bruitées, c'est-à-dire que certaines de leurs valeurs sont aberrantes. Il s'agit le plus souvent de valeurs issues de capteurs déficients, ou bien d'erreurs humaines dans leur manipulation. Il peut aussi arriver que certaines valeurs ne soient pas renseignées. On parle dans ce cas de données manquantes.

Le **Tableau 1.01 :** présente un exemple de données dans lesquelles deux valeurs manquantes (**?**) et une valeur aberrantes (en gras) sont présentés.

Les algorithmes ne sont pas tous égaux devant les données : certains requièrent des tableaux entièrement remplis, d'autres s'arrangent très bien à des valeurs manquantes. Certains, et ce sont en général les plus efficaces, ne savent manipuler que des tableaux complets de nombres. Une donnée uniquement décrite par une liste de nombres peut en effet facilement être assimilée à un point dans un espace vectoriel ou, ce qui revient au même, à un vecteur dont on fournit les coordonnées. Traditionnellement, les données sont disposées en lignes (une donnée par ligne), les attributs en colonnes. L'ordre des lignes et des colonnes n'a aucune importance, au sens où on ne modifiera en rien le résultat des algorithmes de fouille qui y seront appliqués. Seule la dernière colonne joue, pour certaines tâches, un rôle particulier. [3]

1.5.2 Textes bruts

Les textes, même numérisés, ne présentent pas du tout les mêmes propriétés que les tableaux de données. En termes de structures, ils semblent même situés à l'opposé du « spectre » : autant les tableaux ont un haut degré d'organisation, autant les textes sont apparemment faiblement structurés. Et ceci d'autant plus qu'en fouille de textes, on ne s'intéressera principalement qu'à des textes bruts, c'est-à-dire de simples séquences de caractères d'où toute mise en forme est absente. Tout ce qui ne vise qu'à la visualisation (police et taille des caractères, mises en gras ou en italique, alignement de la page, sauts de lignes, etc.) ou à la structuration d'un document (en parties, sous-parties et paragraphes, en listes et énumérations etc.) et constitue la raison d'être des traitements de textes est en effet dans ce cas complétement ignoré. Un texte brut est un simple fichier au format « .txt », uniquement constitué de caractères pris parmi un ensemble fini, codés suivant une certaine norme. Les caractères sont les atomes indivisibles du fichier ; ils sont dits alphanumériques car ils intègrent aussi bien les lettres de l'alphabet et les symboles numériques et mathématiques que tous ceux pouvant être tapés sur un clavier d'ordinateur : ponctuations, symboles monétaires, etc... Toutes les unités d'écriture des langues non alphabétiques (idéogrammes) sont aussi considérées comme des caractères indivisibles, si le codage adopté les accepte comme tels. Ainsi, dans un texte brut, la seule structure présente est l'ordre linéaire dans lequel les caractères apparaissent. [4]

1.5.3 Documents semi-structurés

Le troisième format possible pour les données d'entrée d'un programme de fouille de données est intermédiaire entre les précédents : il est plus structuré qu'un texte brut, mais moins qu'un tableau,

et on l'appelle parfois pour cela « semi-structuré » : c'est celui des documents XML (Extensible Markup Language). En fait, rien n'empêche de traiter un document en XML exactement de la même façon qu'un texte brut : il suffit pour cela d'admettre que les éléments propres au langage utilisé (principalement les balises ouvrantes et fermantes) soient considérés comme des caractères indivisibles supplémentaires, qui s'ajoutent aux autres. La **Figure 1.03** : montre un morceau de code HTML (HyperText Markup Language), qui peut être considéré comme un cas particulier de langage XML tel qu'il apparaît dans un éditeur de texte. Le prétraitement consistant à identifier les balises est trivial ; ce code peut ainsi être considéré comme un texte brut écrit dans un nouvel alphabet : celui contenant tous les caractères alphanumériques ainsi que les balises considérées comme des unités indivisibles. [2] [3]

```
<table>
<tr><th>produit</th><th>marque</th><th>prix en euros</th></tr>
<tr><td>ordinateur portable</td><td>truc</td><td>800</td></tr>
<tr><td>tablette</td><td>machin</td><td>200</td></tr>
</table>
```

Figure 1.03 : *Code HTML*

1.6 Entrepôt de données

Plus précisément, le contexte informationnel du Data Mining est celui des Data Warehouses. Un entrepôt de données, dont la mise en place est assurée par un gestionnaire de données, est un ensemble de bases relationnelles ou cubes multidimensionnels alimenté par des données brutes et relatif à une problématique :

- gestion des stocks, prévision des ventes afin d'anticiper au mieux les tendances du marché ;
- suivi des fichiers clients d'une banque, d'une assurance, associés à des données socio-économiques, à l'annuaire, en vue de la constitution d'une segmentation (typologie) pour cibler des opérations de marketing ou des attributions de crédit. La gestion de la relation client vise à une individualisation ou personnalisation de la production et de la communication afin d'évacuer la notion de client moyen jugée trop globalisante ;
- recherche, spécification, puis ciblage des niches de marché les plus profitables ou au contraire les plus risquées (assurance) ;

- suivi en ligne des paramètres de production en contrôle de qualité pour détecter au plus vite l'origine d'une défaillance ;
- prospection textuelle (Text Mining) ;
- Web Mining ;
- décryptage d'une image astrophysique, du génome ;
- etc…

Un entrepôt de données se caractérise par un environnement informatique hétérogène pouvant faire intervenir des sites distants à travers le réseau de l'entreprise (intranet) ou même des accès extérieurs (internet). En effet, des contraintes d'efficacité (suivi en temps réel), de fiabilité ou de sécurité conduisent à répartir et stocker l'information à la source plutôt qu'à la dupliquer systématiquement ou à la centraliser. [4]

1.7 Data Mining

1.7.1 Définition 1

Le terme de Data Mining signifie littéralement forage de données. Comme dans tout forage, son but est de pouvoir extraire un élément: la connaissance. Ces concepts s'appuient sur le constat qu'il existe au sein de chaque entreprise des informations cachées dans le gisement de données. Ils permettent, grâce à un certain nombre de techniques spécifiques, de faire apparaître des connaissances. [5]

1.7.2 Définition 2

Le Data Mining, ou la fouille de données est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de grandes bases de données informatiques, de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de données permettant d'étayer les prises de décision.

En bref, le Data Mining est l'art d'extraire des informations (ou même des connaissances) à partir des données.

Le Data Mining peut être soit descriptif, soit prédictif. [6]

1.7.2.1 Les techniques descriptives

Les techniques descriptives (ou exploratoires) visent à mettre en évidence des informations présentes mais cachées par le volume de données. [6]

1.7.2.2 Les techniques prédictives

Les techniques prédictives (ou explicatives) visent à extrapoler de nouvelles informations à partir des informations présentes (exemple : le cas de scoring). [6]

1.8 Les objectifs des méthodes de Data Mining

On peut regrouper les objectifs des méthodes de Data Mining en quatre grandes fonctions : [5]

1.8.1 Classifier

On examine les caractéristiques d'un nouvel objet pour l'affecter à une classe prédéfinie. Les classes sont bien caractérisées et on possède un fichier d'apprentissage avec des exemples pré-classés.
On construit alors une fonction qui permettra d'affecter à telle ou telle classe un nouvel individu.

1.8.2 Estimer

La classification se rapporte à des événements discrets. L'estimation, elle, porte sur des variables continues.

1.8.3 Segmenter

Il s'agit de déterminer quelles observations vont naturellement ensemble sans privilégier aucune variable. On segmente une population hétérogène en un certain nombre de sous-groupes plus homogènes (les clusters). Dans ce cas, les classes ne sont pas prédéfinies.

1.8.4 Prédire

Cette fonction est proche de la classification ou de l'estimation, mais les observations sont classées selon un comportement ou une valeur estimée futurs.
Le modèle, construit sur les données d'exemples et appliqué à de nouvelles données, permet de prédire un comportement futur.

1.9 Architecture d'un système type de Data Mining

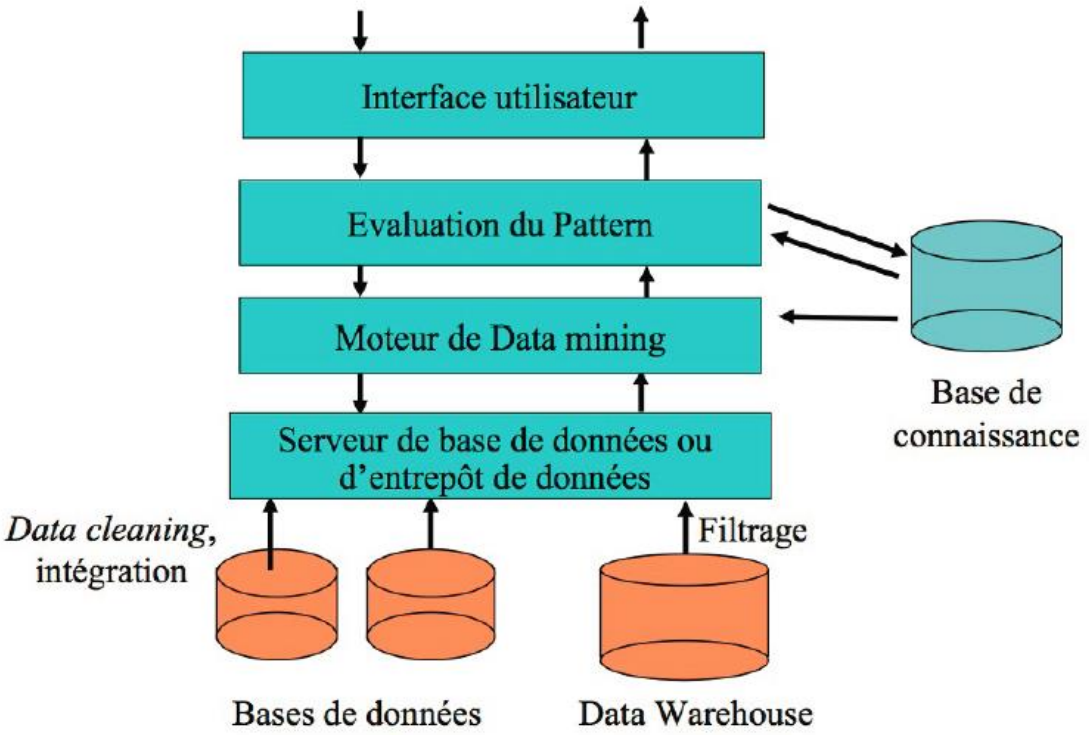


Figure 1.04 : *Architecture d'un système type de Data Mining*

1.10 Les tâches de Data Mining

Plus qu'une théorie normalisée, le Data Mining est un processus d'extraction de connaissances métiers comportant les phases principales suivantes: la description, la classification, l'association, l'estimation, la segmentation et la prévision. [3] [5]

Elles peuvent être découpées comme suit :

Techniques Descriptives			Techniques prédictives		
1. Description	2. Classification	3. Association	4. Estimation	5. Segmentation	6. Prévision

Figure 1.05 : *Les six grands types de techniques du Data Mining*

- Exemples de méthodes descriptives : la classification hiérarchique, la classification des K moyennes, les réseaux de Kohonen, les règles d'association.
- Exemples de méthodes prédictives : les méthodes de régression, les arbres de décision, les réseaux de neurones, les K plus proches voisins.

1.10.1 La description

1.10.1.1 Principe

La description consiste à mettre au jour :

- Pour une variable donnée : la répartition de ses valeurs (tri, histogramme, moyenne, minimum, maximum, etc.).
- Pour deux ou trois variables données : des liens entre les répartitions des valeurs des variables. Ces liens s'appellent des « tendances ».

1.10.1.2 Intérêt

Ceci permet de favoriser la connaissance et la compréhension des données.

1.10.1.3 Méthode

Elle utilise les méthodes graphiques pour la clarté : analyse exploratoire des données.

1.10.2 La classification

1.10.2.1 Principe

Aussi appelée clustering ou segmentation : celle-ci consiste à créer des classes (sous-ensembles) de données similaires entre elles et différentes des données d'une autre classe. L'intersection des classes entre elles doit toujours être vide. Il s'agit pour n variables de créer des sous-ensembles disjoints de données. On dit aussi «segmenter» l'ensemble entier des données. Elle définit les types de regroupement / distinction : on parle de métatypologie (type de type). Et elle permet une vision générale de l'ensemble (de la clientèle, par exemple).

1.10.2.2 Intérêt

La classification permet de favoriser, grâce à la métatypologie, la compréhension et la prédiction. Afin de fixer des segments qui serviront d'ensemble de départ pour des analyses approfondies. Et aussi de réduire les dimensions, c'est-à-dire le nombre d'attributs, quand il y en a trop au départ.

1.10.2.3 Méthodes

On peut servir de différentes méthodes telles que.

- La classification hiérarchique ;
- La classification des K moyennes ;
- Les réseaux de Kohonen ;
- Et les Règles d'association.

1.10.3 L'association

1.10.3.1 Principe

Elle consiste à trouver la corrélation entre les valeurs des variables. Les règles d'association sont de la forme : si antécédent, alors conséquence. L'association ne fixe pas de variable cible. Tous les variables peuvent à la fois être prédicteurs et variable cible. On appelle aussi ce type d'analyse une « analyse d'affinité ».

1.10.3.2 Intérêt

L'association est utilisée pour mieux connaître les comportements.

1.10.3.3 Méthode

Cette technique utilise l'algorithme a priori : Algorithme du IRG (Induction de Règles Généralisées).

1.10.4 L'estimation

1.10.4.1 Principe

L'estimation consiste à définir le lien entre un ensemble de prédicteurs et une variable cible.
Ce lien est défini à partir de données « complètes », c'est-à-dire dont les valeurs sont connues tant pour les prédicteurs que pour la variable cible. Ensuite, on peut déduire une variable cible inconnue de la connaissance des prédicteurs.
À la différence de la segmentation qui travaille sur une variable cible catégorielle, l'estimation travaille sur une variable cible numérique.

1.10.4.2 Intérêt

L'estimation permet l'estimation de valeurs inconnues.

1.10.4.3 Méthodes

Les méthodes dont on pourrait s'en servir sont :

- Analyse statistique classique : régression linéaire simple, corrélation, régression multiple, intervalle de confiance, estimation de points.
- Réseaux de neurones.

1.10.5 La segmentation

1.10.5.1 Principe

La segmentation est une estimation qui travaille sur une variable cible catégorielle. On parle de segmentation car chaque valeur possible pour la variable cible va définir un segment (ou type, ou classe, ou catégorie) de données. La segmentation peut être vue comme une classification supervisée.

1.10.5.2 Intérêt

La segmentation permet l'estimation de valeurs inconnues.

1.10.5.3 Méthode

On peut opter pour :

- Graphiques et nuages de points ;
- Méthode des k plus proches voisins ;
- Arbres de décision ;
- Réseau de neurones.

1.10.6 La prévision ou prédiction

1.10.6.1 Principe

La prévision est similaire à l'estimation et à la segmentation mise à part que pour la prévision, les résultats portent sur le futur.

1.10.6.2 Intérêt

Elle permet l'estimation de valeurs inconnues.

1.10.6.3 Méthode

Les méthodes utilisées dans cette phase sont similaires à celles de l'estimation ou de la segmentation.

1.11 Processus de Data Mining

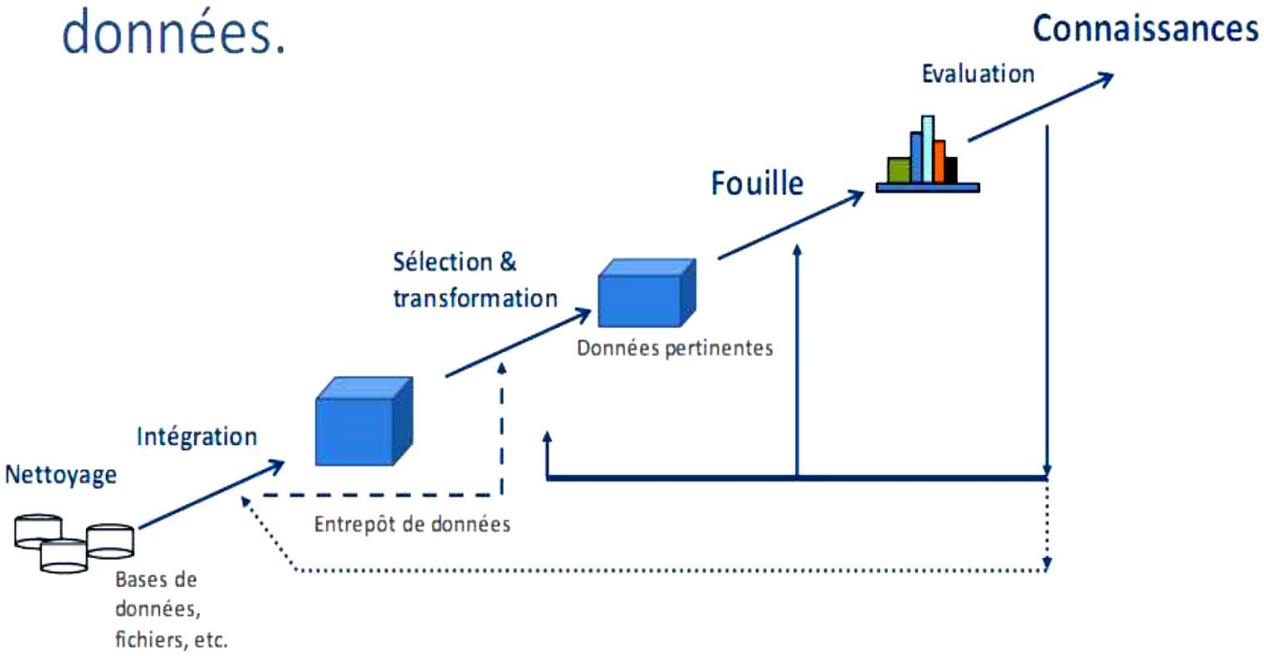


Figure 1.06 : *Le processus de Data Mining*

- Collecte des données : la combinaison de plusieurs sources de données, souvent hétérogènes, dans une base de données.
- Nettoyage des données : la normalisation des données : l'élimination du bruit (les attributs ayant des valeurs invalides et les attributs sans valeurs).
- Sélection des données : Sélectionner de la base de données, les attributs utiles pour une tâche particulière du Data Mining.
- Transformation des données : le processus de transformation des structures des attributs pour être adéquates à la procédure d'extraction des informations.
- Extraction des informations : l'application de quelques algorithmes du Data Mining sur les données produites par l'étape précédente KDD (Knowledge Discovery in Databases).

- Visualisation des données : l'utilisation des techniques de visualisation (histogramme, camembert, arbre, visualisation 3D) pour exploration interactive de données (la découverte des modèles de données).
- Evaluation des modèles : l'identification des modèles strictement intéressants en se basant sur des mesures données. [7]

1.12 Typologie des méthodes de fouilles de données

Deux typologies existent selon le type d'apprentissage utilisé. [8]

1.12.1 Apprentissage supervisé

C'est un processus qui prend en entrée des exemples d'apprentissage contenant à la fois des données d'entrée et de sortie. Les exemples d'apprentissage sont fournis avec leur classe.

- But : ceci permet de classer correctement un nouvel exemple.
- Utilisation : apprentissage utilisé principalement en classification et prédiction. [9]

1.12.2 Apprentissage non supervisé

Ce processus prend en entrée des exemples d'apprentissage ne contenant que des données d'entrée, il n'y a pas de notion de classe.

- But : regrouper les exemples en paquets (clusters) d'exemples similaires.
- Utilisation : apprentissage utilisé principalement en segmentation et association. [9]

1.13 Utilisations de data mining

La fouille de données a aujourd'hui une grande importance économique du fait qu'elle permet d'optimiser la gestion des ressources (humaines et matérielles). Elle est utilisée par exemple dans :

- organisme de crédit : pour décider d'accorder ou non un crédit en fonction du profil du demandeur de crédit, de sa demande, et des expériences passées de prêts.
- optimisation du nombre de places dans les avions, hôtels…
- organisation des rayonnages dans les supermarchés en regroupant les produits qui sont généralement achetés ensemble (pour que les clients n'oublient pas d'acheter un produit parce qu'il est situé à l'autre bout du magasin).
- organisation de campagne de publicité, promotions, ...

- diagnostic médical : par exemple : « les patients ayant tels et tels symptômes et demeurant dans des agglomérations de plus de 10^4 habitants développent couramment telle pathologie ».
- analyse du génome et bio-informatique plus généralement.
- classification d'objets (astronomie, ...).
- commerce électronique, recommandation de produits.
- analyser les pratiques et stratégies commerciales et leurs impacts sur les ventes.
- moteur de recherche sur internet : fouille du web.
- extraction d'information depuis des textes : fouille de textes.
- évolution dans le temps de donnés : fouille de séquences. [4] [7]

1.14 Conclusion

On a vu dans ce chapitre les étapes d'extraction des connaissances à partir de données. On en déduit qu'avant d'avoir les résultats, il faut faire le prétraitement des données ainsi que le codage avant d'appliquer un algorithme de data mining choisi pour l'ECD. La dernière étape du processus d'extraction de connaissances consiste à l'évaluation de résultat à l'aide de diagramme, ou d'un pattern, ainsi que l'interprétation du résultat par des experts dans ce domaine. L'extraction de connaissance ECD est plus importante pour les entreprises car elle permet d'améliorer ses produits, ou de prédire le point stratégique avant de lancer un nouveau produit. Dans le prochain chapitre, on va approfondir nos connaissances sur l'ECD en parlant d'apprentissage automatique ou machine learning, ses types ainsi que quelques exemples de ses algorithmes.

CHAPITRE 2
APPRENTISSAGE AUTOMATIQUE

2.1 Introduction

La Data Science est un vaste champ d'étude interdisciplinaire dont le but fondamental est d'extraire de la connaissance à partir des données. La Data Science étend ces techniques au contexte de l'entreprise par la création de systèmes capables de valoriser cette connaissance des données. L'apprentissage automatique (ou machine learning en anglais), est le moteur de la Data Science. C'est une discipline scientifique, qui est aussi l'un des champs d'étude de l'intelligence artificielle. L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

Ce chapitre se constituera comme suit, on va parler de la définition et des principes d'apprentissage automatique. Nous verrons aussi les différents types d'apprentissage automatiques en commençant par l'apprentissage supervisé qui est l'un de type d'apprentissage les plus importants, ainsi que les applications de l'apprentissage automatique. On va aborder avec des facteurs de pertinence et d'efficacité. Et on terminera ce chapitre en citant quelques algorithmes utilisés en apprentissage automatique.

2.2 Définition de l'apprentissage automatique

L'apprentissage automatique est une technique qui consiste en la conception et le développement d'algorithmes permettant aux ordinateurs (machines) d'améliorer leurs performances au fil du temps sur une base de données. C'est-à-dire qu'il permet la mise en place d'algorithmes en vue d'obtenir une analyse prédictive à partir de données. [9]

2.3 Principe

L'objectif général de l'apprentissage automatique est d'extraire et exploiter automatiquement des connaissances présentent dans un jeu de données, c'est-à-dire sur un ensemble limité de données disponibles. Ce problème se décline en plusieurs variantes, en fonctions des informations disponibles sur le problème traité. Les algorithmes utilisés permettent dans une certaine mesure à un système piloté par ordinateurs, ou assisté par ordinateur d'adapter ses analyses, et comportements

en réponse, en se fondant sur l'analyse de données empiriques provenant d'une base de données ou de capteurs.

La difficulté réside dans le fait que l'ensemble de tous les comportements possibles compte tenu de toutes les entrées possibles devient rapidement trop complexes à décrire dans les langages de programmation disponibles, de sorte qu'on confie en quelque sorte à des programmes le soin d'apprendre de manière à auto-améliorer le système d'analyse ou de réponse, ce qui est une des formes que peut prendre l'intelligence artificielle.

Ces programmes, selon leur degré de perfectionnement intègrent des capacités en probabilités et statistiques, traitement de données et éventuellement d'analyse de données issues de capteurs, de reconnaissance (reconnaissance vocale, reconnaissance de forme, d'écriture, etc.), de data mining et d'informatique théorique. [10] [11]

2.4 Types d'apprentissage

Le système d'apprentissage automatique peut se catégoriser généralement en trois types :

- L'apprentissage supervisé,
- L'apprentissage non-supervisé,
- L'apprentissage par renforcement. [9]

2.4.1 L'apprentissage supervisé

On parle d'apprentissage supervisé si les classes sont prédéterminées et les exemples connus, et par conséquent, le système apprend à classer selon un modèle de classement.

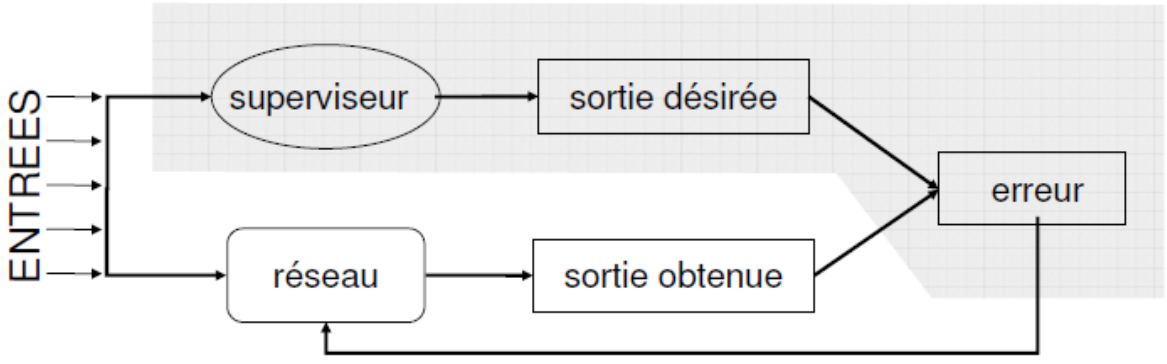


Figure 2.01 : *Structure générale d'un système d'apprentissage supervisé*

Un expert doit alors préalablement correctement étiqueter des exemples. L' « apprenant » peut alors trouver ou approximer la fonction qui permet d'affecter la bonne « étiquette » à ces exemples. Parfois, il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées. On parle alors d'apprentissage supervisé probabiliste.

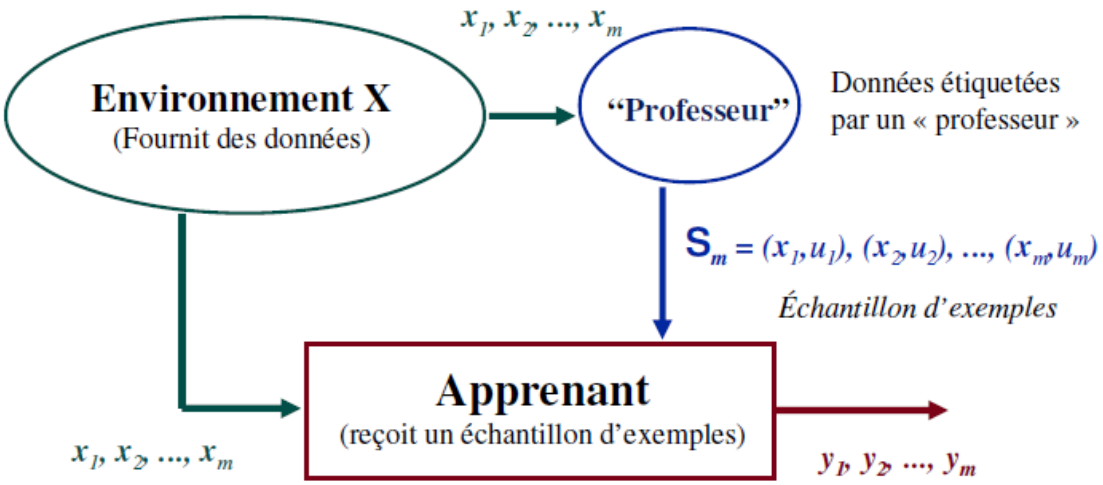


Figure 2.02 : *Système d'approximation de la sortie désirée pour chaque entrée observée*

En sciences cognitives, l'apprentissage supervisé est une technique d'apprentissage automatique qui permet à une machine d'apprendre à réaliser des tâches à partir d'une base d'apprentissage contenant des exemples déjà traités.

De par sa nature, l'apprentissage supervisé concerne essentiellement les méthodes de classification de données et de régression.

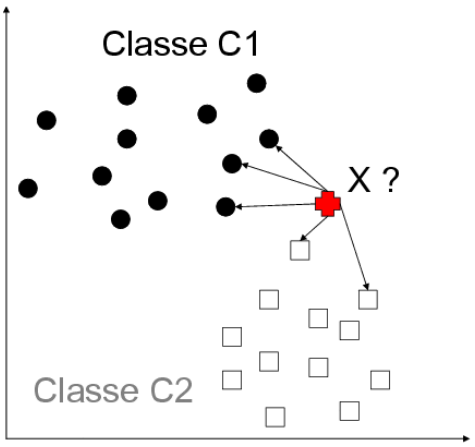


Figure 2.03 : *Un exemple simple de k-plus-proche voisins (kPPV ou kNN)*

La **Figure 2.03 :** montre que la classe d'un exemple est un vote majoritaire des k plus proches voisins. On regarde la classe des voisins les plus proches de l'exemple à tester en se limitant au k plus proche. Et on attribue à l'exemple la classe la plus présente parmi ces voisins. [11]

2.4.2 L'apprentissage non-supervisé

On parle d'apprentissage non-supervisé (ou clustering) quand le système ou l'opérateur ne disposent que d'exemples, mais non d'étiquettes, et que le nombre de classe et leur nature n'ont pas été prédéterminés. Aucun expert n'est disponible ni requis.

L'algorithme doit découvrir par lui-même la structure des données. Le clustering est un algorithme d'apprentissage non-supervisé.

On cherche des régularités sous-jacentes sous forme d'une fonction ou sous forme d'un modèle complexe afin de résumer, détecter des régularités, et comprendre.

Les algorithmes les plus utilisés sont: K-Means clustering, Apriori, etc…

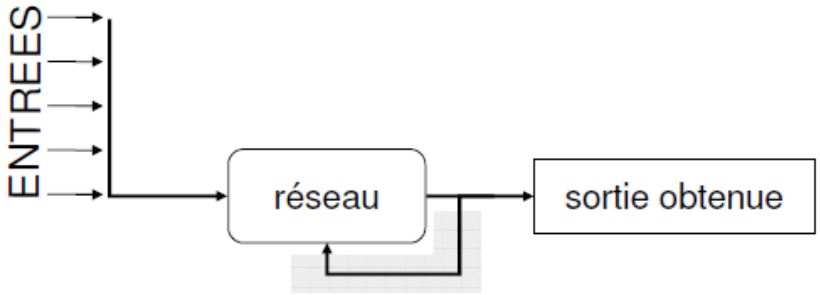


Figure 2.04 : *Structure générale d'un système d'apprentissage non-supervisé*

Le système doit ici, dans l'espace de description (la somme des données), cibler les données selon leurs attributs disponibles, pour les classer en groupe homogènes d'exemples. La similarité est généralement calculée selon la fonction de distance entre paires d'exemples. C'est ensuite à l'opérateur d'associer ou déduire du sens pour chaque groupe et pour les patterns d'apparition des groupes ou groupes dans leur « espace ».

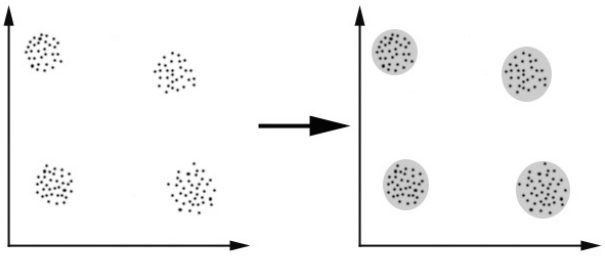


Figure 2.05 : *Exemple de clustering*

On parle aussi d'analyse des données en régression. Si l'approche est probabiliste, c'est-à-dire que chaque exemple au lieu d'être classé dans une seule classe est associé aux probabilités d'appartenir à chacune des classes, on parle alors de « soft clustering ». Il existe de nombreuses applications possibles au clustering, que l'on peut classer en trois groupes principaux : la segmentation, la classification, et l'extraction de connaissances. [12] [13]

2.4.3 L'apprentissage par renforcement

Dans le cas présent, l'algorithme apprend de l'environnement dans lequel il évolue. L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage. Ce type d'apprentissage tente alors de concevoir une stratégie la plus optimisée possible. Le système fonctionne avec des récompenses que renvoie l'environnement à celui-ci.

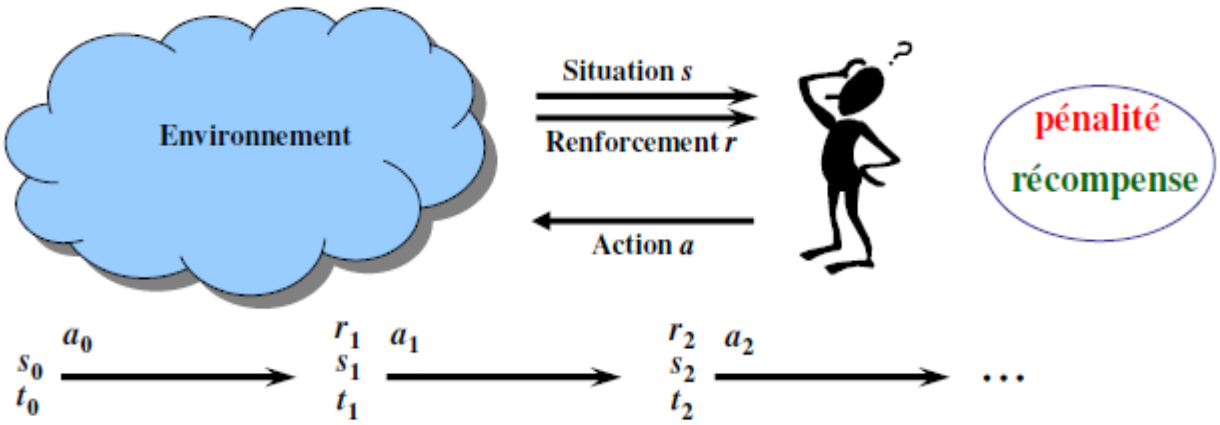


Figure 2.06 : *Approche d'un système d'apprentissage par renforcement*

La **Figure 2.06 :** montre que l'agent apprend à se rapprocher d'une stratégie comportementale optimale par des interactions répétitives avec l'environnement. Les décisions sont prises séquentiellement à des intervalles de temps discrets.

L'Apprentissage par Renforcement se distingue des autres approches d'apprentissage par plusieurs aspects :

- L'apprentissage se fait sans supervision.
- Il repose sur le principe d'essai/erreur. [12]

2.5 Applications

L'apprentissage automatique est utilisé pour doter des machines de systèmes de perception de leur environnement : vision et reconnaissance d'objets, de visages, de schémas, etc. par ordinateur, reconnaissance des langages naturels, de l'écriture, reconnaissance de formes syntaxiques, moteurs de recherche, aide au diagnostic médical, bio-informatique, interfaces cerveau-machine, détection de fraudes à la carte de crédit, analyse financière, dont analyse du marché boursier, classification des séquences d'ADN, jeu, génie logiciel, sites Web adaptatifs ou mieux adaptés, locomotion de robots, etc.

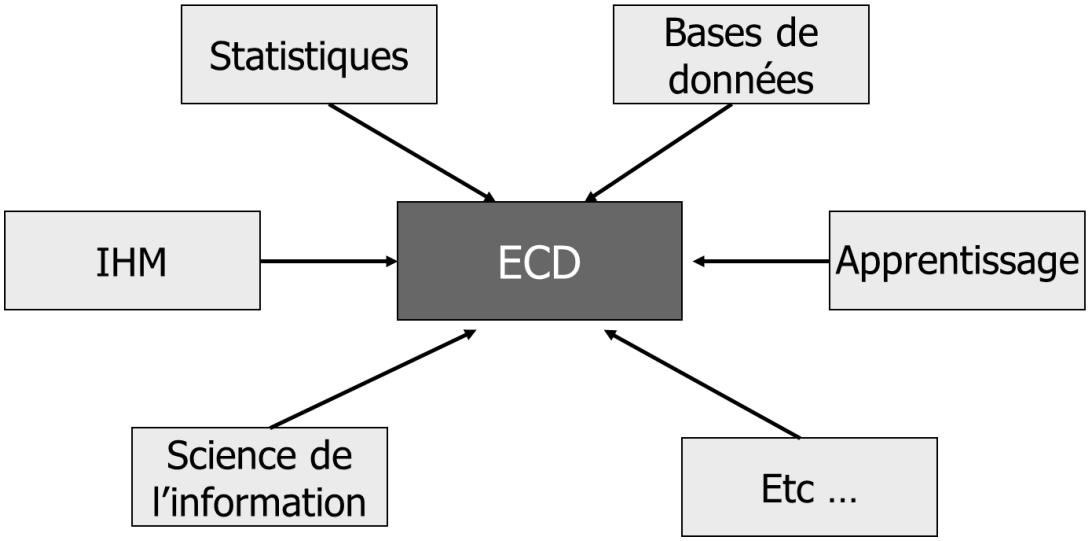


Figure 2.07 : *Domaines d'application de l'apprentissage automatique*

Voici deux exemples d'application du système d'apprentissage automatique :

- Un système d'apprentissage automatique peut permettre à un robot, ayant la capacité de bouger ses membres mais ne sachant initialement rien de la coordination des mouvements permettant la marche, d'apprendre à marcher. Le robot commencera par effectuer des mouvements aléatoires, puis, en sélectionnant et privilégiant les mouvements lui permettant d'avancer, mettra peu à peu en place une marche de plus en plus efficace.
- La reconnaissance de caractères manuscrits est une tâche complexe car deux caractères similaires ne sont jamais exactement égaux. On peut concevoir un système d'apprentissage

automatique qui apprend à reconnaître des caractères en observant des « exemples », c'est-à-dire des caractères connus. [14]

2.6 Facteurs de pertinence et d'efficacité

La qualité du travail dépendra de facteurs initiaux contraignants, liées à la base de données.

- Nombre d'exemples : moins il y en a plus l'analyse est difficile, mais plus il y en a plus le besoin de mémoire informatique est élevé et plus longue est l'analyse.
- Nombre et qualité des attributs décrivant ces exemples : La distance entre deux "exemples" numériques (prix, taille, poids, etc) est facile à établir, celle entre deux attributs catégoriels (couleur, beauté, utilité), est plus délicate.
- Pourcentage de données renseignées et manquantes.
- Le nombre et la « localisation » des valeurs douteuses (erreurs) ou naturellement non conformes au pattern de distribution générale des « exemples » sur leur espace de distribution impacteront sur la qualité de l'analyse.

La qualité de l'apprentissage et de l'analyse dépendent du besoin en amont et a priori en compétence de l'opérateur pour préparer l'analyse. Elle dépend aussi de la complexité du modèle (spécifique ou généraliste) et de son adaptation au sujet à traiter. La qualité du travail dépendra aussi du mode ou représentation visuelle des résultats pour l'utilisateur final parce qu'un résultat pertinent pourrait être caché dans un schéma trop complexe, ou mal mis en évidence par une représentation graphique inappropriée. [15]

2.7 Les algorithmes utilisés

Ce sont dans ce domaine :

- les machines à vecteur de support,
- le boosting,
- la méthode des k plus proches voisins pour un apprentissage supervisé,
- les réseaux de neurones,
- les arbres de décision,
- le Bagging.

Ces méthodes sont souvent combinées pour obtenir diverses variantes d'apprentissage. L'utilisation de tel ou tel algorithme dépend fortement de la tâche à résoudre : classification, estimation de valeurs, etc.

Parmi ces algorithmes, nous ne détaillerons que ceux qui sont les plus utilisés tels que : les machines à vecteur de support, les k plus proches voisins, les réseaux de neurones, les arbres de décisions, et la classification bayésienne. [16]

2.7.1 Les machines à vecteur de support

2.7.1.1 Présentation

SVM (Support Vector Machines) ou Machines à vecteurs supports sont des classifieurs faisant partie des techniques d'apprentissage supervisé. Introduits pour résoudre des problèmes de classification, ils ont connu depuis un grand succès car utilisés massivement dans divers domaines : reconnaissance de formes, OCR (Optical Character Recognition), bioinformatique… L'usage s'est également répandu vers la résolution des problèmes de régression, aussi bien gérés que les classifications.
Cette technique fait appel à un jeu de données dit « d'apprentissage » dont les instances contiennent une valeur cible (target) également appelée « étiquette de classe » ainsi que plusieurs attributs (attributes) représentant l'ensemble des variables observées, et ce pour produire un modèle permettant de prédire les valeurs cibles d'un autre ensemble dit « de test », en ne fournissant à ce modèle là que les attributs des données de test.
En d'autres termes, si on considère un jeu de données divisé en deux groupes : un groupe pour les exemples connus et un autre pour les exemples non connus, le but des SVM est d'apprendre une fonction qui traduit le comportement des exemples connus pour prédire les cibles des exemples inconnus. [17]

2.7.1.2 Fonctionnement

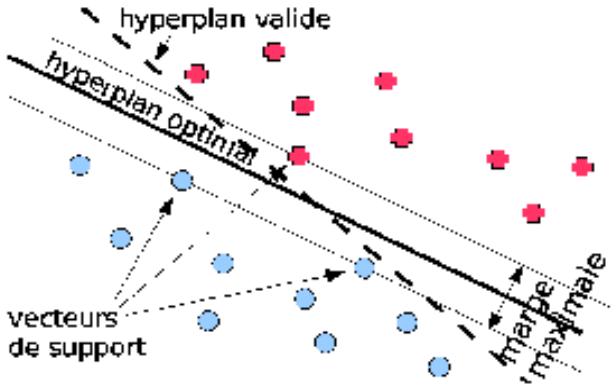


Figure 2.08 : *Machines à vecteur de support*

Les SVM sont également appelés « classifieurs à vaste marge » car leur objectif est de trouver l'hyperplan séparateur optimal qui maximise la marge entre les classes dans un espace de grande dimension. La marge est la distance entre la frontière de séparation et les échantillons les plus proches, ces derniers sont appelés vecteurs supports.
Une marge maximale permet d'obtenir une plus petite dimension de VC (Vapnik-Chervonenkis), « Théorie statistique de l'apprentissage », ce qui assure de bonnes performances en généralisation. [18]

2.7.1.3 Les Régressions à vecteur de supports

Lorsque les SVM sont utilisés dans des problèmes de régression pour prédire des valeurs réelles, on parle des SVR (Support Vector Regression).
Les SVM peuvent également être mis en œuvre en situation de régression, c'est-à-dire pour l'approximation de fonctions quand Y est quantitative. Dans le cas non linéaire, le principe consiste à rechercher une estimation de la fonction par sa décomposition sur une base fonctionnelle. [18]

2.7.2 Le boosting

2.7.2.1 Définition

Le Boosting est une technique servant à améliorer les capacités de généralisation d'un système de classification ou de prédiction en optimisant les performances de son algorithme d'apprentissage. Il fait partie des méthodes de combinaison de modèles, dont l'objectif est de combiner plusieurs hypothèses pour obtenir des estimations meilleures que celles d'un modèle unique. Ces méthodes ont la capacité d'améliorer les performances d'un algorithme d'apprentissage faible. L'idée est qu'en divisant l'ensemble d'apprentissage en plusieurs sous-ensembles moins complexes, il est plus facile d'apprendre les données, et par conséquent d'obtenir une meilleure généralisation.
La véritable émergence du boosting a débuté avec la création d'AdaBoost (Adaptative Boosting). [19]

2.7.2.2 Fonctionnement

Diverses méthode s'inscrivent dans le même registre que le boosting, tels le bagging, le cascading, le stacking, qui agissent toutes par combinaison de modèles, mais qui diffèrent dans leur manière de générer ces modèles-là. Le bagging par exemple divise l'ensemble d'apprentissage en plusieurs sous-ensembles et applique par la suite son algorithme d'apprentissage sur chacun d'entre eux.

Le boosting quant à lui procède de façon séquentielle, à chaque itération la sélection d'un élément du nouveau sous-ensemble dépend des résultats obtenus lors de l'apprentissage précédent sur cet élément-là. Une façon de réduire rapidement l'erreur d'apprentissage en se concentrant sur les exemples les plus difficiles.

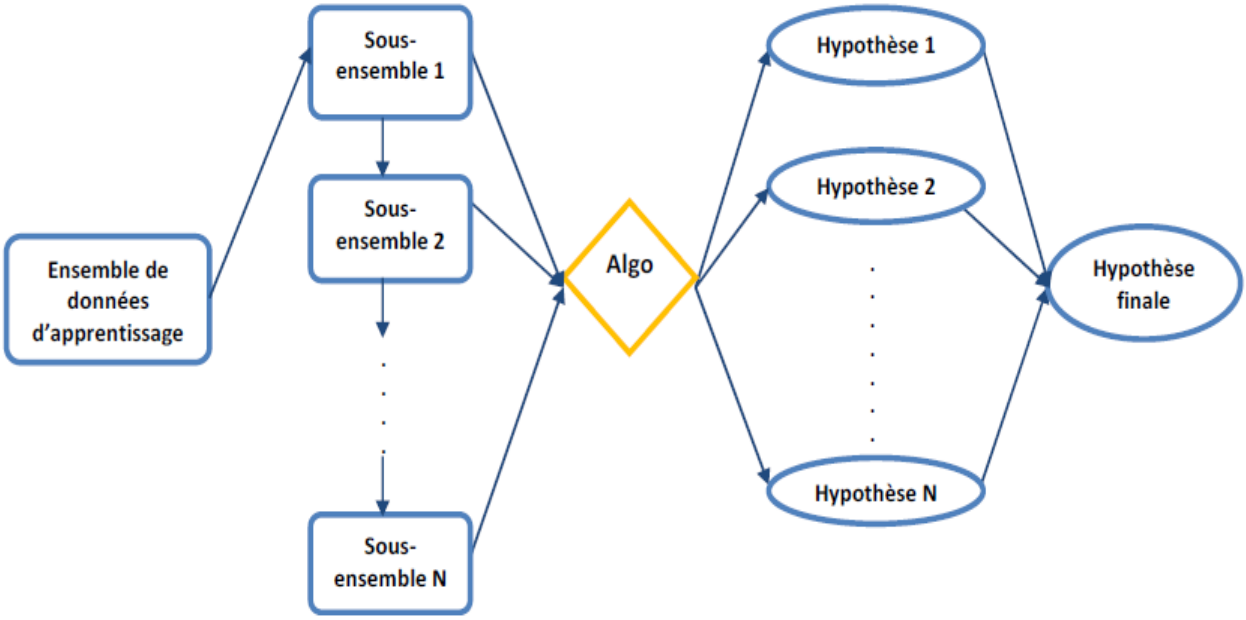


Figure 2.09 : *Fonctionnement du boosting*

Le boosting a connu un grand succès sur divers problèmes d'optimisation, en montrant des performances très satisfaisantes par rapport à d'autres algorithmes, entre autre à celles du bagging, ce qui explique le fait que son étude soit si prisée par les chercheurs actuellement. [18] [19]

2.7.3 Les k plus proches voisins

Le kppV (k plus proches Voisins) méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Elle méthode permet de traiter des problèmes avec un grand nombre d'attributs. Plus le nombre d'attributs est important, plus le nombre d'exemples doit être grand. Les performances de la méthode dépendent du choix de la distance et du nombre de voisins. [20]

2.7.3.1 Algorithme

- On dispose d'une base de données d'apprentissage constituée de m couples « entrée-sortie ».

- Pour estimer la sortie associée à une nouvelle entrée x_1, la méthode consiste à prendre en compte les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x_1, selon une distance à définir.
- Paramètre : le nombre k de voisins.
- Donnée : un échantillon de m exemples et leurs classes.
- La classe d'un exemple S est c(S).
- Entrée : un enregistrement x_2.
- Déterminer les k plus proches exemples de x_2 en calculant les distances.
- Combiner les classes de ces k exemples en une classe c.
- Sortie : la classe de x_2 est $c(x_2)=c$.

Par exemple dans un problème de classification, on retiendra la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x. [19]

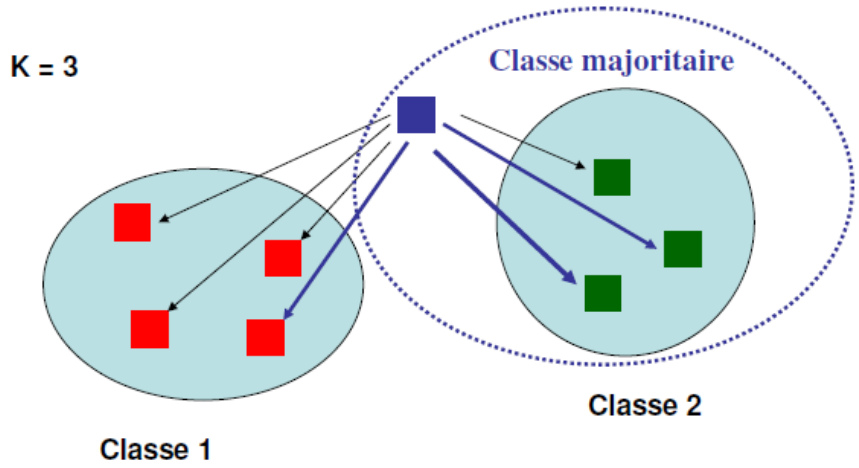


Figure 2.10 : *Exemple de classification par kppv avec K=3*

2.7.3.2 Distance

Le choix de la distance est primordial au bon fonctionnement de la méthode. Les distances les plus simples permettent d'obtenir des résultats satisfaisants. [19]

2.7.3.3 Propriétés de la distance

- $d(A,A)=0$ (2.01)
- $d(A,B)= d(B,A)$ (2.02)

- d(A,B)=d(A,C) + d(B,C) (2.03)
- d(x,y) = |x-y| (2.04)
- d(x,y) = |x-y|/dmax, (2.05)

  où : dmax est la distance maximale entre deux numériques du domaine considéré.
- Données binaires : 0 ou 1. On choisit d(0,0)=d(1,1)=0 et d(0,1)=d(1,0)=1.
- Données énumératives : La distance vaut 0 si les valeurs sont égales et 1 sinon.
- Données énumératives ordonnées : elles peuvent être considérées comme des valeurs énumératives mais on peut également définir une distance utilisant la relation d'ordre.
- Soit X = (x1,..., xn) et Y = (y1,..., yn) deux exemples, la distance euclidienne entre X et Y est:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \quad (2.06)$$

2.7.4 Les réseaux de neurones

2.7.4.1 Définition

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire (neurone artificiel) calcule une sortie unique sur la base des informations qu'il reçoit. On peut dire que l'inspiration naturelle est en analogie avec le cerveau. [20]

2.7.4.2 Le Neurone Formel

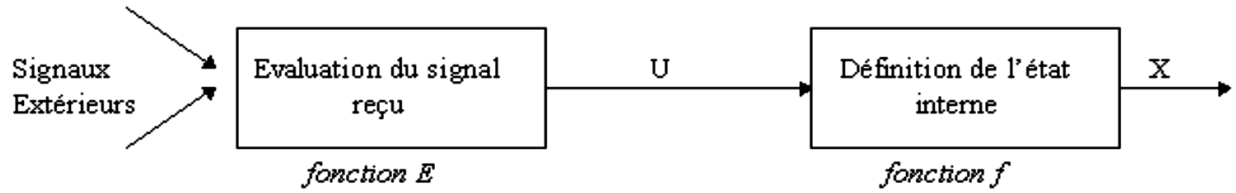


Figure 2.11 : *Structure d'un neurone formel*

Le neurone formel, l'unité élémentaire d'un RNA, se compose de deux parties. [20]

- évaluation de la stimulation reçue (fonction E) :

  E (entrées) = U

- évaluation de son activation (fonction f) :

f (U) = X

La fonction d'entrée est alors la somme pondérée des signaux d'entrée. [21]

$$U_i = E(x_1 \ldots\ldots x_j \ldots\ldots x_n) = \sum_{j=1}^{n} W_{ij} x_j \qquad (2.07)$$

2.7.4.3 Interprétation mathématique

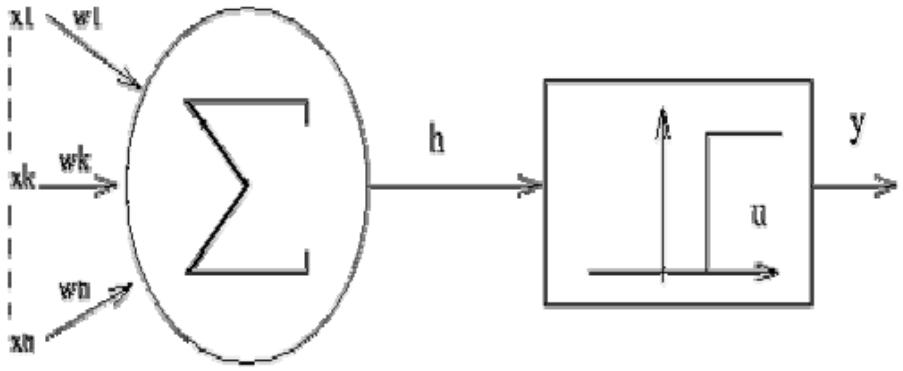


Figure 2.12 : *Translation mathématique d'un réseau de neurone*

Voici sa représentation mathématique :

$$y = f(\sum_{j=1}^{n} w_j . xj - \theta) \qquad (2.08)$$

Où : θ est le seuil du neurone de sortie. [22]

2.7.5 Les arbres de décision

2.7.5.1 Présentation

L'arbre de décision est une méthode symbolique. Elle est utilisée lorsqu'on cherche une procédure de classification compréhensible.

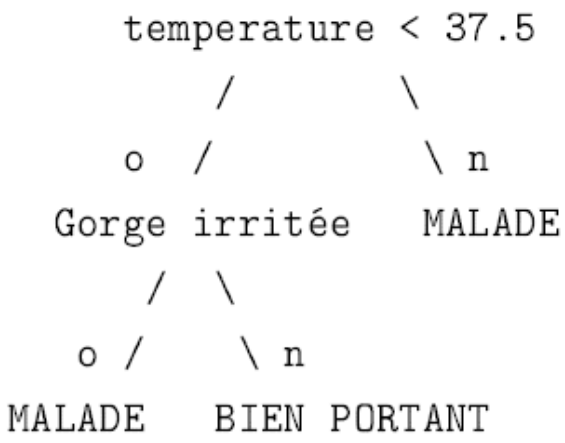


Figure 2.13 : *Exemple arbre de décision*

Un arbre de décision est un arbre au sens informatique. Les nœuds sont repérés par des positions appartenant à {1, . . ., p}*, où p est l'arité maximale des nœuds. Avec laquelle les nœuds internes sont les nœuds de décision. Un nœud de décision est étiqueté par un test qui peut être appliqué à chaque description d'un individu d'une population. Et chaque test examine la valeur d'un unique attribut.

Dans les arbres de décision binaires on omet les labels des arcs et les feuilles sont étiquetées par une classe.

À chaque arbre, on associe naturellement une procédure de classification, et à chaque description complète est associée une seule feuille de l'arbre.

La procédure de classification représentée par un arbre correspond à des règles de décision. [23]

2.7.5.2 Construction des arbres de décision

Étant donné un échantillon S et des classes $\{1, . . ., c\}$, on veut construire un arbre t.

À chaque position p de t correspond un sous-ensemble de S qui contient les éléments de S et satisfant les tests de la racine jusqu'à p.

On définit pour chaque p :

- $N(p)$ = le cardinal de l'ensemble des exemples associé à p.
- $N(k/p)$ = le cardinal de l'ensemble des exemples associé à p de classe k.
- $P(k/p) = N(k/p)/N(p)$ = la proportion d'éléments de classe k à la position p. [24]

2.7.5.3 Avantages

- Ceci permet d'avoir des connaissances « intelligibles ».
- Elle permet une traduction directe de l'arbre vers une base de règles, et aussi la sélection automatique des variables pertinentes.
- Elle est non paramétrique.
- C'est un traitement indifférencié selon le type des variables prédictives.
- Une technique robuste face aux données aberrantes, solutions pour les données manquantes, et aussi robuste face aux variables redondantes.
- Elle est avantageuse en termes de rapidité et capacité à traiter des très grandes bases.
- Elle permet d'enrichir l'interprétation des règles à l'aide des variables non sélectionnées.
- Elle offre la possibilité pour le praticien d'intervenir dans la construction de l'arbre. [24]

2.7.5.4 Inconvénients

- L'arbre de décision engendre un problème de stabilité sur les petites bases de données (feuilles à très petits effectifs).
- A cause de son principe de recherche « pas-à-pas » : ceci provoque une difficulté à trouver certaines interactions.
- Elle est peu adaptée au « scoring ».
- Elle n'offre que des performances moins bonnes en général par rapport aux autres méthodes (en réalité, performances fortement dépendantes de la taille de la base d'apprentissage). [25]

2.7.6 Le Bagging

2.7.6.1 Présentation

Le bagging regroupe un ensemble de méthodes d'apprentissage automatique. Tout comme le boosting, les méthodes bagging sont des méthodes d'agrégation. Ce terme vient de la contraction de Boostrap Aggregating. L'approche bagging consiste ainsi à tenter d'atténuer la dépendance entre les estimateurs que l'on agrège en les construisant sur des échantillons bootstrap. Nous présentons, dans l'algorithme qui suit, cette famille de méthodes dans un contexte de régression. Elles s'étendent aisément à la classification supervisée.

2.7.6.2 Algorithme de Bagging

L'algorithme est simple à implémenter : il suffit de construire B estimateurs sur des échantillons bootstrap et de les agréger. Le fait de considérer des échantillons bootstrap introduit un aléa supplémentaire dans l'estimateur. Afin de prendre en compte cette nouvelle source d'aléatoire, on note $\theta_k = \theta_k(D_n)$ l'échantillon bootstrap de l'étape k et $\widehat{m}\ (., \theta_k)$ l'estimateur construit à l'étape k. On écrira l'estimateur final : [25] [26] [27]

$$\widehat{m_B}(x) = \frac{1}{B} \sum_{k-1}^{B} \widehat{m}(x, \theta_k) \tag{2.09}$$

Où :

- $D_n = (X_1, Y_1), \ldots, (X_n, Y_n)$, un n échantillon.
- $m(x) = E[Y \mid X = x]$, $x \in IR^p$.

Entrées :

- x l'observation à prévoir
- un régresseur (arbre CART, 1 plus proche voisin...)
- d_n l'échantillon
- B le nombre d'estimateurs que l'on agrège.

Pour k = 1, . . . ,B :

1. Tirer un échantillon bootstrap d_n^k dans d_n
2. Ajuster le régresseur sur cet échantillon bootstrap : $\widehat{m_k}$

Sortie : L'estimateur $\widehat{m_B}(x) = \frac{1}{B} \sum_{k-1}^{B} \widehat{m}(x, \theta_k)$.

2.8 Conclusion

Dans ce chapitre concernant l'apprentissage automatique, on a pu se familiariser avec cette technologie après avoir expliqué sa définition, ses principes, les différents types d'apprentissage et leurs algorithmes. Son idée principale est d'apprendre et d'évoluer depuis des exemples ou des problèmes déjà rencontrés auparavant. Cette technologie est très souvent synonyme de l'intelligence artificielle, et elle possède de nombreuses algorithmes comme en apprentissage supervisé et en non-supervisé. Dans la suite, on va aborder sur les techniques de modélisation des règles et patterns dans un système d'apprentissage supervisée pour la prédiction.

CHAPITRE 3
MODELISATION DES REGLES ET PATTERNS DANS UN SYSTEME D'APPRENTISSAGE SUPERVISEE POUR LA PREDICTION

3.1 Introduction

La croissance actuelle de la puissance des ordinateurs et des réseaux de télécommunications crée de nouveaux besoins, et permet à des applications innovantes d'apparaître. Une de ses évolutions est la naissance d'un système d'apprentissage automatique, utilisée dans de nombreux domaines tels le système d'information. On a vue précédemment qu'il existe principalement deux grandes catégories de système d'apprentissage : les méthodes *supervisées*, qui ont un rôle *prédictif*, permettant d'évaluer la distribution d'une quantité sans la mesurer directement, mais en se basant sur des valeurs qui lui sont, et les méthodes *non-supervisées*, dont le rôle est principalement *descriptif*, s'attachant à isoler l'information utile au sein d'un jeu de données. Ce chapitre se focalise sur la modélisation des règles et patterns dans un système d'apprentissage supervisé, qui répond à un objectif clair : minimiser un coût ou une erreur de prédiction sur des données de test. Ce système d'apprentissage se base surtout sur les algorithmes de classification et de régression qu'on va détailler dans cette partie.

3.2 Principe de l'apprentissage supervisé

L'apprentissage supervisé fait intervenir deux types de variables :

- les variables d'entrée, notées X. Ces variables sont appelées variables exogènes ou quelconques,
- les variables de sortie, notées Y. Ces variables doivent être prédites à partir de la valeur de X associé et appelées aussi endogène. [30]

On veut construire une fonction de classement telle que :

$$Y = f(X, \propto) \qquad (3.01)$$

La modélisation du lien entre X et Y est donc primordiale. Lorsque la variable Y est discrète, on peut associer chaque valeur possible à une catégorie. C'est la classification supervisée.

L'apprentissage supervisé consiste alors à apprendre une fonction objective pour prédire la valeur d'une classe. Ceci se passe généralement en deux étapes :

- Apprentissage du modèle sur un jeu de données d'apprentissage,
- Test du modèle sur un jeu de données test. [27] [30]

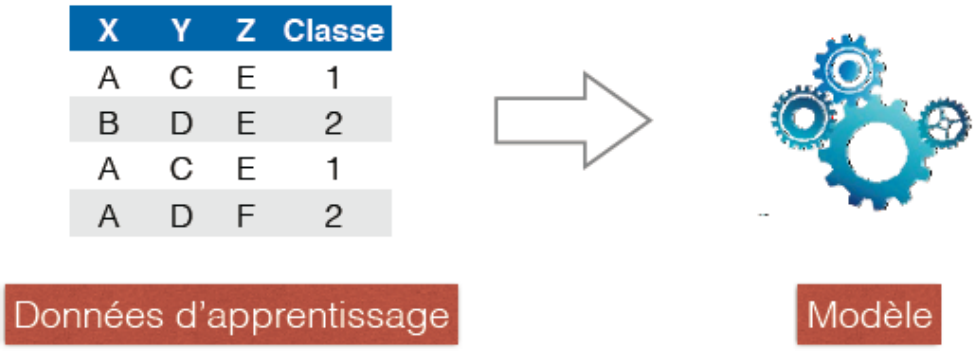


Figure 3.01 : *Apprentissage du modèle sur un jeu de données d'apprentissage*

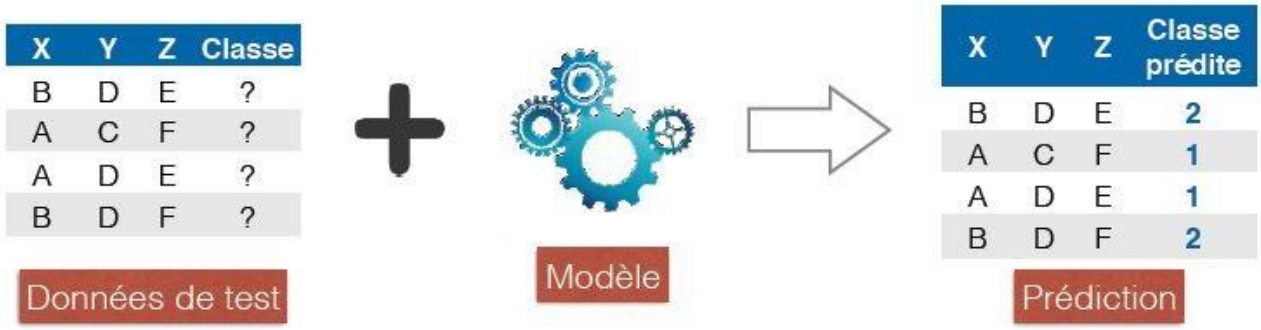


Figure 3.02 : *Test du modèle sur un jeu de données test*

3.3 Objectif de l'apprentissage

Utiliser un échantillon Ω, extraite d'une population, pour choisir la fonction f et ses paramètres α telle que l'on minimise l'erreur théorique. [30]

$$ET = \frac{1}{card(\Omega)} \sum_{\Omega} \Delta\,[Y, \hat{f}\,(X, \hat{\alpha})\,] \tag{3.02}$$

Où : $\Delta\,[.] = \begin{cases} 1\ si\ Y \neq \hat{f}\,(X, \hat{\alpha}) \\ 0\ si\ Y = \hat{f}\,(X, \hat{\alpha}) \end{cases}$

3.4 Jeu de données

3.4.1 Choix des jeux de données de test et d'apprentissage

Soit D le jeu de données. On note D_{train} le jeu d'apprentissage et D_{test} le jeu de test. On a :

- $D = D_{train} \cup D_{test}$ (3.03)
- $D_{train} \cap D_{test} = \emptyset$ (3.04)

L'objectif dans ce processus est de pouvoir être représentatif pour l'ensemble du jeu de donnée et aussi d'éviter l'overfitting ou le sur-apprentissage. [31]

3.4.2 Validation simple

Le principe de la validation simple est de découper le jeu de données en deux, formant ainsi une donnée d'apprentissage et une donnée test. Ceci est surtout recommandé lorsque le jeu de données est large. Ces données se répartissent couramment comme suit : 50 % apprentissage - 50% test et/ou 2/3 apprentissage - 1/3 test. [30]

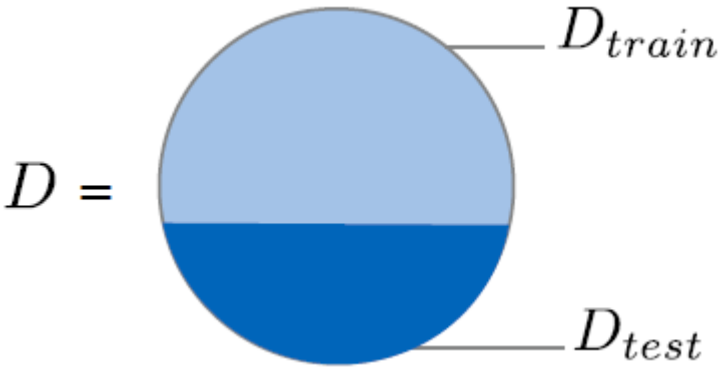


Figure 3.03 : *Découpage par validation simple*

3.4.3 Validation croisée

Comme la validation simple, la validation croisée consiste à découper le jeu de données en K (5 ou 10 généralement) parties. Et on apprend sur les K-1 parties et on teste sur la K^{ème} partie. Ainsi, le processus est répété K fois c'est-à-dire que chaque partie sert de jeu de données de test. [30]

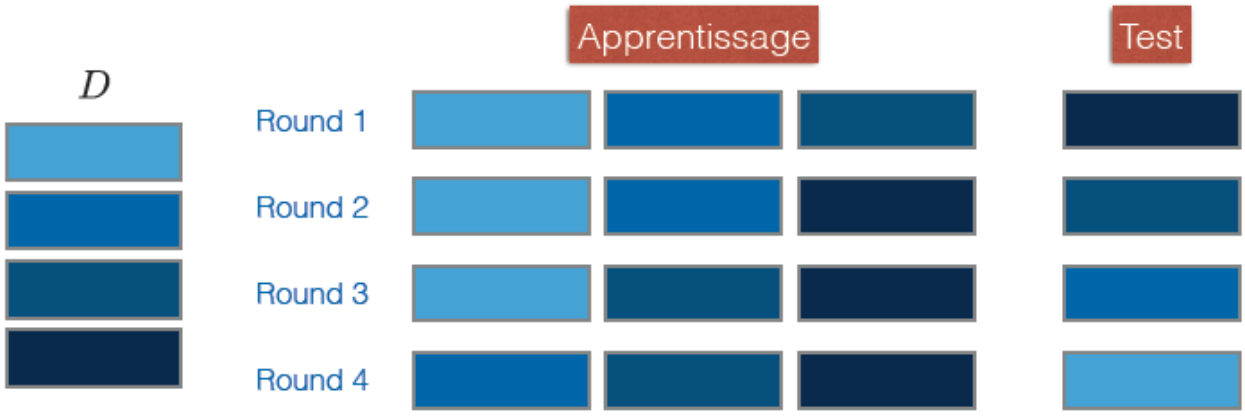


Figure 3.04 : *Découpage par validation croisée*

3.5 Évaluation de l'apprentissage

3.5.1 Matrice de confusion

La matrice de confusion ou de contingence consiste à confronter la vraie valeur avec la prédiction comme le tableau ci-dessous nous reflète. [32]

		Prédite		
		+	-	Total
Observée	+	a	b	a+b
	-	c	d	c+d
	Total	a+c	b+d	n

Tableau 3.01: *Matrice de confusion*

3.5.2 Receiver Operating Characteristic

3.5.2.1 Principe

La courbe ROC (Receiver Operating Characteristic) est communément utilisée pour évaluer les performances d'un classifieur bi-classe. Dans cette méthode, il est nécessaire d'ordonner les instances selon la vraisemblance d'appartenir à la classe positive. [32]

3.5.2.2 Exploitation de la courbe

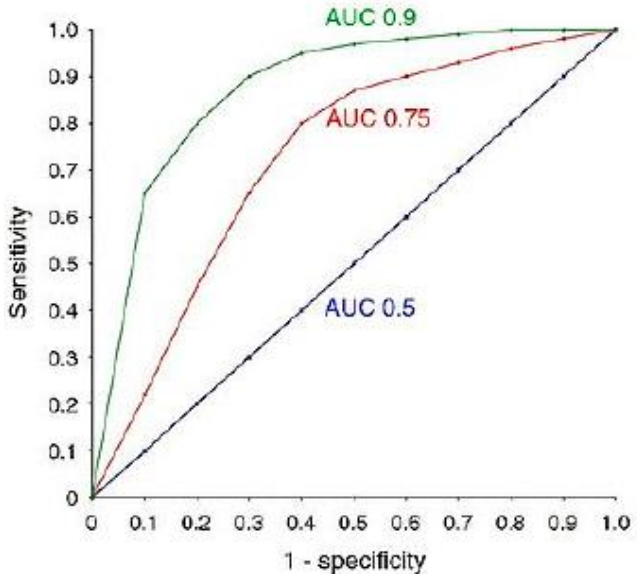


Figure 3.05 : *Exploitation de la courbe ROC*

La courbe ROC est nécessaire pour le calcul de l'aire sous la courbe ou AUC (Air Under Curve). Le résultat de cette dernière permet d'en déduire les conditions suivantes :

- AUC = 1, équivaut à un tirage aléatoire,
- AUC = 1, équivaut aussi à un classifieur parfait.

3.5.2.3 Construction de la courbe ROC

Rang		1	2	3	4	5	6	7	8	9	10
Classe		+	+	-	-	+	-	-	+	-	-
VP	0	1	2	2	2	3	3	3	4	4	4
FP	0	0	0	1	2	2	3	4	4	5	6
VN	6	6	6	5	4	4	3	2	2	1	0
FN	4	3	2	2	2	1	1	1	0	0	0
RVP	0	0,25	0,5	0,5	0,5	0,75	0,75	0,75	1	1	1
RFP	0	0	0	0,17	0,33	0,33	0,50	0,67	0,67	0,83	1

Tableau 3.02: *Matrice de contingence*

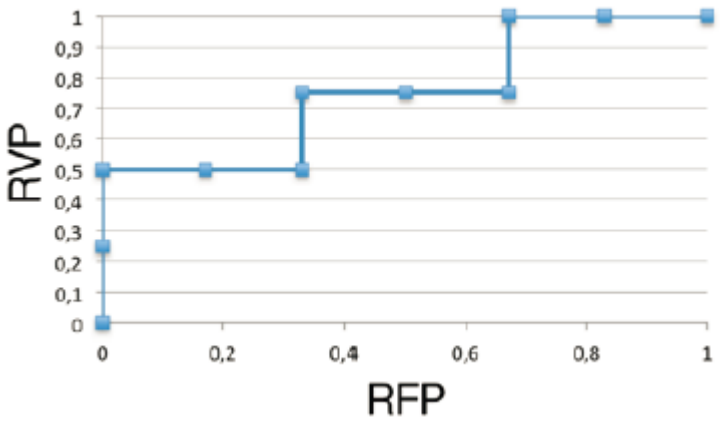


Figure 3.06 : *Exemple de courbe ROC*

3.5.3 Quelques indicateurs

Voici quelques indicateurs permettant d'évaluer l'apprentissage :

- Vrais positifs $VP = a$ (3.05)
- Vrais négatifs $VN = b$ (3.06)

- Faux positifs $FP = c$ (3.07)
- Faux négatifs $FN = d$ (3.08)
- Taux d'erreur = $\frac{(c+b)}{n}$ (3.09)
- Sensibilité = Rappel = Taux de $VP = \frac{a}{(a+b)}$ (3.10)
- Précision = $\frac{a}{(a+c)}$ (3.11)
- Taux de $FP = \frac{c}{(c+d)}$ (3.12)
- Spécificité = $\frac{d}{(c+d)}$ = 1 – Taux de FP (3.13)
- Ratio Vrai Positif (RVP) = $\frac{a}{(a+d)}$ = sensitivité (3.14)
- Ratio Faux Positif (RFP) = $\frac{c}{(b+c)}$ = 1 - spécificité (3.15)

3.6 Classification supervisée

3.6.1 Classification

L'objectif de la classification est l'identification des classes auxquelles appartiennent des objets à partir de traits descriptifs (attributs, caractéristiques). [33]

3.6.2 Principe de la classification supervisée

La classification supervisée consiste à inférer, à partir d'un échantillon de données classées, une procédure de classification. [30]

3.6.3 Condition de la classification et prédiction

Au vu d'un échantillon, il faut bien classer et bien prédire. Il s'agit alors d'inférer une procédure de classification dont l'erreur de classification est minimale, c'est-à-dire que la probabilité qu'un exemple tiré aléatoirement et mal classé par la procédure soit minimale.

On s'intéresse, par conséquent, à générer des procédures ayant un bon pouvoir prédictif, soit encore, à des procédures capables de classer de nouveaux exemples.

Cependant, l'apprenant n'a pour données que l'échantillon et il est possible pour lui de générer une procédure qui classifie bien tous les exemples de l'échantillon mais qui ait un mauvais pouvoir de prédiction. [29]

3.6.4 L'erreur apparente

Soit S un échantillon et C une procédure de classification, le taux d'erreur apparent sur S est :

$$Eapp(C) = \frac{err}{S} \tag{3.16}$$

Où :

- err : est le nombre d'exemples de S qui sont mal classés par C.
- S : est le cardinal de S.

Nous savons que l'erreur réelle ou erreur de classification *E(C)* est la somme pondérée des probabilités d'erreur sur l'ensemble des descriptions d de D. L'erreur réelle est indépendante de l'échantillon alors que l'erreur apparente est mesurée sur l'échantillon. Et apprendre, c'est trouver une procédure de classification C qui minimise l'erreur de classification *E(C)*, or l'apprenant n'a accès qu'à l'erreur apparente *Eapp(C)* mesurée sur *S*. On peut, cependant, démontrer que, lorsque la taille de l'échantillon tend vers l'infini, l'erreur apparente converge en probabilité car les éléments de S sont supposés tirés aléatoirement vers l'erreur réelle, soit :

$$lim_{S \to \infty} E_{app}(C) = E(C) \tag{3.17}$$

Mais, en général, on ne dispose que d'un échantillon de taille trop petite pour que ce résultat puisse être utilisé. Le problème est donc de concevoir des méthodes ou algorithmes qui vont générer des procédures de classification d'erreur apparente petite tout en assurant une erreur de classification petite. [33]

3.6.5 Les méthodes de classification supervisée

Il existe plusieurs méthodes de classification supervisée, parmi eux, ce sont : [31]

- Le classifeur naïf de Bayes
- Méthodes paramétriques et non paramétriques
- Minimiser l'erreur apparente
- Choix de l'espace des hypothèses
- Estimer l'erreur réelle
- Utilisation d'un ensemble Test
- Re-échantillonage

3.6.5.1 Le classifeur naïf de Bayes

Soit d un langage de description, et {1,...,c} l'ensemble des classes, sous les hypothèses usuelles d'existence de lois de probabilité, la règle de classification de Bayes est la procédure qui, à toute description d de D associe, [33]

$$C_{Bayes}(d) = argmax\, P(k/d) = argmax\, P(d/k) \times P(k) \tag{3.18}$$

Où :

- $k \in \{1,...,c\}$
- k : argmax f(k), retourne la valeur de k qui maximise f.

Mais, en règle générale, les quantités P(d/k) et P(k) ne sont pas connues.
En supposant que D est un produit cartésien de domaines, et également en disposant d'un échantillon S. La règle de classification de Bayes s'écrit : [33]

$$C_{NaiveBayes}(\, d\,) = argmax \prod_{i \in \{1,...,n\}} \hat{P}\,(d_i/k) \times \hat{P}\,(k) \tag{3.19}$$

Où : $k \in \{1,...,c\}$

3.6.5.2 Méthodes paramétriques et non paramétriques

Nous avons supposé l'existence de lois de probabilité fixées mais inconnues. Le classifieur naïf de Bayes suppose que les probabilités de certains événements peuvent être estimées par leurs fréquences et fait une hypothèse forte d'indépendance des attributs. En statistiques, on classe habituellement les méthodes d'apprentissage selon les hypothèses que l'on fait sur les lois de probabilité. Si elles font partie d'une famille paramétrée de distributions, on parlera de méthodes paramétriques. Par exemple, si l'on sait que P est une distribution normale, il suffit de connaître deux paramètres, sa moyenne m et son écart-type s, pour identifier P totalement. Il s'agit alors de mettre en œuvre des techniques permettant d'estimer ces paramètres pour avoir une bonne approximation de P pour ensuite déterminer une procédure de classification.
Lorsqu'on ne fait aucune hypothèse a priori sur la distribution P, on parle de problèmes et de méthodes non paramétriques. Les problèmes à résoudre sont alors plus complexes. Comme exemple

de méthodes non paramétriques, on peut citer les méthodes des k-plus proches voisins et des noyaux de Parzen. Ces deux méthodes sont basées sur des notions de proximité entre éléments de D, la classe attribuée à une nouvelle description se fait en fonction des classes des descriptions proches dans l'échantillon. Les méthodes développées en apprentissage automatique (arbres de décision, réseaux de neurones, algorithmes génétiques) sont également non paramétriques. [34]

3.6.5.3 Minimiser l'erreur apparente

En classification supervisée, il faut choisir une fonction de classement au vu d'un échantillon S. Nous sommes confrontés aux deux difficultés suivantes :

- L'erreur apparente est, en général, une version très (trop) optimiste de l'erreur réelle.
- L'espace de toutes les fonctions de D dans {1,...,c} est de taille considérable et, pour des raisons de complexité en temps de calcul et en espace mémoire, il est impossible d'explorer cet espace.

Ces deux difficultés nous amènent à limiter la recherche d'une fonction à un espace d'hypothèses C qui est un ensemble de procédures de classification de D dans {1,...,c}. En effet, si on limite le nombre de fonctions, on diminue la complexité des calculs et ceux-ci deviennent envisageables.
Soit C_{Bayes} la procédure de classification de Bayes qui est la procédure d'erreur de classification minimale dans l'ensemble de toutes les fonctions de D dans {1,...,c}. L'erreur de classification $E(C_{Bayes})$ est une borne indépassable qui représente d'une certaine manière la difficulté intrinsèque du problème. Dans la plupart des cas pratiques C_{Bayes} n'appartient pas C. Soit C_{opt} la procédure optimale de C au sens de l'erreur de classification, c'est-à-dire la procédure appartenant à l'ensemble C qui est d'erreur de classification minimale. C étant fixé, le problème est de trouver ou d'approcher C_{opt}, ce qui n'est pas facile en raison du problème de l'estimation du taux d'erreur : « On ne peut, à la fois sélectionner un classifieur à l'aide d'un ensemble d'apprentissage et juger de sa qualité avec ce même ensemble ».
En effet, pour le choix de C, la meilleure solution est de choisir la procédure C_{emp} qui minimise l'erreur apparente.
Mais, on n'a alors que peu d'indication sur l'erreur réelle $E(C_{emp})$ et donc sur la proximité de C_{emp} et C_{opt}. Les seuls résultats théoriques dont on dispose sont des résultats de convergence (en probabilité) de $E_{app}(C_{emp})$ vers $E(C_{opt})$ lorsque la taille de l'échantillon tend vers l'infini, sous certaines conditions sur C. Ce résultat a peu d'implications pratiques car on ne dispose, en général, que d'échantillons de

tailles limitées. La minimisation de l'erreur apparente ne peut donner de bons résultats que lorsque l'espace des hypothèses est bien choisi. [32] [34]

3.6.5.4 Choix de l'espace des hypothèses

Il est important de bien choisir C pour que le système puisse inférer une « bonne » solution. Pour cela, on introduit une notion de « capacité » d'un espace d'hypothèses. Dans le cas de problèmes discrets, la capacité de C est son cardinal. Dans le cas d'espaces infinis, la capacité peut être définie comme égale à la VC-dimension. Pour le choix de l'espace d'hypothèses, nous sommes confrontés au problème suivant :

- si la capacité de C est trop petite, la meilleure procédure de C appelée C_{opt} peut être éloignée de C_{Bayes} et donc, il sera impossible que le système donne de bons résultats.
- si la capacité de C est trop grande, la procédure C_{emp} qui minimise l'erreur apparente peut être éloignée de C_{opt} (erreur apparente très optimiste). Le calcul de C_{emp} peut être complexe.

La plupart des algorithmes utilisés effectuent la recherche d'une procédure qui minimise l'erreur apparente dans un espace d'hypothèses préalablement choisi. La situation est en fait plus complexe car les algorithmes utilisent des heuristiques qui orientent la recherche dans l'espace d'hypothèses. En effet, la recherche de C_{emp} peut être coûteuse en temps de calcul. Dans le cas des réseaux de neurones, le choix de l'espace des hypothèses est le choix d'une bonne architecture pour le réseau (capacité ni trop grande, ni trop petite), ensuite, on recherche une bonne solution en cherchant à minimiser l'erreur apparente.

Cependant, dans la plupart des situations pratiques, on peut considérer des suites emboîtées d'ensembles de procédures de classification $C_1 \subseteq C_2 \subseteq ... \subseteq C_k \subseteq ...$
Où : k représente une mesure de complexité du système d'apprentissage (taille des arbres de décision, taille du réseau de neurones, ...) liée à la capacité ; plus k est grand, plus la capacité de l'espace est grande.

Il faut alors trouver la valeur du paramètre de complexité k telle que $C_{k,emp}$, la procédure de C_k qui minimise l'erreur apparente, ait la plus faible erreur réelle possible. Il existe en général un bon compromis ; en effet, lorsque k augmente, l'erreur réelle $E(C_{k,emp})$ diminue lentement, se stabilise, puis croît lentement. Le bon compromis se situe dans la région où l'erreur réelle est stable. Ceci est illustré par la **Figure 3.07** représentant l'évolution des erreurs réelle et apparente dans le cas d'un système d'apprentissage pour la reconnaissance de caractères utilisant des arbres de décision.

Une solution est de minimiser l'erreur apparente en complexifiant de plus en plus l'espace de recherche. Par exemple, on peut construire des arbres de décision de plus en plus grande avec l'objectif de minimaliser l'erreur apparente. À la fin de ce processus, l'arbre obtenu est peut-être trop spécialisé (erreur apparente faible mais erreur réelle grande). On essaie alors de diminuer sa taille en diminuant l'erreur réelle. Ceci suppose que l'on soit capable d'estimer l'erreur réelle. Une telle technique peut être appliquée aux réseaux de neurones. Enfin, la méthode dite de « minimisation du risque structurel » consiste à choisir conjointement le bon compromis entre erreur réelle et capacité de l'espace d'hypothèses. Cette méthode est mise en œuvre par les machines à support de vecteurs. [31]

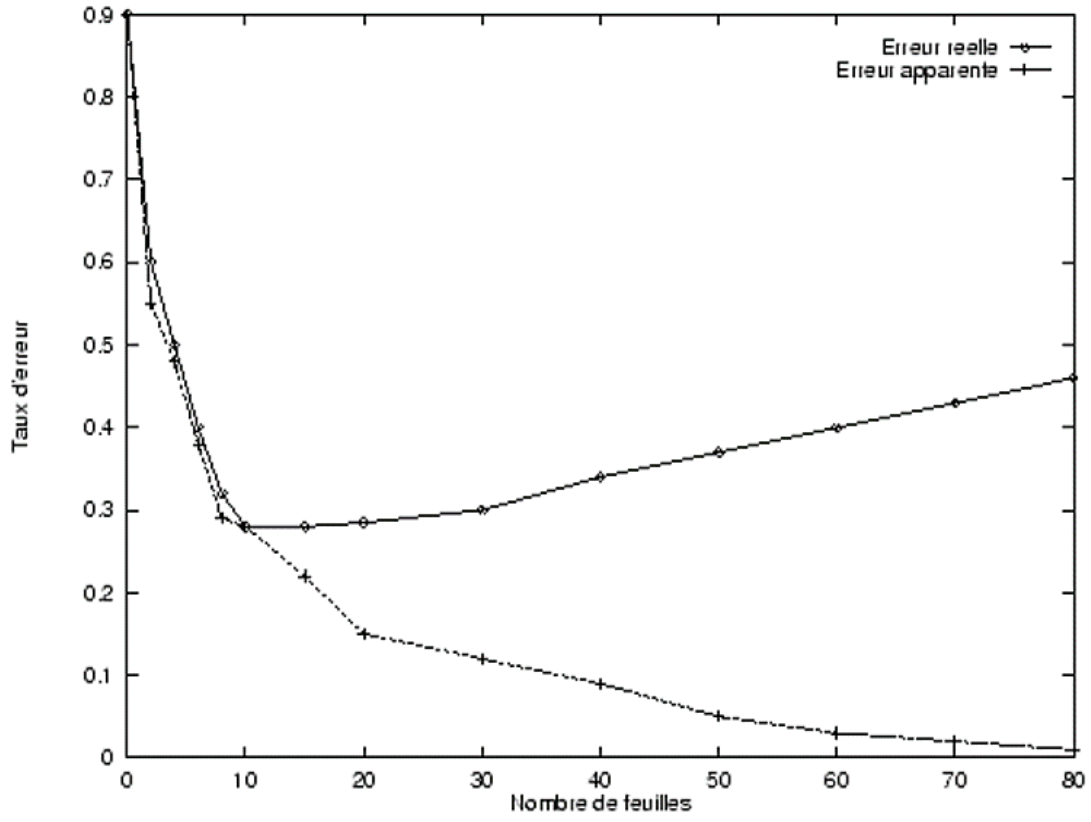


Figure 3.07 : *Erreur réelle et erreur apparente pour des données réelles de reconnaissance de caractères*

3.6.5.5 Estimer l'erreur réelle

Lorsque l'on apprend à partir d'un échantillon, se pose immédiatement la question de la pertinence statistique de la procédure induite. Supposons, par exemple, que nous disposions d'un échantillon

de 500 patients tels que 100 d'entre eux soient malades. Une procédure qui prédit toujours « bien portant » (la procédure majoritaire) fait une prédiction correcte dans 80% des cas. Tout système d'apprentissage qui prétend apporter un éclairage sur les données doit faire mieux. Il faut donc être capable d'estimer la qualité d'une procédure induite par un système à partir d'un échantillon. Nous avons également signalé, dans la section précédente, que l'estimation de l'erreur réelle pouvait être utilisée par les algorithmes pour éviter la surspécialisation. [33]

3.6.5.6 Utilisation d'un ensemble Test

L'idée dans ceci est de disposer d'un ensemble permettant de tester la qualité de la procédure de classification induite. On partitionne l'échantillon en un ensemble d'apprentissage S et un ensemble test T. La répartition entre les deux ensembles doit être faite aléatoirement. On effectue l'apprentissage à l'aide de l'ensemble S et on génère une procédure de classification C. L'estimation $\hat{E}$(C) de l'erreur réelle E(C) est alors l'erreur apparente de C mesurée sur l'ensemble test T, soit

$$\hat{E}(\mathrm{C}) = \frac{malclassés(T)}{T} \qquad (3.20)$$

Où : malclassés(T) est l'ensemble des éléments de T mal classés par la procédure C.

L'estimation est faite sur un ensemble indépendant de celui qui a servi à l'apprentissage ce qui permet de supposer que l'erreur apparente sur l'ensemble test est une bonne estimation de l'erreur réelle.

Cependant, comme la qualité de l'apprentissage augmente avec la taille de l'ensemble d'apprentissage et que de même, la précision de l'estimation augmente avec la taille de l'ensemble test, cette méthode ne donne de bons résultats que lorsque l'échantillon est « suffisamment » grand pour pouvoir être divisé en deux échantillons de tailles significatives. Il existe peu de résultats théoriques sur les tailles d'échantillon nécessaires pour utiliser cette méthode, on ne dispose que de résultats empiriques qui dépendent du problème (souvent, plusieurs centaines d'exemples). La répartition de l'échantillon entre les deux ensembles se fait en général dans des proportions 1/2, 1/2 pour chacun des deux ensembles ou 2/3 pour l'ensemble d'apprentissage et 1/3 pour l'ensemble test. Dans les problèmes réels d'extraction de connaissances à partir de données, on dispose en général de jeux de données de taille suffisante pour utiliser un ensemble test. Si l'algorithme utilisé emploi,

pour son fonctionnement, une estimation de l'erreur réelle (élagage, choix de l'architecture du réseau, réglage de paramètres), il est alors nécessaire de posséder trois ensembles :

- un ensemble d'apprentissage S,
- un ensemble test T et
- un ensemble de validation V.

Lorsque, dans l'exécution de l'algorithme, celui-ci sollicite une estimation de l'erreur réelle, on utilise l'ensemble test T. L'ensemble de validation permet, quant à lui, d'estimer la qualité de la procédure produite en sortie. [32] [33]

3.6.5.7 Re-échantillonage

Pour certains domaines d'applications où les données sont rares, il arrive que l'échantillon de travail soit trop petit pour que l'on puisse envisager de sacrifier des éléments pour tester le classifieur.
L'objectif ici est alors d'estimer l'erreur réelle d'une procédure de classification C produite par un algorithme A sur un échantillon S. Une première méthode pour estimer cette erreur réelle est la validation croisée ou cross validation.
Cette méthode est très largement utilisée. Les valeurs usuelles pour *k* sont *k*=10 (ten-fold cross validation). Il faut noter que la méthode est coûteuse en temps de calcul car il faut effectuer autant *k* sessions d'apprentissages, ce qui peut être rédhibitoire.
Cette méthode est présentée dans l'algorithme suivant où k est un paramètre :

Validation croisée - k fois

Partitionner S aléatoirement en k sous-ensembles S1,...,Sk

Pour tout i de 1 à k

Appliquer A à l'échantillon S - Si et générer Ci

Calculer $\hat{e}$i erreur apparente de Ci sur Si

Retourner $\hat{E}(C)=\sum_{1\ leq\ i \le k}\ \hat{e}_i/k$ comme estimation d'E (C)

Figure 3.08 : *Algorithme de validation croisée*

Cette méthode fournit de bons estimateurs de l'erreur réelle mais très coûteuses en temps de calcul. Elle est très utile pour les « petits » échantillons. [30]

3.7 Régression

3.7.1 Le modèle linéaire gaussien

On appelle modèle linéaire un modèle statistique qui peut s'écrire sous la forme :

On définit les quantités qui interviennent dans ce modèle,

$$Y = \sum_{j=1}^{k} \theta_j X^j + E \qquad (3.21)$$

- Y est une variable que l'on observe et que l'on souhaite expliquer et/ou prédire ; on l'appelle variable à expliquer ou variable réponse.
- Les k variables X^1, ..., X^k sont des variables réelles, non aléatoires et également observées ; l'écriture de ce modèle suppose que l'ensemble des X^j est censé expliquer Y par une relation de cause à effet ; les variables X^j sont appelées variables explicatives ou prédicteurs.
- Les θ_j (j= 1,..., k) sont les paramètres du modèle, non observés et donc à estimer par des techniques statistiques appropriées.
- E est le terme d'erreur dans le modèle ; c'est une variable non observée pour laquelle on pose les hypothèses suivantes :

  $$\mathrm{E(E)} = 0\,;\ \mathrm{Var(E)} = \sigma^2 > 0 \qquad (3.22)$$

  Où : σ^2 est un paramètre inconnu, à estimer.

En moyenne, Y s'écrit donc comme une combinaison linéaire des X^j : la liaison entre les X^j et Y est de nature linéaire. C'est la raison pour laquelle ce modèle est appelé modèle linéaire.
L'estimation des paramètres de ce modèle est basée sur n observations simultanées des variables X^j et Y réalisées sur n individus supposés indépendants. Pour la i^{ème} observation, les valeurs observées des variables sont notées y_i, x_i^1 , ..., x_i^k, de sorte que le modèle s'écrit :

$$\mathrm{yi} = \theta_j x_i^j + e_i \qquad (3.23)$$

Introduisons maintenant,

- **y:** le vecteur de IR^n composé des valeurs y_1,...,y_n,
- **X** la matrice (n,k) de rang k, contenant les valeurs observées des k variables explicatives disposées en colonnes,

- $\boldsymbol{\theta}$ le vecteur de IR^k contenant les k paramètres du modèle,
- **e** le vecteur de IR^n des erreurs du modèle.

On peut donc écrire le modèle généralement sous la forme matricielle : [35]

$$y = X\theta + e \tag{3.24}$$

3.7.2 Régression linéaire simple

On cherche à modéliser la relation entre deux variables quantitatives continues. Un modèle de régression linéaire simple est de la forme suivante :

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{3.25}$$

Où :

- y est la variable à expliquer ;
- x est la variable explicative ;
- ε est le terme d'erreur aléatoire du modèle ;
- β_0 et β_1 sont deux paramètres à estimer.

La désignation « simple » fait référence au fait qu'il n'y a qu'une seule variable explicative x pour expliquer y. Et la désignation « linéaire » correspond au fait que le modèle est linéaire en β_0 et β_1. Pour n observations, on peut écrire le modèle de régression linéaire simple sous la forme : [34]

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{3.26}$$

Où :

- ε_i est une variable aléatoire, non observée,
- x_i est observée et non aléatoire,
- y_i est observée et aléatoire.

On peut écrire matriciellement ce modèle de la manière suivante : [35]

$$Y = X\beta + \varepsilon \tag{3.27}$$

Où :

- Y désigne le vecteur à expliquer de taille n x1,
- X la matrice explicative de taille n x 2,
- ε le vecteur d'erreurs de taille n x 1.

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \quad \text{et} \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

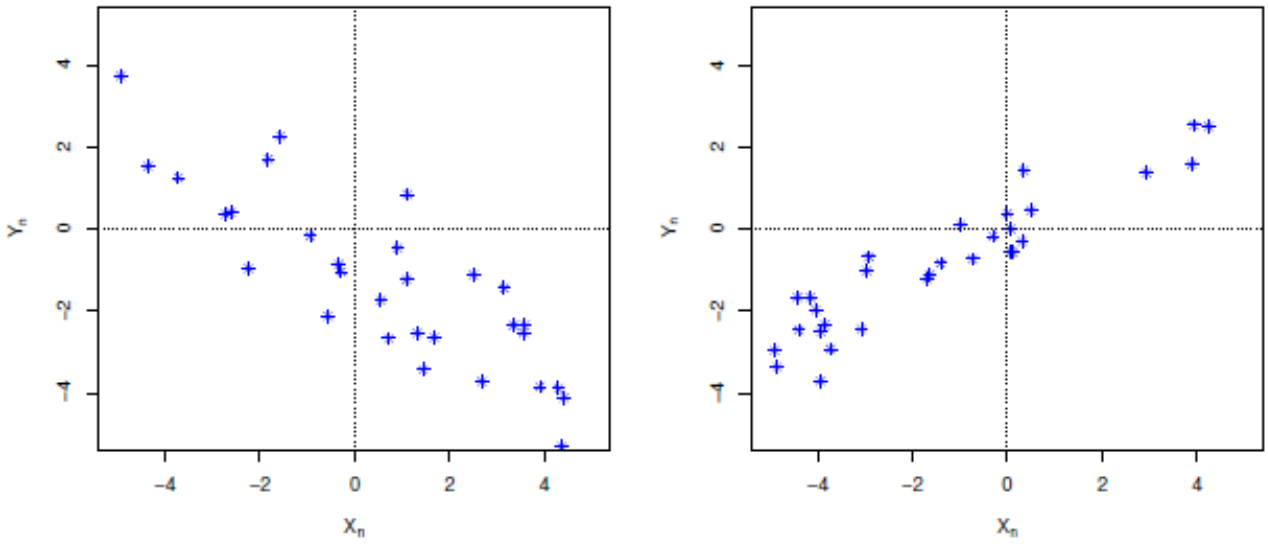


Figure 3.09 : *Exemples de deux échantillons ($x_1, \ldots, x_n$) et ($y_1, \ldots, y_n$)*

3.7.3 Régression linéaire multiple

Le modèle de régression linéaire multiple s'écrit sous la forme : [33]

$$Y = \beta_0 + \sum_{k=1}^{p} \beta_k X^k + \varepsilon \qquad (3.28)$$

Où :

- les p variables explicatives X^k sont non aléatoires réelles,
- l'erreur ε est aléatoire,
- la variable à expliquer Y est donc aléatoire.

3.7.3.1 Objectif

L'objectif est d'estimer à l'aide des données $(\beta_0, \ldots, \beta_p)$ afin de prédire la valeur moyenne de Y pour une nouvelle valeur de $(X_1, \ldots, X_p)$. [33]

3.7.3.2 Modélisation

On modélise les variables considérées comme des variables aléatoires réelles. À partir de celles-ci, le modèle de régression linéaire multiple est caractérisé par les points suivants.
Pour tout i $\in \{1, \dots, n\}$,

- $(x_{1,i}, \dots, x_{p,i})$ est une réalisation du vecteur aléatoire réel (X_1,….,X_p).
- Sachant que (X_1,….,X_p) = $(x_{1,i}, \dots, x_{p,i})$, y_i est une réalisation de

$$Y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_p x_{p,i} + \varepsilon_i \qquad (3.29)$$

Où : ε_i est une variable modélisant une somme d'erreurs. [35]

3.7.3.3 Ecriture matricielle

Le modèle de régression linéaire multiple s'écrit sous la forme matricielle :

$$Y = X\beta + \varepsilon \qquad (3.30)$$

Où :

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{p,1} \\ 1 & x_{1,2} & \cdots & x_{p,2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1,n} & \cdots & x_{p,n} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

3.7.4 Régression non-linéaire

Une régression non linéaire consiste à ajuster un modèle, en général non linéaire, $y = f a_1, \dots, a_m(x)$ pour un ensemble de valeurs $(x_i, y_i)_{1 \le i \le n}$. Les variables x_i et y_i peuvent être des scalaires ou des vecteurs. [36]

3.7.4.1 Objectif

L'objectif est de générer une équation pour décrire la relation non linéaire entre une variable de réponse continue et une ou plusieurs variables de prédiction, et aussi de prévoir de nouvelles observations. On utilise la régression non linéaire plutôt que la régression sur les moindres carrés lorsqu'on ne peut pas modéliser de manière adéquate la relation avec des paramètres linéaires. [31]

3.7.4.2 Modélisation

On modélise les variables considérées comme des variables (définies sur un espace probabilisé (Ω, A, P)). À partir de celles-ci, le modèle de régression non-linéaire multiple est caractérisé par les points suivants. [32]

Pour tout i ∈ $\{1, \dots, n\}$,

- $(x_{1,i}, \dots, x_{p,i})$ est une réalisation du vecteur aléatoire réel $(X_1, \dots, X_p)$,
- sachant que $(X_1, \dots, X_p) = (x_{1,i}, \dots, x_{p,i})$, yi est une réalisation de

$$Y_i = f(x_{1,i}, \dots, x_{p,i}, \beta_0, \dots, \beta_q) + \varepsilon_i \qquad (3.31)$$

Où : ε_i est une variable modélisant une somme d'erreurs.

Pour tout x = $(x_1, \dots, x_p) \in \mathrm{IR}^p$, sous l'hypothèse que $E(\varepsilon|\{(X_1, \dots, X_p) = x\}) = 0$ le modèle de régression non-linéaire peut s'écrire comme : [36]

$$E(Y|\{(X_1, \dots, X_p) = x\}) = Y_i = f(x_1, \dots, x_p, \beta_0, \dots, \beta_q) \qquad (3.32)$$

3.7.5 Régression logistique

3.7.5.1 Objectif

La régression logistique est une technique prédictive. Elle vise à construire un modèle permettant de prédire et/ou expliquer les valeurs prises par une variable cible qualitative à partir d'un ensemble de variables explicatives quantitatives ou qualitatives. On cherche la meilleure combinaison linéaire des données d'entrée pour modéliser la réponse, à ceci près que c'est une transformation de cette combinaison qui est utilisée en sortie. [36]

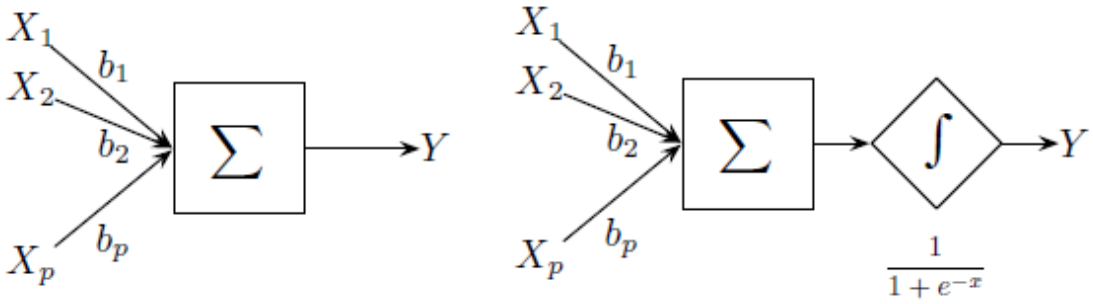


Figure 3.10 : *Parallélisme avec la régression linéaire*

3.7.5.2 Le modèle de régression logistique

Si l'on note π la probabilité d'observer l'événement y = 1, alors le log odds (transformation logit) peut s'exprimer comme une fonction linéaire des paramètres du modèle à p prédicteurs : [37]

$$g(x) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{3.33}$$

D'où : la probabilité prédite s'écrit alors : [37]

$$\widehat{y_i} = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)} \tag{3.34}$$

3.8 CONCLUSION

Le système d'apprentissage supervisé est une tâche qui permet d'extraire des informations à partir d'un ensemble des données d'entrée connues. Sur cet effet, l'objectif principal de la classification supervisée est alors d'inférer, à partir d'un échantillon de données classées, une procédure de classification. Et la régression consiste surtout à modéliser la relation entre deux variables quantitatives continues. On a vu aussi dans ce chapitre les différentes méthodes respectives de ces deux techniques cités précédemment. Pour la classification, ce sont le classifeur naïf de Bayes, méthodes paramétriques et non paramétriques, minimiser l'erreur apparente, choix de l'espace des hypothèses, estimer l'erreur réelle, utilisation d'un ensemble test, re-échantillonage. Pour la régression, régression linéaire simple, régression linéaire multiple, régression non-linéaire et régression logistique. Dans le prochain chapitre, nous allons voir la simulation d'un système optimisé d'analyse de données par le data mining supervisé.

CHAPITRE 4
SIMULATION D'UN SYSTEME D'ANALYSE DE DONNEES PAR LE DATA MINING SUPERVISE

4.1 INTRODUCTION

Ce dernier chapitre se consacrera surtout sur la partie simulation. On va simuler un test de pré analyse de données bancaire permettant ainsi de visualiser de tout près le fonctionnement d'un système d'apprentissage supervisée ainsi que ses performances. Voici les méthodes et ou algorithmes qu'on a utilisés dans notre simulation : Generalized Linear Model (Regression Logistique), Discriminant Analysis, Classification Nearest Neighbors (K Plus Proche Voisins), Naive Bayes Classification (Classifieur Naive Bayes), SVM (Support Vector Machines), Decision Trees (Arbres de decision), Ensemble Learning: TreeBagger. Ces méthodes seront ainsi comparées en performance à l'aide de la matrice de confusion, et une représentation graphique des classifications. Et en deuxième partie, on passera par une partie d'optimisation de la prédiction.

4.2 PRESENTATION DE LA SIMULATION

4.2.1 Les processus d'analyse de données

Les différentes étapes de notre travail se procèdent respectivement comme suit :

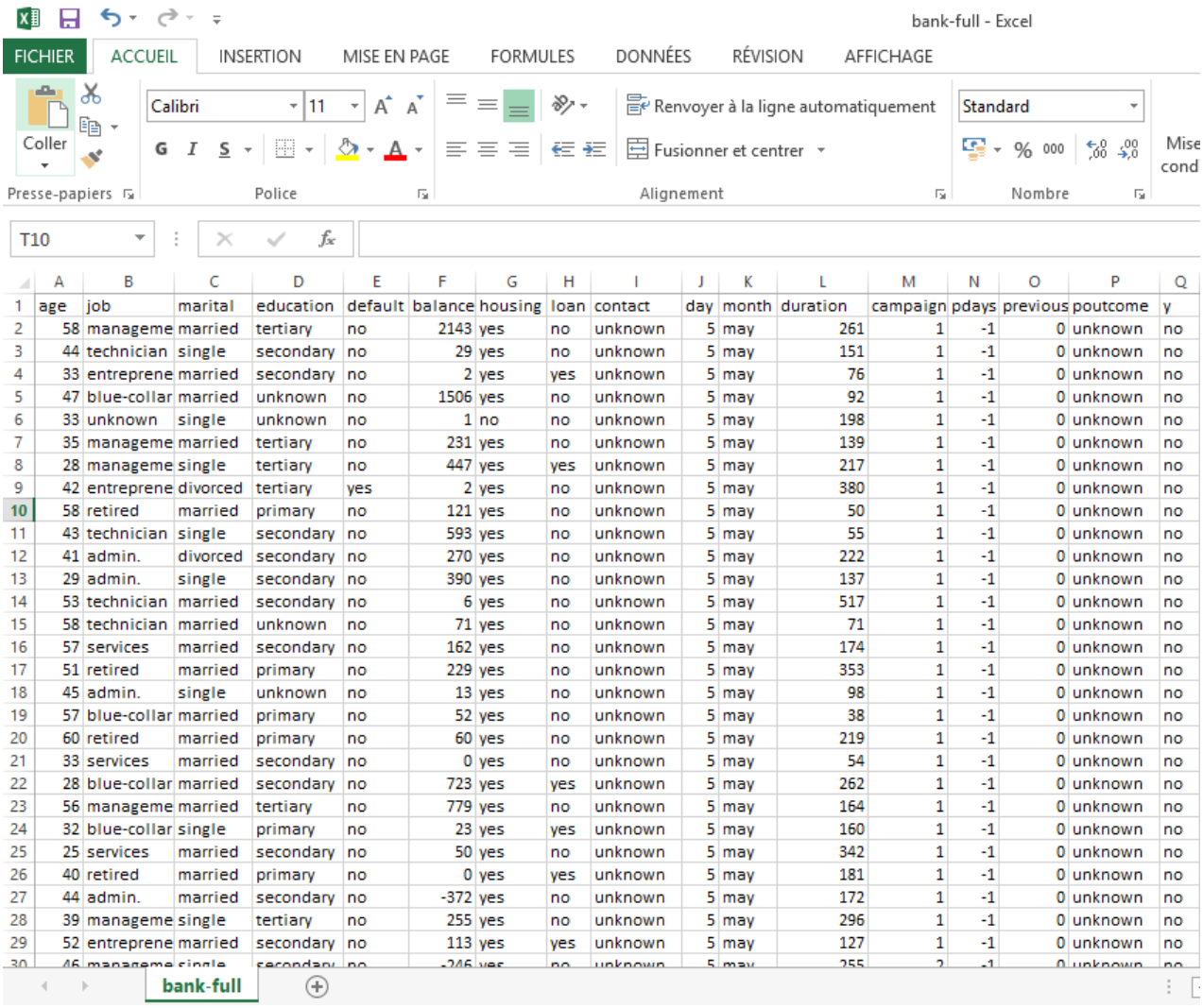
- L'importation de la donnée bancaire,
- L'exploration de la donnée,
- La conversion de notre donnée catégorielle en valeur type nominal,
- La visualisation de la sortie y du donnée en mode graphique,
- La préparation de donnée : réponses et prédicteurs,
- La validation croisée,
- Applications des différentes méthodes d'apprentissage supervisées pour la classification,
- Comparaisons de résultats de prédictions : matrice de confusions, histogramme,
- Représentation de la performance : courbe ROC de TreeBagger,
- Représentation de l'erreur de classification,
- L'estimation de l'importance des attributs et optimisation du modèle.

Telles sont les différentes étapes établies lors de la simulation. Pour une bonne représentation, nous avons développé une interface de présentation proposant ainsi les fonctionnalités suivantes.

- L’importation de donnée,
- La visualisation de donnée,
- La préparation de donnée,
- Visualisation comparaisons de résultats : matrice de confusions, histogramme,
- Représentation de la performance : courbe ROC de TreeBagger,
- Représentation de l’erreur de classification,
- Représentation de l’importance élémentaire des attributs,
- Optimisation du modèle de prédiction.

4.2.2 Description de la donnée

Voici un petit extrait de la base de données qu’on a choisi d’employer dans cette simulation.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
2	58	manageme	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
3	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
4	33	entreprene	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
5	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
6	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
7	35	manageme	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
8	28	manageme	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	unknown	no
9	42	entreprene	divorced	tertiary	yes	2	yes	no	unknown	5	may	380	1	-1	0	unknown	no
10	58	retired	married	primary	no	121	yes	no	unknown	5	may	50	1	-1	0	unknown	no
11	43	technician	single	secondary	no	593	yes	no	unknown	5	may	55	1	-1	0	unknown	no
12	41	admin.	divorced	secondary	no	270	yes	no	unknown	5	may	222	1	-1	0	unknown	no
13	29	admin.	single	secondary	no	390	yes	no	unknown	5	may	137	1	-1	0	unknown	no
14	53	technician	married	secondary	no	6	yes	no	unknown	5	may	517	1	-1	0	unknown	no
15	58	technician	married	unknown	no	71	yes	no	unknown	5	may	71	1	-1	0	unknown	no
16	57	services	married	secondary	no	162	yes	no	unknown	5	may	174	1	-1	0	unknown	no
17	51	retired	married	primary	no	229	yes	no	unknown	5	may	353	1	-1	0	unknown	no
18	45	admin.	single	unknown	no	13	yes	no	unknown	5	may	98	1	-1	0	unknown	no
19	57	blue-collar	married	primary	no	52	yes	no	unknown	5	may	38	1	-1	0	unknown	no
20	60	retired	married	primary	no	60	yes	no	unknown	5	may	219	1	-1	0	unknown	no
21	33	services	married	secondary	no	0	yes	no	unknown	5	may	54	1	-1	0	unknown	no
22	28	blue-collar	married	secondary	no	723	yes	yes	unknown	5	may	262	1	-1	0	unknown	no
23	56	manageme	married	tertiary	no	779	yes	no	unknown	5	may	164	1	-1	0	unknown	no
24	32	blue-collar	single	primary	no	23	yes	yes	unknown	5	may	160	1	-1	0	unknown	no
25	25	services	married	secondary	no	50	yes	no	unknown	5	may	342	1	-1	0	unknown	no
26	40	retired	married	primary	no	0	yes	yes	unknown	5	may	181	1	-1	0	unknown	no
27	44	admin.	married	secondary	no	-372	yes	no	unknown	5	may	172	1	-1	0	unknown	no
28	39	manageme	single	tertiary	no	255	yes	no	unknown	5	may	296	1	-1	0	unknown	no
29	52	entreprene	married	secondary	no	113	yes	yes	unknown	5	may	127	1	-1	0	unknown	no

Figure 4.01 : *Bank-full.csv*

Cette base de données de S. Moro concerne alors le résultat d'une compagne marketing direct d'un institut bancaire Portugais. Ses principales caractéristiques se situent dans le tableau suivant.

Caractéristiques des données	Multivariable	**Nombre d'instances**	45211	**Domaine**	Business
Caractéristiques des attributs	Réel	**Nombre d'attributs**	17	**Date de création**	14-02-2012
Taches associés	Classification	**Valeurs manquants**	N/A	**Nombre de téléchargement**	268635

Tableau 4.01: *Caractéristiques de « Bank-full.csv »*

4.2.2.1 Information des attributs pour les variables d'entrée

- Les données des clients de la banque

1 - age (numérique)

2 - job : catégories: 'admin.', 'bluecollar', 'entrepreneur', 'housemaid', 'management', 'retired', 'selfemployed', 'services', 'student', 'technician', 'unemployed', 'unknown'.

3 - marital : catégories: 'divorced', 'married', 'single', 'unknown'.

4 - education : catégories: 'primary', 'secondary', 'tertiary', 'unknown'.

5 - default: aura du crédit par défaut (catégories: 'no', 'yes', 'unknown').

6 - balance: balance moyenne annuel, en euros (numérique).

7 - housing: catégories: 'no', 'yes', 'unknown'.

8 - loan: catégories: 'no', 'yes', 'unknown'.

- En relation avec les derniers contacts de la compagne

9 - contact: catégories: 'cellular', 'telephone'

10 - month: dernier mois de contact de l'année (catégories: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

11 - day_of_week: dernier jour de contact de la semaine (catégories: 'mon', 'tue', 'wed', 'thu', 'fri')

12 - duration: (numérique : seconde) durée de la dernière contacte.

- Autres attributs

13 - campaign: (numérique, avec le dernier contact) nombre de contacts effectué pour un client durant la compagne.

14 - pdays: (numérique, 999 : le client n'a jamais été contacté) nombre de jour où le client n'a pas été contacté depuis.

15 - previous: (numérique) nombre de contacts effectué pour un client avant la compagne.

16 - poutcome: résultat de la dernière compagne marketing (catégories: 'failure', 'nonexistent', 'success').

4.2.2.2 Information des attributs pour les variables de sortie

y – réponse si le client a souscrit à un dépôt à terme (binaire: 'yes', 'no').

4.3 Simulation classification, prédiction et analyse de performances des algorithmes

4.3.1 Fenêtre principale

Voici la fenêtre d'accueil de notre simulation. Pour commencer la simulation, il faut cliquer sur le bouton « COMMENCER », et le bouton « QUITER » pour fermer la fenêtre.

Figure 4.02 : *Fenêtre d'accueil de la simulation*

4.3.2 L'importation de donnée

Notre simulation commence par la sélection de la base de données utilisée tout au long de notre analyse. C'est-à-dire le choix du fichier de données d'une banque nommée bank-full.csv. Une fois la base de données sélectionnée, son nom sera affiché automatiquement en bas du chemin de répertoire.

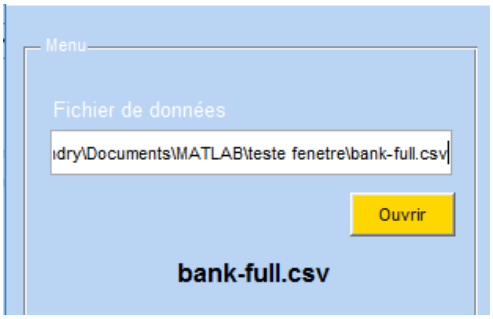


Figure 4.03 : *Sélection du fichier de données*

4.3.3 La visualisation de la donnée

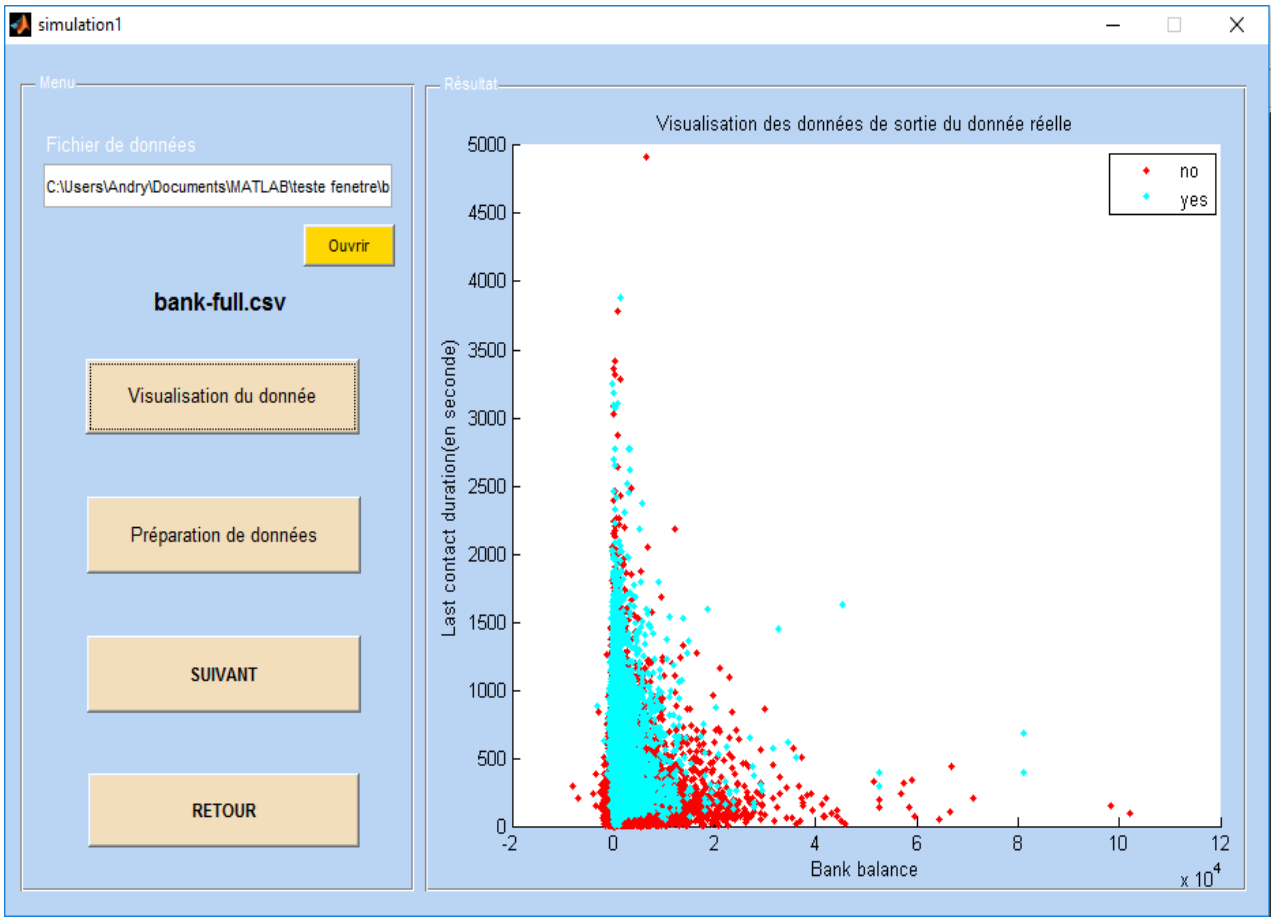


Figure 4.04 : *Visualisation du résultat de la sortie de données*

Cette étape permet de visualiser le résultat, sortie d'une compagne marketing faite dans une banque. La **Figure 4.04** précédente présente par conséquent la visualisation de notre donnée en fonction de la durée de communication avec les clients et leurs balances moyenne annuels. Et ceci permet aussi de déterminer les meilleures conditions de la compagne et d'anticiper en avance l'efficacité de la compagne face à un client de la banque.

4.3.4 La préparation de la donnée

Apres la visualisation, notre application permet de voir les détails de l'information se passant au cours de la préparation de la donnée avant la classification en apprenant une fonction objective afin de prédire la valeur future d'une classe, c'est-à-dire en effectuant une apprentissage du modèle sur le jeu de données d'apprentissage et aussi en testant ce modèle sur le jeu de données test. Voici ci-dessous, les résultats du découpage de la donnée en deux modèles qui sont les modèles d'apprentissage ou training set et celle de la donnée de test ou test set.

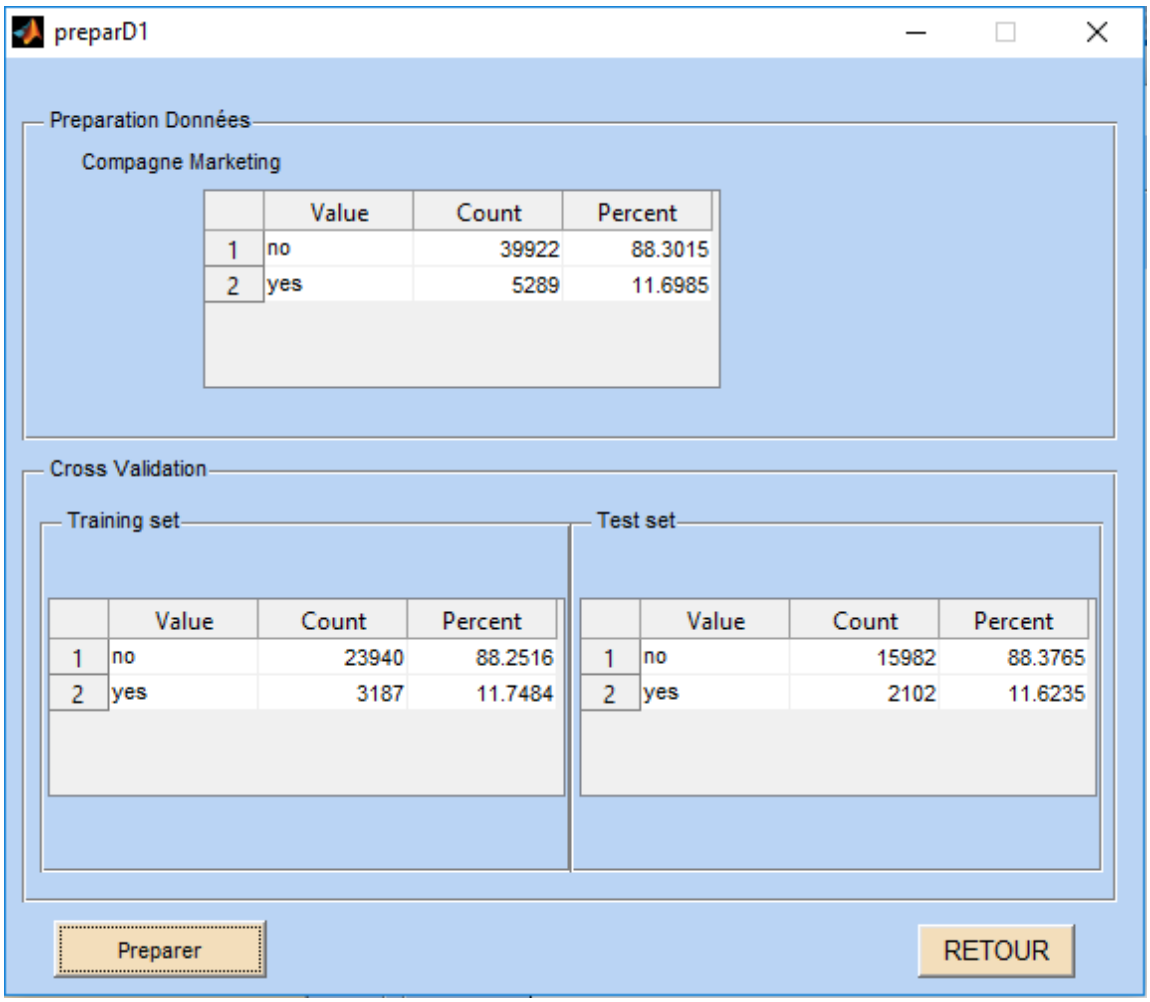
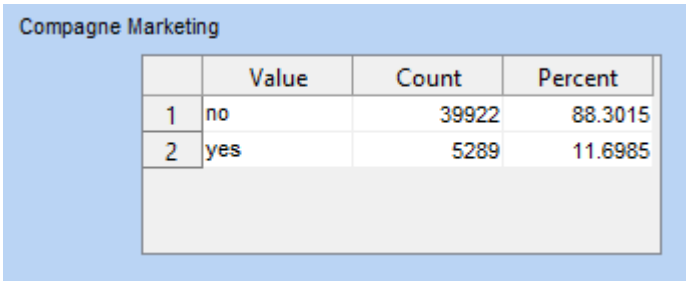


Figure 4.05 : *Interface pour la préparation de données*

En cliquant sur le bouton préparer, le calcul se lance et montre les résultats de classifications suivants :

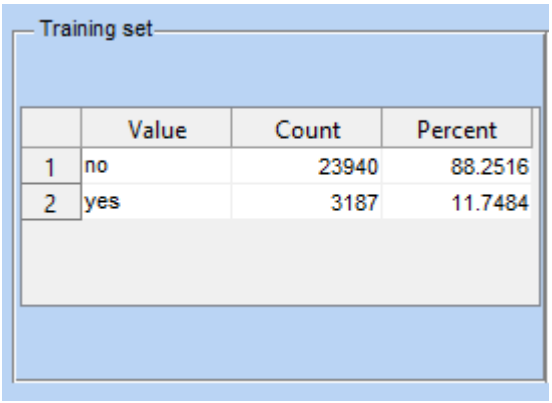

Compagne Marketing

	Value	Count	Percent
1	no	39922	88.3015
2	yes	5289	11.6985

Figure 4.06 : *Répartition du donnée réel*

Dans notre cas, on a partitionné le donnée en deux : les données d'apprentissage ou training set et les données test ou test set. Le découpage de ce jeu de données nous permettra d'avoir le modèle d'apprentissage utilisé lors de la prédiction. C'est-à-dire que la prédiction est obtenue en analysant les données test avec le modèle obtenue.

Avant d'en arriver à la génération du modèle, une étape de validation est nécessaire. Cette technique permet le découpage des données et aussi de mesurer les performances des différents algorithmes. Ici, nous avons utilisé la technique de validation croisée nommé « holdout », servant à maintenir 40% des données pour la phase de test. Les **Figure 4.06** et **Figure 4.07** montrent les résultats de ces processus.


Training set

	Value	Count	Percent
1	no	23940	88.2516
2	yes	3187	11.7484

Figure 4.07 : *Génération des caractéristiques du modèle sur le donnée d'apprentissage*

Après la création du modèle, une nouvelle valeur de prédiction sur les données test sera obtenue et celle-ci serait comparée avec les données actuelles ou réelles, et nous permettra ainsi le calcul des

matrices de confusion pour la visualisation des performances de chacun des algorithmes d'apprentissage, dans notre cas supervisé.

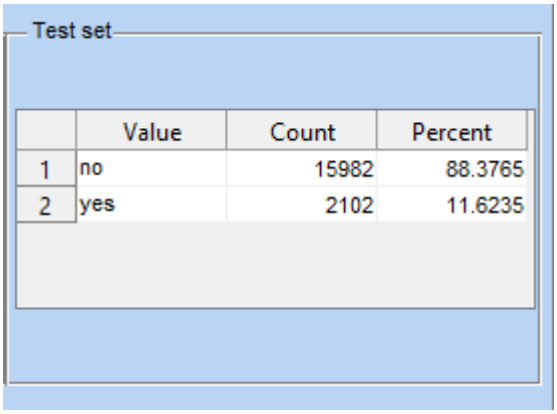


Figure 4.08 : *Répartition du donnée Test*

4.3.5 Evaluations des algorithmes : matrice de confusions et histogrammes

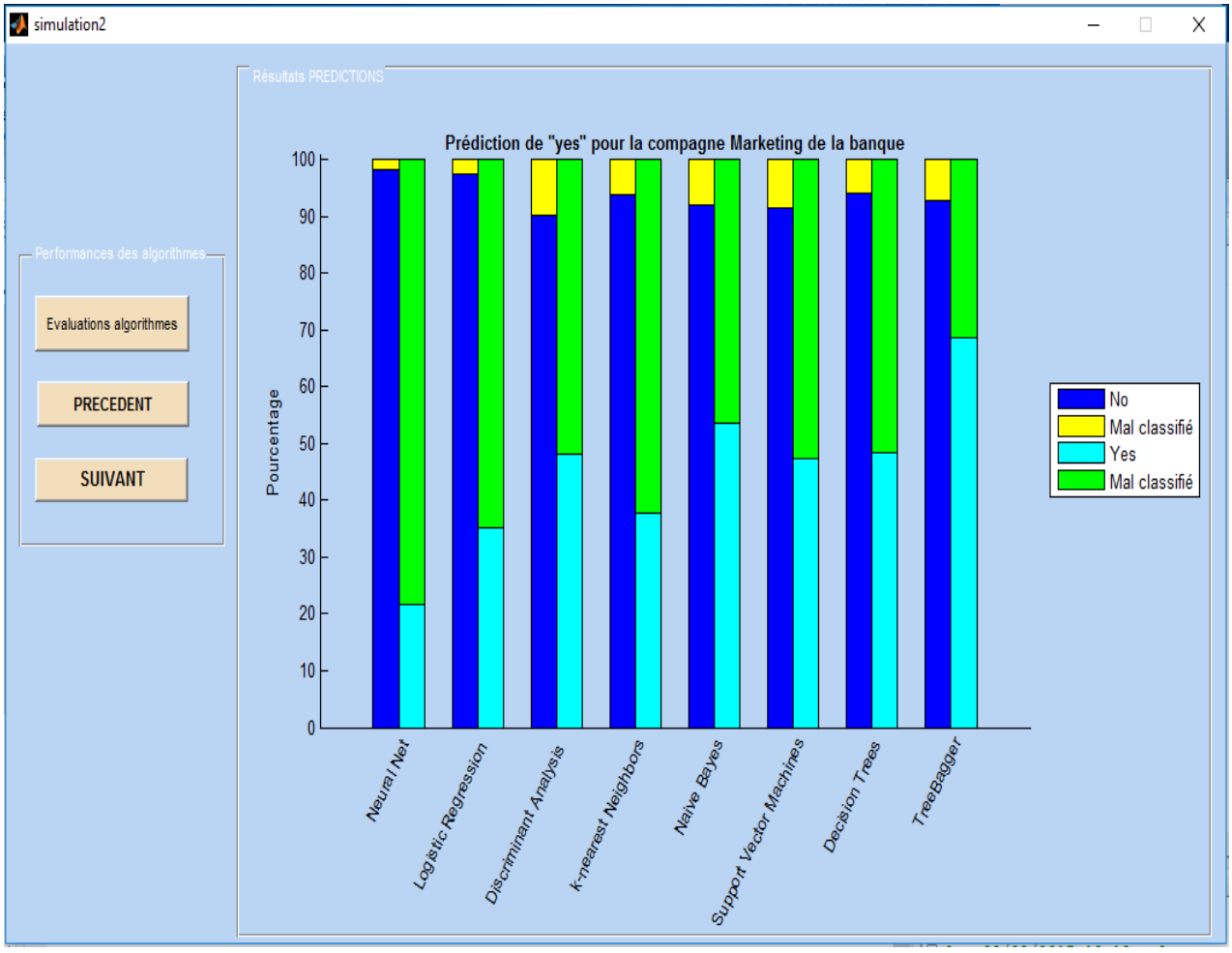


Figure 4.09 : *Comparaison des différentes classifications pour la prédiction de « yes »*

La **Figure 4.09** permet de montrer quelle méthode est efficace pour la base de données choisie. Ceci est une représentation graphique, explicitée sous forme d'histogramme, de l'analyse de performance déduit à partir des matrices de confusions des algorithmes utilisées pour la classification. Cette dernière est représentée sur la **Figure 4.10.**

En effet, on a pu constater que la méthode TreeBagger nous permet d'avoir une bonne classification en prédiction d'une réponse « yes » pour la base de données utilisée, justifié aussi par la valeur de sa matrice de confusion « C_tb ».

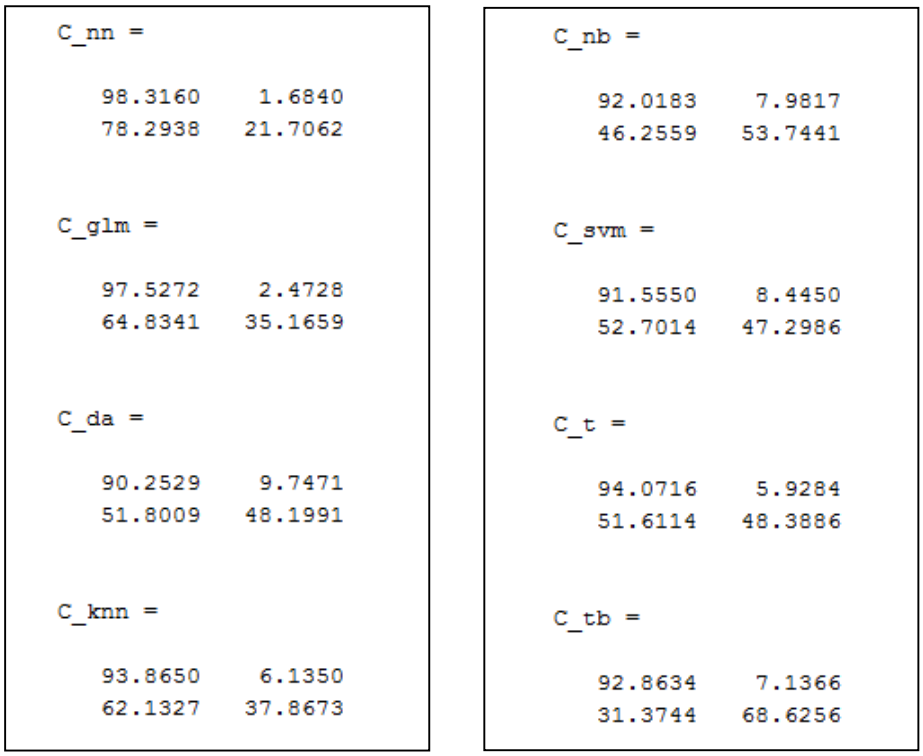


Figure 4.10 : *Matrices de confusions des différents algorithmes*

4.3.6 Représentation de la performance : courbe ROC de TreeBagger

La courbe ROC ou « caractéristique de fonctionnement du récepteur » est communément utilisée pour évaluer les performances d'un classifieur bi-classe (classe présente et classe absente).

Chaque donnée peut appartenir à la classe d'intérêt (Classe présente) ou à une autre classe (Classe absente) ; cette information est supposée connue pour toutes les données. Chaque donnée peut être affectée par le modèle à la classe d'intérêt (Classe détectée) ou à une autre classe (Classe non détectée). Si une donnée fait partie de la classe d'intérêt mais est affectée par le modèle à une autre

classe, on parle d'un « faux négatif ». Si une donnée est affectée par le modèle à la classe d'intérêt alors qu'elle fait partie d'une autre classe, on parle d'un « faux positif ».

Dans cette simulation, nous avons choisi d'évaluer la performance de l'algorithme « TreeBagger », sur laquelle se présente une courbe représentant le rapport entre RVP (Ration Vrai Positif) et RFP (Ratio Faux Positif). La sensibilité, ou le taux de vrais positifs, mesure la capacité du modèle à détecter les vrais positifs. Si tous les vrais positifs sont détectés (s'il n'y a pas de faux négatifs, c'est à dire des positifs non détectés comme positifs), alors la sensibilité est égale à 1.

La 1 - spécificité, ou le taux de faux positifs, mesure la capacité du modèle à détecter seulement les vrais positifs. Si aucun négatif n'est détecté comme positif (s'il n'y a pas de faux positifs), alors la 1 – spécificité est égale à 0.

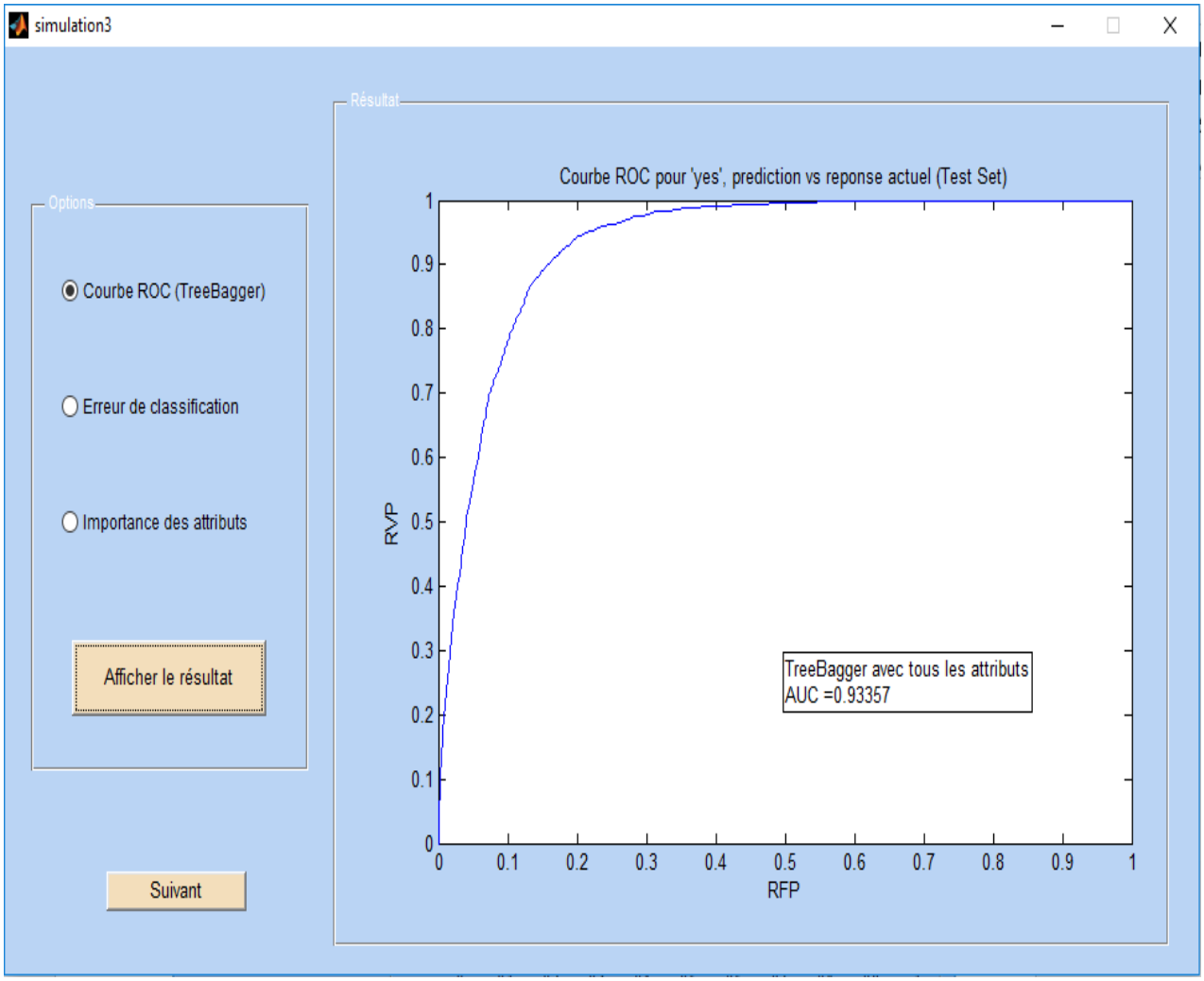


Figure 4.11 : *Courbe ROC de TreeBagger*

La **Figure 4.11** nous présente alors le résultat de la courbe ROC de l'algorithme TreeBagger. Et cette résultat nous permette ainsi d'en déduire un autre acteur de performance : AUC=0,93357. Cette dernière conclue que le classifieur TreeBagger permet au cours de l'analyse de notre donnée une bonne classification meilleure que le hasard pour la prédiction.

4.3.7 Représentation de l'erreur de classification

Le but de notre simulation c'est de pouvoir choisir une méthode efficace afin d'avoir un meilleur taux de classification permettant de prédire une valeur « yes » plutôt qu'une réponse « no ».

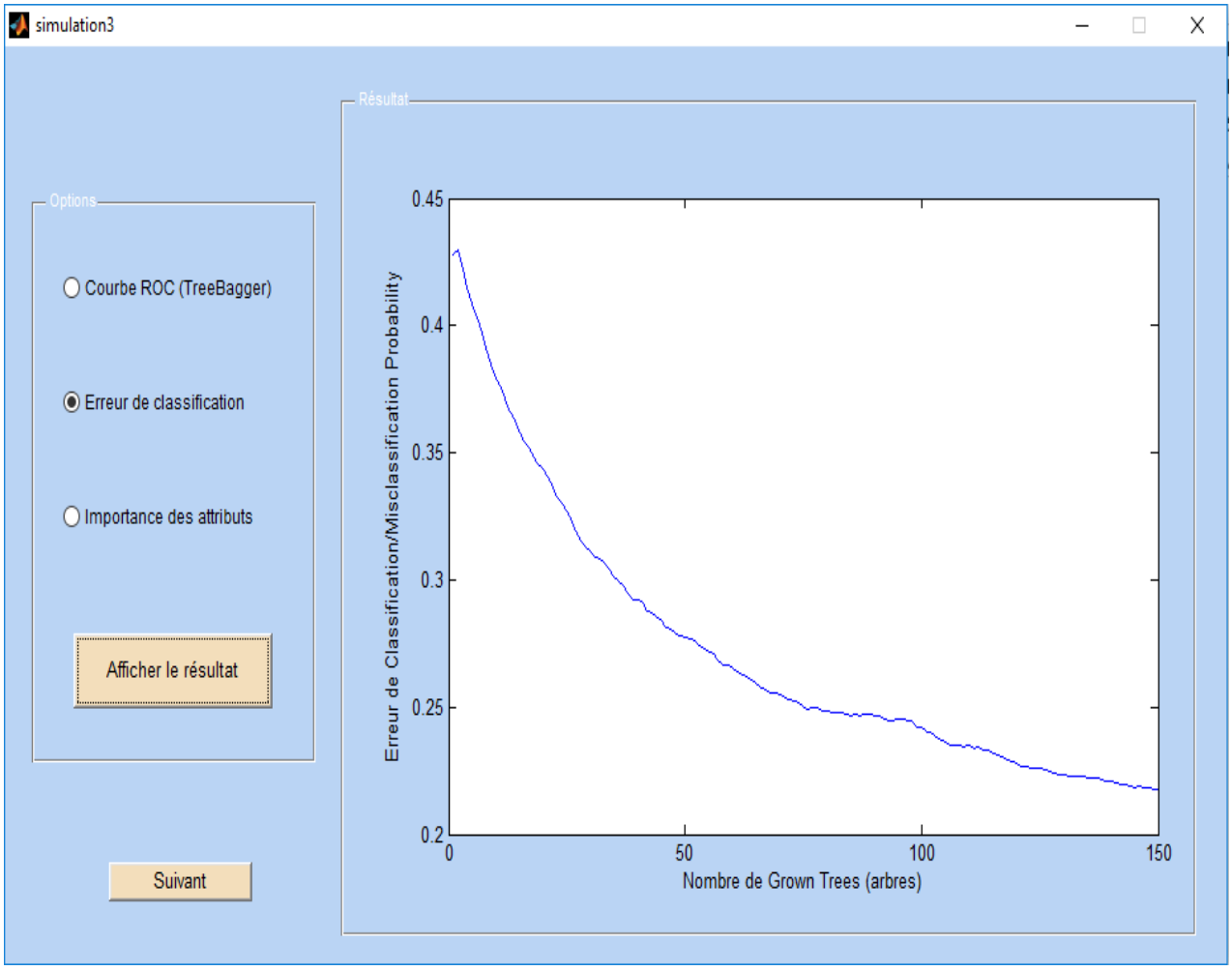


Figure 4.12 : *Erreur de classification*

La **Figure 4.12** ci-dessus nous montre que plus le nombre de ramifications de l'arbre augmente, plus on aura une faible probabilité d'erreur de classification. Ceci veut dire que le classifieur utilisait était meilleure surtout avec un nombre de ramifications au-delà de 100.

4.3.8 Estimation de l'importance des attributs

L'estimation de l'importance élémentaire des attributs est une technique d'observation permettant de mesurer l'évolution de l'erreur de prédiction. Dans notre cas, nous avons 16 attributs et la **Figure 4.13** la représente sous forme d'histogramme qui sont respectivement : age, job, marital, education, default, balance, housing, loan, contact, day, mounth, duration, campaign, pdays, previous, poutcome. Cette mesure est calculée à travers tous les ramifications de l'arbre et se calcule en divisant la moyenne de tout l'ensemble par l'écart de cet ensemble.

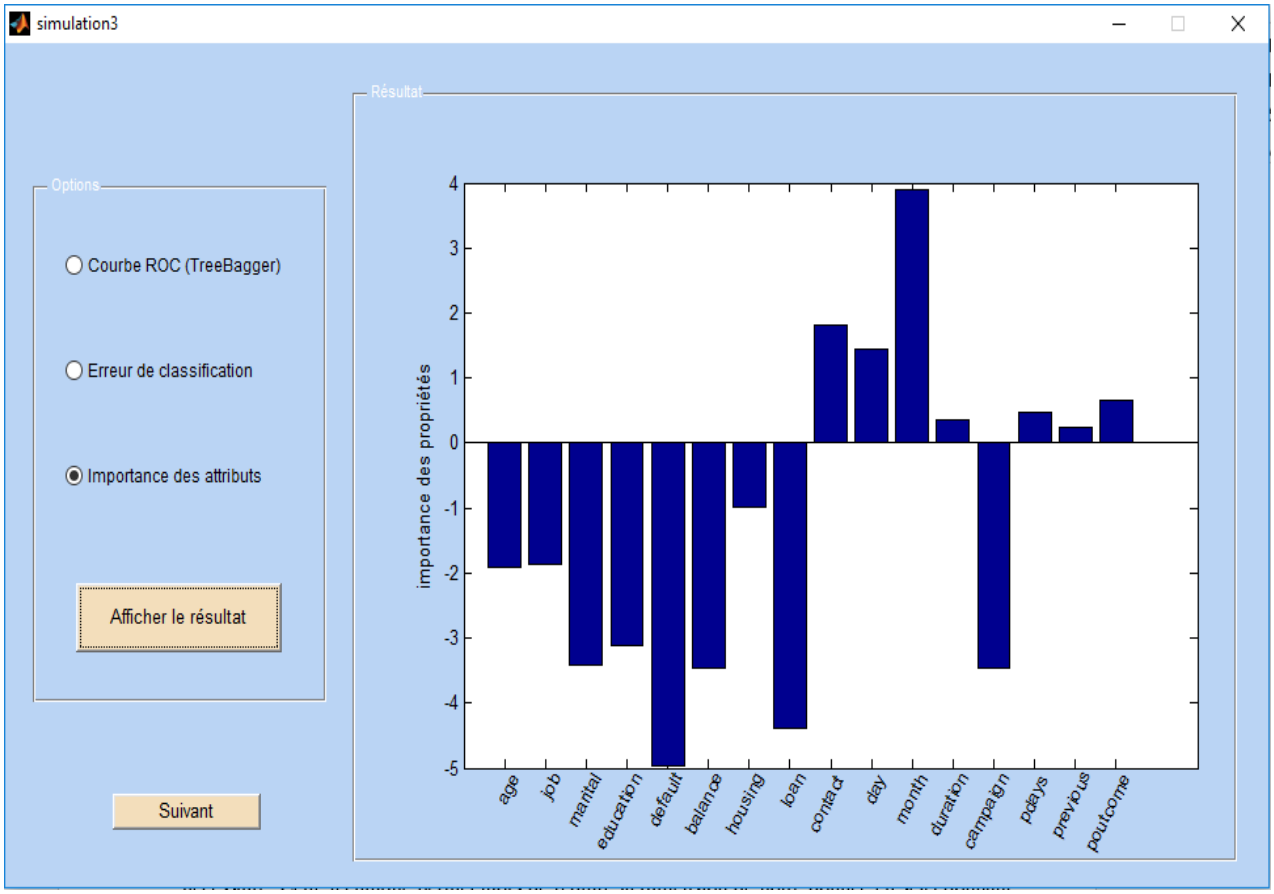


Figure 4.13 : *Importance des attributs*

4.3.9 Simplification du modèle

4.3.9.1 Réduction des attributs

Afin de simplifier notre modèle, une technique de réduction ou de sélection des attributs est nécessaire. Cette technique permet alors de réduire la dimension de notre donnée en sélectionnant seulement les variables prédicteurs pour la création du modèle et son simplification. La **Figure 4.14**

nous permet de lister les attributs sélectionnés. Dans notre cas, ils ne sont plus qu'en nombre de 14 c'est-à-dire que 2 des attributs ont été éliminés.

Pour parvenir à ce résultat, on a recouru à l'utilisation de la méthode « sequentialfs », une méthode permettant de sélectionner les meilleurs attributs X qui pourront bien être utilisé dans la prédiction d'une donnée dans Y.

Ceci peut être couteux en temps de calcul, mais elle pourra être résolue en utilisant la technique de distribution de tache parallèle (parallele computing en anglais). Sous matlab, on peut reconfigurer le paramètre de la méthode « statset ».

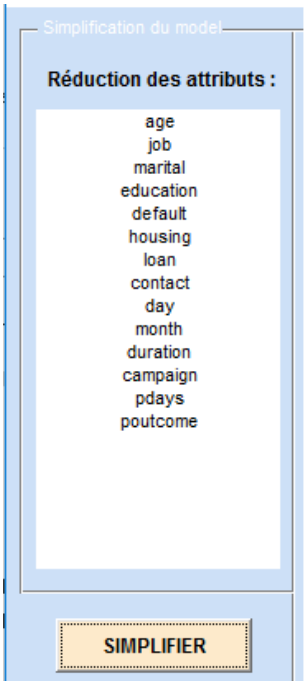


Figure 4.14 : *Sélection des attributs*

4.3.9.2 Comparaison de la classification pour TreeBagger

- Matrices de confusion

```
C_tb =

   92.8634    7.1366
   31.3744   68.6256
```

```
C_tb_r =

   92.3438    7.6562
   30.3318   69.6682
```

Figure 4.15 : *Matrices de confusion des deux modèles*

La **Figure 4.15** montre les résultats de la confrontation de la vraie valeur avec la prédiction. Le premier résultat (C_tb) était l'ancienne valeur et le deuxième est le résultat de la simplification du modèle prédit. D'après ces résultats, on constate que de même qu'on a diminué le nombre des attributs utilisés lors de la création du modèle de prédiction, la valeur de la matrice C_tb_r prouve encore qu'on obtient toujours une meilleure performance surtout en utilisant la méthode de TreeBagger.

- Histogramme

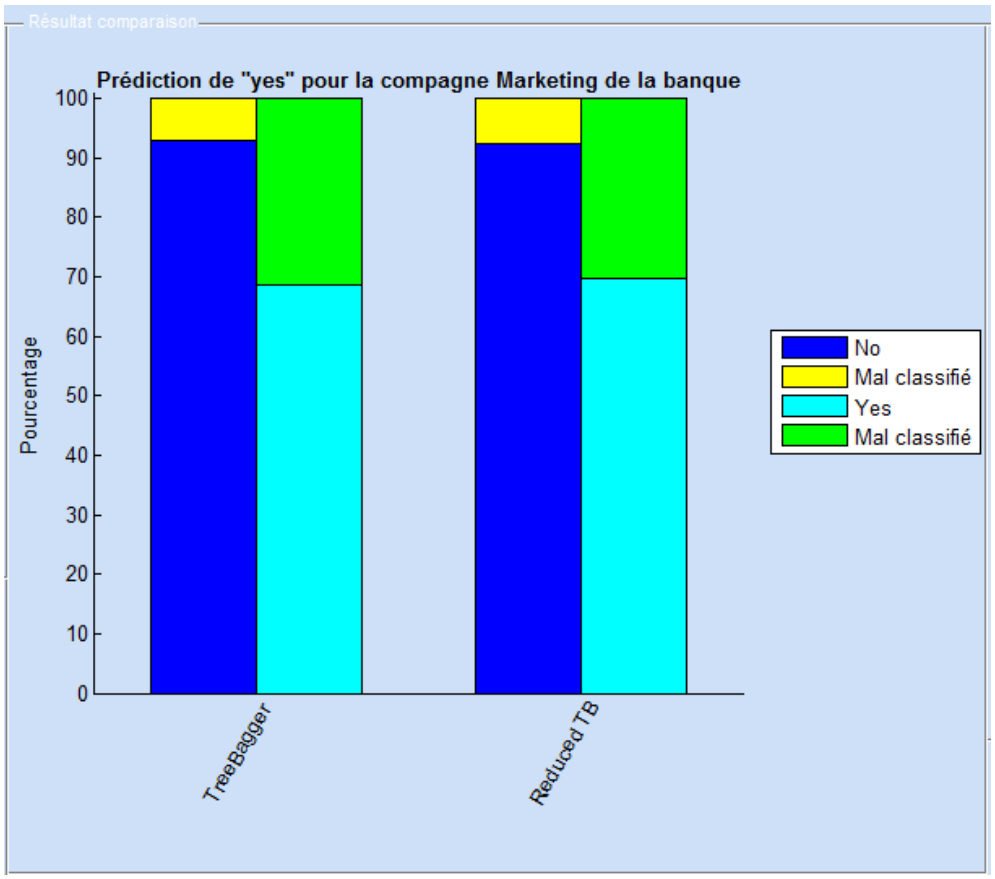


Figure 4.16 : *Comparaison de la classification pour l'anticipation d'une réponse 'yes' avec TreeBagger et Reduced TreeBagger*

La **Figure 4.16** ci-dessus représente graphiquement la comparaison entre la classification de l'ancienne prédiction et celle du modèle simplifiée. Ce résultat démontre que malgré la simplification de notre modèle avec la réduction des attributs, la performance de prédiction pour une valeur de classification « yes » de la sortie Y demeure intacte et s'améliore même en fonction de l'algorithme utilisé. En occurrence, la prédiction pour une valeur de classification « no » en sortie diminue, donnant ainsi un résultat satisfaisant pour la société.

4.3.9.3 Evaluation de l'efficacité de la classification

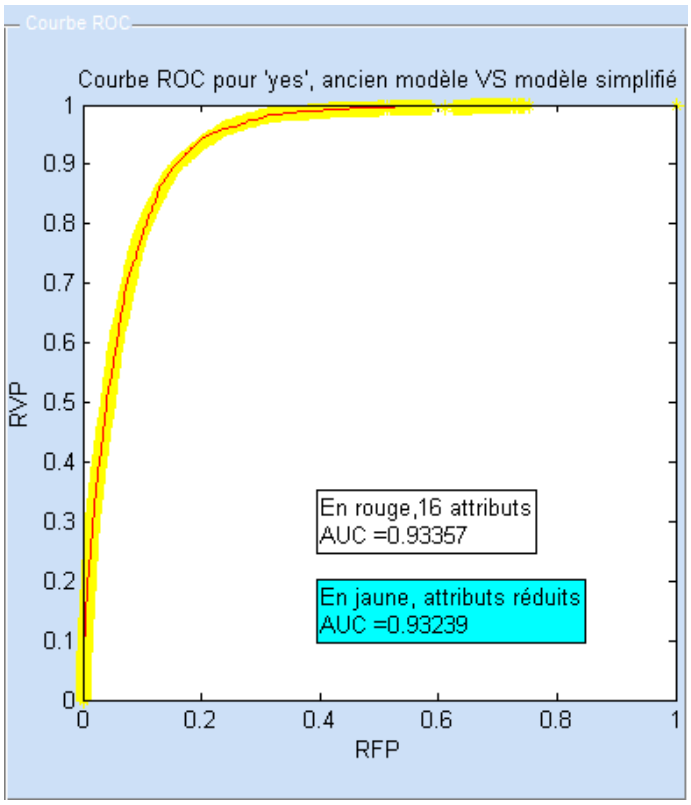


Figure 4.17 : *Courbe ROC après simplification du modèle*

Les courbes ROC représentées par la **Figure 4.17** permettent de montrer que la performance de l'algorithme a été maintenue malgré la simplification du modèle ; la courbe ROC de l'ancienne modèle est représentée en couleur rouge tandis que celle qui est simplifiée est dressée en couleur jaune. Ce résultat nous a permis aussi d'en déduire la valeur de l'AUC qui est égale à 0,93239, très proche de 1, indiquant que la classification est toujours satisfaisante. Par conséquent, le modèle simplifié est justifié fiable et optimisant pour la prédiction.

4.4 Conclusion

Nous avons vu au cours de ce dernier chapitre une simulation de l'analyse de données de marketing bancaire. Celle-ci nous a permis de visualiser les données de sortie de la base de données, sa classification, ainsi que le fonctionnement d'un système d'apprentissage supervisé pour le data mining : la construction de modèles et prédictions en fonction des différentes algorithmes utilisés. Et enfin, nous avons essayé de simplifier et optimiser le modèle de prédiction en réduisant les attributs utilisés. L'analyse de performances de prédiction et classification après cette opération nous a donné des résultats très promettant.

CONCLUSION GENERALE

Les systèmes d'information des entreprises actuelles sont de plus en plus submergés par des données de tous types : structurées, semi-structurées et non structurées. En raison de l'augmentation constante du volume d'information, la conception et la mise en œuvre d'outils efficaces, fiable et conviviale, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, devient une nécessité absolue. Pour y parvenir, plusieurs étapes devraient être exécutées. Le premier cas relève du domaine de la préparation de données, le second du domaine de l'extraction d'informations et enfin les performances de ces méthodes face à la qualité de la base de données utilisée, ainsi que son préalable optimisation.

Dans le premier chapitre, nous avons pu se familiariser sur l'extraction de connaissances à partir de données qui étant un processus non trivial permettant d'identifier, dans des données, des patterns ultimement compréhensibles, valides et potentiellement utiles. Il nécessite la préparation des données, la recherche de patterns et l'évaluation des connaissances extraites et leur raffinement. Et on a passé aussi par le Data Mining, qui soit descriptif, soit prédictif, un ensemble de méthodes et techniques destinées à l'exploration et l'analyse de grandes bases de données, de façon automatique, en vue de détecter des règles, des associations, des tendances inconnues, des structures particulières restituant l'essentiel de l'information utile.

Nous avons vu aussi dans le second chapitre quelque notion sur le système d'apprentissage automatique. Il consiste en la conception et le développement d'algorithmes permettant aux ordinateurs d'améliorer leurs performances et d'obtenir une analyse prédictive à partir de données. Ce système d'apprentissage peut se catégoriser selon le type d'algorithmes qu'il utilise tels l'apprentissage supervisé, l'apprentissage non-supervisé, l'apprentissage par renforcement.

Et puis, le troisième chapitre se focalise sur la modélisation des règles et patterns dans un système d'apprentissage supervisé. On constatait également dans ceci que l'apprentissage supervisé consiste à apprendre une fonction objective pour prédire la valeur d'une classe et il concerne essentiellement les méthodes de classification et de régression.

Et enfin, dans le dernier chapitre, on a proposé quelques simulations permettant de voir tous les étapes d'analyse de données, comme la classification, la prédiction anticipant un résultat future, l'analyse de l'efficacité d'une méthode ainsi qu'une proposition de méthode pour l'optimisation du modèle existant.

ANNEXE 1
LES SOLUTIONS DE DATA MINING

A1.1 Méthodes et outils

A1.1.1 Modèles

Construire des modèles a toujours été une activité des statisticiens. Un modèle est un résumé global des relations entre variables, permettant de comprendre des phénomènes, et d'émettre des prévisions.

Le Data Mining ne traite pas d'estimation et de tests de modèles préspécifiés, mais de la découverte de modèles à l'aide d'un processus de recherche algorithmique d'exploration de modèles:

- linéaires ou non,
- explicites ou implicites: réseaux de neurones, arbres de décision, SVM, régression logistique.

Les modèles ne sont pas issus d'une théorie mais de l'exploration des données. [19]

A1.1.2 Nouveaux gisements

A1.1.2.1 Textmining

Extraction de l'information de textes. Une part croissante de l'information se présente sous forme digitalisée: documents électroniques, nouvelles, brevets, réclamations, e-mails etc. Des techniques spéciales de classification supervisée ou non sont développées. [19]

A1.1.2.2 Webmining

Analyse de la fréquentation de sites web et du comportement des utilisateurs :

- fidélisation,
- mesures d'efficacité de campagnes de promotion,
- click analysis : optimisation des sites. [7]

A1.2 Solutions industrielles

A1.2.1 SAS Enterprise Miner

La solution de Datamining de SAS (Statistical Analysis System) est caractérisée par un référentiel partagé des modèles. Les modèles de scoring peuvent être déployés dans des environnements d'exécution divers, avec un runtime SAS ou au sein même de la base de données relationnelle.

SEM (Search Engine Marketing) propose un processus global intégré de traitement de données : échantillonnage, exploration, modification, modélisation et validation (SEMMA : Sample Explore Modify Model and Assess). [30]

Les algorithmes de Datamining suivants sont disponibles :

- Statistiques descriptives,
- Segmentation,
- Analyse de séquences,
- Analyse factorielle,
- Séries temporelles,
- Régression linéaire et logistique,
- Arbres de décision,
- Réseaux neuronaux,
- Induction de règles,
- Classification.

A1.2.2 SPSS Clementine

Autre solution client-serveur basée sur un référentiel centralisé, SPSS (Statistical Package for Social Sciences), la solution de Data Mining de l'éditeur britannique SPSS propose un panel d'algorithmes de Datamining très riche. Le modèle de processus CRISP-DM (CRoss- Industry Standard Process for Data Mining), d'initiative principalement européenne, et dont SPSS est un des principaux initiateurs, se veut un effort de standardisation de la démarche de mise en œuvre du Datamining en entreprise. On a vu que SAS proposait son propre modèle de processus, SEMMA, qui se place donc en concurrence de CRISP-DM. Ce dernier est évidemment implémenté dans SPSS et mis en avant comme un des points forts de la solution. A noté la capacité de SPSS à exploiter les algorithmes disponibles au niveau des SGBDR DB2 (Source de Gestion de Bases de Données Relationnelles), Oracle ou SQL Server (Structured Query Language Server), selon ce que proposent ces éditeurs, et les possibilités de déploiement PMML (Predictive Model Markup Language) des modèles. [30]

SPSS propose les algorithmes suivants :

- Arbres de décision
- Régression
- Segmentation
- Apprentissage bayésien

- Classification
- Réseaux neuronaux
- Induction de règles
- Régression linéaire et logistique
- Analyse factorielle

A1.2.3 Oracle Darwin

Moins riche sur le plan des algorithmes proposés que les deux précédents, mais probablement plus abordable dès lors que l'on dispose déjà du SGBDR de l'éditeur, la solution d'Oracle est présentée comme une alternative assez complète, dont l'atout principal réside dans l'intégration supposée au plus près du SGBDR et la disponibilité des données que cette intégration est censé apporter. [30]
Elle propose les algorithmes suivants :

- Réseaux neuronaux
- Régression linéaire
- Régression logistique
- Arbres de décision
- Règles d'association
- Apprentissage bayésien
- Segmentation et analyse de données exploratoire

A1.2.4 IBM Intelligent Miner

Il s'agit en réalité d'une suite de produits sous la forme d'extension des SGBDR associés à une interface de programmation (Intelligent Miner Scoring ou Intelligent Miner Modeling), de composants applicatifs (Intelligent Miner Vizualization) ou bien d'application indépendante (Intelligent Miner for Data). L'approche est similaire à celle d'Oracle du point de vue de la proximité de la solution avec le SGBDR et la simplicité relative, avec l'utilisation de la norme PMML comme format d'échange. [30]
Algorithmes proposés :

- Associations
- Classification (neuronale ou hiérarchique)
- Segmentation
- Prédiction

ANNEXE 2

BIG DATA

A2.1 Définitions

La définition s'orientait vers la question technologique, avec la célèbre règle des 3V : un grand Volume de données, une importante Variété de ces mêmes données et une Vitesse de traitement s'apparentant parfois à du temps réel. Ces technologies étaient censées répondre à l'explosion des données dans le paysage numérique. Puis, ces qualificatifs ont évolué, avec une vision davantage économique portée par le 4ème V de la définition, celui de Valeur, et une notion qualitative véhiculée par le 5ème V, celui de Véracité des données (disposer de données fiables pour le traitement).

Ces cinq éléments ont servi pendant longtemps de boîte à outils pour comprendre les fondements du Big Data, à savoir l'apparition de technologies innovantes capables de traiter en un temps limité de grands volumes de données afin de valoriser l'information non exploitée de l'entreprise. [38]

A2.2 Les acteurs de BIG DATA

De nombreux acteurs se sont positionnés rapidement sur la filière du Big Data, dans plusieurs secteurs : [30]

A2.2.1 Dans le secteur IT ou Information Technology

On trouve ainsi :

- Les fournisseurs historiques de solutions IT (ex : IBM, SAP, Oracle, HP…)
- Les acteurs du Web (ex : Facebook, Google…)
- Les spécialistes de solutions data et Big Data (ex : Teradata, MapR, Hortonworks, EMC…)
- Les intégrateurs (ex : Atos, Sopra Group, Accenture, Cap Gemini…)

A2.2.2 Dans le secteur de l'analytique

De nombreux acteurs sont également présents :

- Les éditeurs Business Intelligence (ex : SAS, Microstrategy, Qliktech…)
- Des fournisseurs spécialisés dans l'analytique Big Data (ex : Datameer, Zettaset…)

A2.3 Data Mining versus Big Data

La communication, les noms changent mais fondamentalement les méthodes restent. Le traitement des grandes masses de données, associé au "nouveau" métier de data scientist, occupe une grande place dans les médias notamment en relation avec les risques annoncés et réels du contrôle

d'internet. Beaucoup d'entreprises et de formations suivent le mouvement en renommant les intitulés sans pour autant se placer dans le cadre de grandes masses de données nécessitant des traitements spécifiques. Celui-ci devient effectif à partir du moment où le volume et le flux de données imposent une parallélisassions des tâches : les données sont réparties en nœuds, chacun associé à un processeur ou calculateur relié aux autres par un réseau haut débit au sein d'un cluster. Les mots clefs et outils de cette architecture sont Hadoop et Map Reduce, NoSQL (not only SQL, Cassandra, MongoDB, Voldemort...). Hadoop est un projet de la fondation logicielle Apache (open source en java) destiné à faciliter la création d'applications distribuées et échelonnables. Un algorithme, une méthode est dite échelonnable (scalable) si le temps de calcul est divisé par le nombre de processeurs (nœuds) utilisés ce qui permet aux applications de travailler avec des milliers de nœuds et des pétaoctets de données. Le principe, initié par Google et Yahoo, est de répartir les tâches parallèles (Map) puis d'intégrer (Reduce) tous les résultats obtenus.

Exemple très élémentaire : chaque nœud calcule la moyenne d'une variable avant de calculer la moyenne des moyennes. Bien entendu, toute méthode statistique ou d'apprentissage n'est pas scalable ou au pris d'un algorithme stochastique plus sophistiqué. D'autre part, les requêtes complexes comme celle de SQL sont impossibles. D'autres systèmes dits NoSQL, développés à l'origine par des grands sites comme Amazon, eBay, reposent également sur un système de fragmentation (sharding) des données tout en autorisant des possibilités de requêtes intermédiaires avec SQL. Bien entendu les principaux acteurs commerciaux historiques comme (Oracle) prennent position de façon complémentaire ou concurrente avec ces systèmes émergents.

Confronté à cette problématique, il appartient au statisticien data scientist de :

- s'initier aux interfaces d'accès à une architecture Hadoop ou NoSQL, notamment par l'utilisation d'outils comme Mahout ou RHadoop...
- optimiser sa stratégie : sonder dans les données et se ramener à des méthodes usuelles ou traiter les données de façon exhaustives uniquement avec une technique scalable.
- prendre en compte, ou non, les aspects temporels dus aux flux de données : estimer des modèles sur une fenêtre glissante, adopter des algorithmes adaptatifs.
- Aborder de nouveaux sur les anciens langages de programmation pour développer ou redévelopper des méthodes d'apprentissage directement parallélisables. C'est en effet ce que permettent ces langages fonctionnels par opposition aux langages objet (C, java...). [6]

ANNEXE 3
EXTRAITS DES CODES SOURCES

A3.1 Conversion de données catégoriels en nominaux, et visualisation de la donnée

```
names = bank.Properties.VarNames;
bank = datasetfun(@removequotes,bank,'DatasetOutput',true);

% Conversion
[nrows, ncols] = size(bank);
category = false(1,ncols);
   for i = 1:ncols
       if isa(bank.(names{i}),'cell') || isa(bank.(names{i}),'nominal')
          category(i) = true;
          bank.(names{i}) = nominal(bank.(names{i}));
       end
   end

catPred = category(1:end-1);
rng('default');

%visualisation
gscatter(bank.balance,bank.duration,bank.y);
  xlabel('Bank balance')
  ylabel('Last contact duration')
  title('Outcome de la compagnie')
```

A3.2 Script importation de la donnée bancaire

```
function bank1 = ImportBankData(filename, startRow, endRow)
delimiter = ';';
if nargin<=2
    startRow = 2;
    endRow = inf;
end

formatSpec = '%f%s%s%s%s%f%s%s%s%f%s%f%f%f%f%s%s%[^\n\r]';
fileID = fopen(filename,'r');
```

```
dataArray = textscan(fileID, formatSpec, endRow(1)-startRow(1)+1, 'Delimiter',
delimiter, ...
'EmptyValue' ,NaN,'HeaderLines', startRow(1)-1, 'ReturnOnError', false);
for block=2:length(startRow)
    frewind(fileID);
    dataArrayBlock = textscan(fileID, formatSpec, endRow(block)-
startRow(block)+1, 'Delimiter', delimiter, ...
    'EmptyValue' ,NaN,'HeaderLines', startRow(block)-1, 'ReturnOnError',
false);
    For col=1:length(dataArray)
        dataArray{col} = [dataArray{col};dataArrayBlock{col}];
    end
end
fclose(fileID);
bank1 = dataset(dataArray{1:end-1}, 'VarNames',
{'age','job','marital','education','default',...
'balance','housing','loan','contact','day','month','duration','campaign','pdays
','previous','poutcome','y'});
```

BIBLIOGRAPHIES

[1] F. Tomson, *« Knowledge Discovery in Databases (KDD) »*, Université de Paris, 2007.

[2] F. Denis, « *Apprentissage automatique* », Laboratoire d'Informatique Fondamentale de Marseille LIF-UMR CNRS 6166, 2005.

[3] S. Tollari, http://www.ia.lip6.fr, *« Apprentissage automatique et réduction du nombre de dimensions »*, Consulté le Novembre 2016.

[4] http://www.wikipedia.org, *« Apprentissage automatique »*, Consulté le 20 Octobre 2016.

[5] P. Fabien, *« Intelligence Artificielle »*, USTL, Février 2004.

[6] X. Lefa, *« Statistique, Apprentissage, Big–Data–Mining »*, WikiStat, 2009.

[7] M. Outtara, *« Fouilles de données: Nouvelle approche intégrant de façon cohérente et transparente la composante spatiale »*, Ouvrage, Université LAVAL Québec, 2010.

[8] M. Charrad, http://www.memoireOnline.com, *« Techniques d'extraction de connaissances appliqués aux données du web »*, Université de la Manouba, Tunis, Février 2016.

[9] P. Preux, *« Fouilles de données »*, Notes de cours, Université de Lille 3, 26 Mai 2014.

[10] P. Besse, C. L. Gall, N. Raimbault, S. Sarpy, *« Data Mining et Statistique »*, Support de cours master, Université Paul Sabatier Toulouse, 2008.

[11] G. Calas, *« Etudes des principaux algorithmes de data mining »*, Ouvrage, EPITA, 2009.

[12] S. Tufféry, *« Data mining & statistique décisionnelle »*, Ouvrage, Edition Eyrolle, Janvier 2009.

[13] A. Rakotomamonjy, G. Gasso, *« Introduction au Data-Mining »*, INSA Rouen, Département ASI, Laboratoire PSI, 2016.

[14] M. Tom, *« Au-delà de l'apprentissage supervisé »*, Ouvrage, ENST Bretagne, 2010.

[15] R. Rakotomalala, *« Apprentissage supervisé »*, Tutoriels Tanagra, Université Lyon2, 2012.

[16] F. Rosie, *« Apprentissage supervisé »*, Ouvrage, Telecom Paris Tech, Juin 2009.

[17] T. Broxen, http://www.timeislife.eu, *« Machine learning »*, Consulté le Novembre 2016.

[18] S. Tuffery, *« Cours de data mining »*, Ouvrage, Master 2, Université Rennes 1, février 2014.

[19] A. Djeffal, *« Fouille de données Avancée »*, Cours 2ème année Master Informatique de l'Optimisation et de la décision, Année universitaire: 2016-2017.

[20] P. Schwartz, http://www.develloppez.com, *« Les algorithmes génétiques »*, Septembre 2016.

[21] P. Habermehl et D. Kesner, *« Algorithmes d'apprentissage »*, Programmation Logique et IA, mars 2001.

[22] F. Santos, *« Arbres de décision »*, CNRS, UMR 5199 PACEA, 27 Mars 2015.

[23] G. Forestier, http://docslide.fr, « *Cobweb* », Novembre 2016.

[24] J. Han, M.Kamber, *« Data Mining, Concepts and Techniques »*, Edition Eyrolle, 2010.

[25] G. Calas, *« Études des principaux algorithmes de data mining »*, EPITA, France, 2009.

[26] D. Baum, J. Eckels, *« Solving Data Management and Scalability Challenges with Oracle Coherence »*, Oracle Business White Paper, Juillet 2013.

[27] L. Rouvière, *« Introduction aux methods d'agrégation: boosting, bagging et forêts aléatoires»*, Support de cours, Université Rennes *2, 2009.*

[28] M. Jeremy, *« Data Mining avec Weka »*, Polytechnique Montréal, 2013.

[29] S. Lallich, E. Prudhomme, *« Représentation des données pour l'apprentissage supervisé »*, Université Paris 13, Institut Galilée, avril 2008.

[30] E. Bokhari, *« D*a*ngerous predictions: evaluation methods for and consequences of predicting dangerous behavior »*, Dissertation, University of Illinois, Urbana-Champaign, 2014.

[31] P. Preux, *« Fouille de données-Approches supervisées »*, Ouvrage, 2011.

[32] K. Tannir, *« Evaluation des performances de regroupements de données avec Apache-Mahout »*, Ouvrage, 2011.

[33] M. Crucianu, P. Cubaud, R. Fournier, *« Apprentissage supervisé à large échelle »*, Cours CNAM 2016.

[34] G. Lebrun, *« Sélection de modèles pour la classification supervisée avec des SVM* », Ecole Doctorale SIMEM, Novembre 2006.

[35] G. Bouchard, *« Les modèles génératifs en classification supervisée et applications à la catégorisation d'images et à la fiabilité industrielle »*, Thèse, INRIA, 2005.

[36] Altman, *« Régression logistique »*, Ouvrage, Introduction à R pour la recherche biomédicale, 2012.

[37] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *« Classification and regression Trees »*, Rapport technique, Wadsworth International, Montere, 1984.

[38] B. Laffargue, *« Guide du big data»*, Annuaire de reference, Corp events, 2014.

FICHE DE RENSEIGNEMENTS

Nom : RASOANAIVO
Prénom : Andrianirina
Tel : +261 34 46 298 00
E-mail : rasoanaivoandrianirina@gmail.com
Adresse de l'auteur : Lot O2-F-78 Tomboarivo Mahafaly
Antsirabe 110 – Madagascar

Titre du mémoire :
« IMPLEMENTATION ET PERFORMANCE D'UN SYSTEME D'ANALYSE DE DONNEES PAR LE DATA MINING SUPERVISEE ».

Nombre de pages : 77
Nombre de tableaux : 4
Nombre de figures : 45

Directeur de mémoire :
Nom : RAMAFIARISONA
Prénoms : Malalatiana Hajasoa
Grade : Maitre de conférences, Enseignant-chercheur
Tel : +261 34 16 542 48
E-mail : mhramafiarisona@Yahoo.fr

RESUME

L'objectif de ce mémoire est la mise en œuvre, l'étude de performances et l'optimisation d'un système d'analyse de données et d'extraction d'informations, satisfont les besoins des utilisateurs suivant les critères de fiabilité et de convivialité quel que soit les types de base de données utilisées. En s'appuyant sur les algorithmes d'apprentissage supervisé surtout celle de Tree Bagger, ce système permet de classifier les données selon les modèles générés au cours de l'apprentissage, et fourni ainsi un résultat de prédiction, et de performances. Ceci est toujours attendu dans plusieurs domaines à savoir la Télécommunication, le business intelligente, le domaine médicale, etc.

Mots clés :

Fouille de donnée, apprentissage automatique, prédiction, classification, régression, estimation.

ABSTRACT

The objective of this memory is the implementation, performances and the optimization of a data system's analysis and an effective information's extraction, in wich satisfy user's needs according to the criteria of reliability and conviviality whatever the basic types of data used. While being based on the supervised learning algorithms endeed Tree Bagger's method, this system insure to classify data according to models' generated during the training, and thus provided a prediction and performances. This is always used in several fields such as Telecommunication, business intelligent, the medicals' field, and so on.

Keywords:

Data mining, machine learning, prediction, classification, regression, estimating.