

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

**FACULTE DE MATHEMATIQUE ET
INFORMATIQUE**

DEPARTEMENT : INFORMATIQUE

N° :



FILIERE : INFORMATIQUE

**OPTION : SYSTEME D'INFORMATION
ET GENIE LOGICIEL.**

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par : Amira Bouguerra

Amina Amroune

**Prédiction du trafic internet dans l'université de
M'sila**

Soutenu devant le jury composé de :

Dr. Mokhtari rabah

Université de m'sila Président

Dr. Mehenni Tahar

Université de m'sila Rapporteur

Dr. Heraguemi Kamel Eddine

Université de M'sila Co-Rapporteur

Dr. Nouioua laila

Université de m'sila Examineur

Année universitaire : 2020 /2021

Remerciement

Avant tout,

Je remercie **ALLAH** qui nous a donné la force,

Le courage et l'espoir nécessaire pour accomplir ce travail.

Ce travail aussi modeste.

N'a été rendu possible que grâce aux orientations éclairées de notre encadreur : Monsieur **Mehenni Tahar** que nous tenons à lui exprimer notre parfaite gratitude et nos sincères remerciements pour la qualité de son encadrement et pour ses conseils judicieux et avisés.

Nous tenons à remercier notre Co-encadreur Mr Heraguemi Kamel Eddine pour sa collaboration notamment dans l'acquisition des données du trafic internet de l'université, Sans oublier toute l'équipe du centre de réseaux de l'université.

Nous remercions également les membres de jury pour avoir fait le plaisir d'accepter d'examiner ce travail.

Nous dédions ce mémoire à tous ceux qui ont contribué à notre formation et qui nous ont soutenus dans nos études.

DEDICACE

A mes chers parents

Quoi que je fasse ou que je dise, je ne saurai point vous remercier comme il se doit, votre affection me couvre, votre bienveillance me guide et votre présence à mes cotés a toujours été ma source de force.

Que vous trouverez ici le témoignage de mon attachement, ma reconnaissance, gratitude et respect, que dieu vous préservent bonne santé et longue vie.

A Ma chère sœur Azhar

Que ce travail soit l'expression de ma profonde affection. je te remercie pour le soutien moral et l'encouragement que tu m'as accordé, je te souhaite tout le bonheur que tu mérites, en te souhaitant un brillant avenir.

A mon binôme Amina amroune Pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Tous les gens m'aiment surtout : Ma belle Nourelhouda et yassmina ...

A tous ceux que j'aime. Merci!

« *Amira bouguerra* »

DEDICACE

Je dédie humblement ce modeste travail avec ma sincère reconnaissance à :

Ma Mère ali zeghlache naima ... Tu représentes pour moi la bonté, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prière pour moi.

Mon cher Père abdelkarim ... Rien au monde ne vaut les efforts fournis jours et nuit pour mon éducation et mon bien être

A Mes chères sœurs Loubna ,israa,assma et alae et mon unique frère mohamed
Que dieu vous garde avec mes sentiment de fraternité et toute la réussite.

Je le dédie aussi à tous mes oncles et tantes, cousins et cousines.

A tous mes amies et mes collègues de la promo et surtout ma belle manel et ma chérie houda

A mon binôme amira bouguerra

A mon encadreur Dr. MEHENNI tahar et à mes chers enseignants sans aucune exception.

Enfin je le dédie à tous ceux qui me connaissent.

« *Amina Amroune* »

Table des matières

Remerciement.....	3
Liste des abréviations.....	9
Liste des figures.....	10
Liste des tableaux	11
Chapitre 1 Trafic internet.....	17
1.1 Caractéristiques de trafic internet :	18
1.1.1 Non-stationnarité :	18
1.1.2 Autosimilarité et dépendance à long terme :	19
1.1.3 Distribution à décroissance lente :	21
1.1.4 La sporadicité du trafic :	21
1.2 Analyse de trafic internet :	22
1.2.1 Analyse spectrale (recherche des périodicités) :	22
1.2.2 Analyse des dépendances temporelles :	22
1.3 Conclusion :	24
Chapitre 2 Modèles de prévision : Algorithmes de classification	25
2.1 Introduction	25
2.2 Définition de Data Mining :	25
2.3 Les tâches de Data Mining :	25
2.4 Les méthodes de Data Mining :	26
2.5 Domaines d'utilisation de Data Mining :	27
2.6 Classification automatique :	27
2.6.1 Apprentissage non supervisé (Clustering) :	27
2.6.2 Apprentissage supervisé (Catégorisation) :	28

2.6.3	Avantages et inconvénients :	28
2.7	Algorithmes d'apprentissage :	28
2.8	k-plus proches voisins (k-nearest neighbor):	29
2.9	L'algorithme k-NN :	29
2.9.1	Principe de fonctionnement :	30
2.9.2	Critiques de la méthode :	31
2.9.3	Les domaines d'application :	31
2.10	Arbre de decision:	31
2.10.1	Algorithme ID3 :	33
2.10.2	Algorithme C4.5 :	34
2.10.3	Algorithme Classification And Regression Trees (CART) :	34
2.10.4	Les domaines d'application :	34
2.11	Naïve bayes :	35
2.11.1	Description du modèle Bayésien :	35
2.11.2	Estimation de la valeur des paramètres :	37
2.11.3	Construire un classifieur à partir du modèle de probabilités :	38
2.11.4	Analyse :	38
2.12	Machines à support de vecteurs (ou SVM) :	39
2.12.1	Définition :	39
2.12.2	Principe de fonctionnement :	40
2.12.3	Domaine d'application :	40
2.13	Réseau de neurones (Réseau neuronal) :	41
2.13.1	Architecture du neurone formel :	41
2.13.2	Principe de fonctionnement d'un neurone formel :	42
2.13.3	Fonctionnement :	43
2.13.4	Domaine d'application :	43
2.14	Les points forts et les points faibles :	44

2.15	Evaluation des méthodes de classification :	45
2.16	Conclusion :	48
Chapitre 3 Etat de l’art		49
3.1	Introduction :	49
3.2	Revue de la littérature :	49
3.3	Conclusion :	52
Chapitre 4 Implementation et realization		53
4.1	Introduction :	53
4.2	Acquisition des données et méthodologie :	53
4.2.1	Fichier LOG :	53
4.3	Présentation de l’espace de travail :	55
4.3.1	Présentation de la plateforme WEKA :	56
4.3.2	Présentation de la plateforme R : que R ?	56
4.4	Implémentation et Réalisation :	58
4.4.1	Le processus de classification par la méthode arbre de décision avec WEKA :	58
4.4.2	Le processus de classification par arbre de décision avec R	64
4.4.3	Le processus de classification par la méthode de Naïve Bayes avec WEKA	67
4.4.4	Le processus de classification par des méthodes naïves bayes à travers R :	68
4.5	Conclusion :	71
Bibliographie		73

Liste des abréviations

ACF : Fonction d'autocorrélation

PACF : Fonction d'autocorrélation partielle

AD : Les arbres de décision.

SVM : Machines à support de vecteurs

RN : Les réseaux de neurones (RNA).

NB : L'algorithme de Naïve Bayes.

CT: classification automatique.

K-NN: k-nearest neighbor.

PPV : plus proches voisins

CART : Algorithme Classification And Regression Trees

WEKA: Waikato Environment for Knowledge Analysis.

API : Application Programming Interface.

CLI : interface de ligne de commande.

ACP : l'analyse en composantes principales.

ARFF : Attribute Relation File Format.

Liste des figures

Figure 2-1 Les méthodes de Data Mining	26
Figure 2-2 K-plus proche voisin	31
Figure 2-3 l'arbre de décision.....	32
Figure 2-4 vecteurs de support.....	39
Figure 2-5 Les vecteurs à support	40
Figure 2-6 Architecture du réseau de neurone formel.....	42
Figure 2-7 modèle de McCulloch et Pitts/Modèle linéaire.....	42
Figure 4-1 interface de logiciel WEKA	56
Figure 4-2 Interface de R sous Windows	58
Figure 4-3 Sélection d'un fichier dans WEKA	59
Figure 4-4 Choix de classifieur WEKA	60
Figure 4-5 résultat des mesures de classification.....	61
Figure 4-6 la méthode pour afficher l'arbre WEKA	62
Figure 4-7 l'arbre résultat WEKA	63
Figure 4-8 la page Principale de R.....	64
Figure 4-9 résultat de L'exécution de l'algorithme arbre de décision en R.....	65
Figure 4-10 résultat des mesures de classification en R.....	65
Figure 4-11 L'arbre de décision en R.....	66
Figure 4-12 résultat da classificateur Naivebayes.....	68
Figure 4-13 l'algorithme de Naivebayes sur R.....	69
Figure 4-14 résultat des mesures de classification par Naivebayes	69

Liste des tableaux

Tableau 2-1 Les points forts et les points faibles	45
Tableau 2-2 Matrice de confusion	45
Tableau 4-1 Extrait de fichier log.....	54
Tableau 4-2 Les données de trafic internet	55
Tableau 4-3 classification de la taille du données de trafic internet	55
Tableau 4-4 extrait en fichier ARFF.....	59
Tableau 4-5 matrice de confusion de l'arbre WEKA	61
Tableau 4-6 matrice de confusion en R.....	66
Tableau 4-7 Comparaison des statistiques de Precision ,Recall et F_measure	67
Tableau 4-8 les résultats de R et WEKA.....	67
Tableau 4-9 matrice de confusion de Naivebayes en WEKA.	68
Tableau 4-10 matrice de confusion de Naivebayes en R	70
Tableau 4-11 comparaison de performance de algorithme de Naivebayes sur plateforme WEKA et R	70
Tableau 4-12 Comparaison des statistiques de Precision ,Recall et F_measure en R	70

INTRODUCTION GENERALE

L'Internet est devenu le média universel pour tous les types d'informations tels que les données et la voix, la vidéo et les applications temps-réel.

Sa Croissance et son évolution rapide en taille et en complexité ne permettent pas une maîtrise complète du réseau et de son comportement. Par conséquent, les opérateurs réseaux utilisent des techniques de métrologie pour la mise en place et la maintenance de leurs réseaux de communication afin d'offrir les meilleurs services à leurs clients et d'optimiser au maximum l'utilisation des ressources.

La métrologie de l'Internet, ou science de mesure de l'Internet, vise à superviser, mesurer et modéliser le trafic Internet afin d'évaluer la performance des réseaux en termes des différents critères de qualité de service.

Elle permet, entre autres, de comprendre le fonctionnement des divers protocoles qui gèrent le réseau et d'analyser la manière avec laquelle ces protocoles interagissent entre eux.

Dans **le premier chapitre** vise à présenter la prévision du trafic a été intensivement étudiée depuis la découverte de la nature *auto-similarité* et de la *dépendance à long terme* (LRD) du trafic des différents types de réseaux.

En effet, il est établi que le trafic présente de forte autocorrélation Bien que ces caractéristiques particulières causent la dégradation de la performance du réseau en termes de perte et de délai, plusieurs études ont prouvé que l'auto-similarité peut être exploitée pour caractériser et prévoir le trafic afin de contrôler le réseau.

La prévision du trafic peut être classée selon la taille de l'intervalle de temps pour lequel elle est effectuée. Cet intervalle est appelé *horizon* de la prévision. Ainsi, on peut distinguer la prévision à court terme, à moyen et à long terme.

Le **deuxième chapitre** introduit des notions générales sur le domaine de : Data Mining, en donnant quelques définitions, les taches principales, les applications de chacun et surtout présentation des différents algorithmes d'apprentissage automatique supervisée ainsi que leurs avantages et leurs inconvénients. Nous avons également introduit les différents moyens d'évaluation d'un classificateur.

Le troisième chapitre présente le contexte de notre projet. Dans un premier temps, les étapes de développement de trafic internet et résumé des articles universitaire utilisés dans le domaine du trafic internet.

Le **quatrième chapitre** permettra d'évaluer les performances des différentes approches implémentées en présentant les résultats obtenus avec interprétation. Et nous avons terminé par une conclusion que nous allons voir ultérieurement.

Chapitre 1

Trafic internet

1. Définition de Trafic Internet :

Les réseaux de communication (téléphonie, Internet, réseaux étendus, réseaux locaux, etc.) ont connu, au cours des dernières années, un développement extraordinaire. Pour leurs opérateurs, une question centrale est de savoir contrôler les flux d'information de façon optimale, afin d'éviter tout problème de congestion ainsi que des problèmes matériels et d'offrir aux utilisateurs un service de bonne qualité, fiable et rapide. Or pour concevoir des procédures efficaces de contrôle de la circulation des informations, pour bien spécifier les équipements matériels nécessaires, une connaissance bien approfondie des propriétés du trafic des communications dans de tels réseaux s'impose.

Mais les réseaux de communication d'aujourd'hui ne sont plus ceux d'hier. Internet a connu une expansion phénoménale ces cinq dernières années (on estime que le trafic de communications vocales qui représentait 90 % du trafic global en 1997, puis représentait 50 % en 2000, n'en représentera que 10 % d'ici 2008) [3] .

Cet essor a radicalement changé une situation qui était stable depuis plus d'un demi-siècle. Les raisons profondes de ce développement rapide résident dans l'utilisation, pour l'acheminement de l'information et le contrôle du trafic, de nouveaux protocoles de routage (routage IP, pour Internet Protocol) et de contrôle (TCP, pour Transmission Control Protocol) décentralisés, qui rendent le réseau Internet indéfiniment extensible.

L'Internet est devenu un réseau multi-service quasi universel sur lequel cohabitent de très nombreux utilisateurs aux usages différents, et de nombreuses applications ayant des besoins différents. En particulier, les premières études du trafic montrent que le trafic ne possède pas les propriétés de régularité que l'on pensait (modèles de Poisson, de Gilbert, de Markov, etc.), mais bien au contraire, le trafic est particulièrement instable, du à des propriétés de dépendance à long terme particulièrement néfastes à la QoS (Qualité de Service) des services de communication Internet.

Le trafic du réseau Internet est en général un processus complexe et très variable avec le temps. Par conséquent, avant d'effectuer la prévision du trafic, il est nécessaire d'étudier les caractéristiques du trafic et ses propriétés ainsi que les paramètres qui peuvent l'affecter. Certes, les méthodes statistiques standards constituent un outil important permettant de traiter, étudier et modéliser les données représentant le trafic Internet. Ces méthodes englobent principalement des techniques de prétraitement, d'analyse et de modélisation de données [3].

1.1 Caractéristiques de trafic internet :

L'importante autocorrélation et l'extrême variabilité du trafic Internet sont la conséquence de sa non-stationnarité ainsi que son caractère *d'auto-similarité* et de *dépendance à long terme*. Dans ce qui suit, nous présentons ces caractéristiques ainsi que leurs formulations mathématiques.

1.1.1 Non-stationnarité :

- Un processus $(X_i)_{i \geq 1}$ est stationnaire si pour tout $p \geq 1$, $(X_1, \dots, X_p)$ a même loi que $(X_2, \dots, X_{p+1})$.
- Un processus $(X_i)_{i \in \mathbb{Z}}$ est stationnaire si pour tout $p \geq 1$, $(X_{-p}, \dots, X_p)$ a même loi que $(X_{-p+1}, \dots, X_{p+1})$.

Définition 1 : Un processus $\{Y(t); t \geq 0\}$ est dit stationnaire au sens large si son espérance et ses autocovariances sont invariantes par translation dans le temps :

$$-\forall t \in \mathbb{N}, \mathbb{E}[Y(t)] = \mu \text{ où } \mu \text{ est constantes}$$

$$-\forall t, h \in \mathbb{N}, \text{Cov}(Y(t), Y(t+h)) = \text{Cov}(Y(0), Y(h)).$$

Ainsi, l'autocovariance entre $Y(t)$ et $Y(t+h)$ ne dépend que de l'intervalle h et non de la date t . Par conséquent, la variance est constante pour un processus stationnaire :

$$\text{var}[Y(t)] = \text{Cov}(Y(0), Y(0)) \quad \forall t \in \mathbb{N}$$

Définition 2 : Un processus $\{Y(t); t \geq 0\}$ est dit stationnaire au sens strict si $\forall h, t_1, \dots, t_n$.

$$f_{Y(t_1), \dots, Y(t_n)} = f_{Y(t_1+h), \dots, Y(t_n+h)}$$

Où $f_{Y(t_1), \dots, Y(t_n)}$ est la fonction de distribution conjointe de $Y(t_1), \dots, Y(t_n)$

Ainsi, un processus est stationnaire lorsque sa structure probabiliste (sa distribution) est stable au cours du temps (c.-à-d. indépendante du temps). Une série temporelle $y(t)$ est stationnaire si elle est la réalisation d'un processus stationnaire.

Les études effectuées sur le trafic réel collecté sur Internet montrent que le trafic n'est pas stationnaire. En effet, le trafic se caractérise par une périodicité mensuelle, hebdomadaire et journalière. Cependant, des travaux ont montré qu'il est possible de considérer que le trafic est stationnaire au sens large seulement à très faible échelle de temps.

La non-stationnarité rend la prévision une tâche difficile puisque les caractéristiques statistiques du trafic sont très variables au cours du temps. Cependant, il existe des techniques permettant de stationnariser des données. La méthode la plus utilisée pour stationnariser les données est appelée *la différenciation*.

La différenciation d'ordre d de la série $y(t)$ est la série $(1 - L)^d y(t)$ où L est l'opérateur

« retard ».

Nous pouvons ainsi effectuer plusieurs différenciations d'une façon répétée (d fois) jusqu'à ce que la série devienne stationnaire [3] .

1.1.2 Autosimilarité et dépendance à long terme :

Les processus à autosimilaire et de la dépendance à long terme ont été étudiés au milieu du dernier siècle. Ils ont été découverts expérimentalement puis introduits mathématiquement dans de nombreux domaines de la science, comme l'économie et les statistiques et la finance. Au cours de ces dernières années, ces processus ont été utilisés pour modéliser le trafic dans les réseaux de communication moderne que ce soit dans les réseaux locaux Ethernet, les réseaux étendus ou encore dans les réseaux métropolitains. Ce choix a été motivé à la suite des observations statistiques réalisées sur le trafic réel.

Dans ce qui suit, nous présentons *les formulations mathématiques* des processus autosimilarité et de la dépendance à long terme [3].

1.1.2.1 Autosimilarité :

Définition 1 : Un processus continu $Y = \{Y(t); t \geq 0\}$ est dit autosimilaire d'ordre H s'il satisfait à la condition suivante :

$$Y(t) \stackrel{d}{=} a^{-H} Y(at), \forall t \geq 0, \forall a > 0, \frac{1}{2} \leq H < 1$$

Où H est appelé le paramètre de Hurst et le symbole $\stackrel{d}{=}$ signifie une égalité en distribution.

Il existe aussi une seconde définition de l'autosimilarité en temps discret qui est plus appropriée à la théorie standard des séries temporelles [3] .

Définition 2 : Soit $X = \{X(t); t \in \mathbb{N}\}$ un processus discret et stationnaire, le processus agrégé d'ordre m , noté $X^m = \{X^m(t); t \in \mathbb{N}\}$, est défini comme étant le processus obtenu par le calcul de la moyenne des éléments du processus original X contenus dans des blocs contigus de taille m :

$$X^m(k) = \frac{1}{m} \sum_{t=(k-1)m+1}^{km} X(t)$$

Définition 3 : Un processus discret X est dit autosimilaire d'ordre H si quel que soit. L'entier $m > 0$, X^m a la même distribution que X multipliée par m^{H-1} , c. -à-d. :

$$X(t) \stackrel{d}{=} m^{1-H} X^m(t)$$

Où H est le paramètre de Hurst $\frac{1}{2} < H < 1$ et le symbole $\stackrel{d}{=}$ signifie une égalité en distribution.

Théorème : Soit $Y = \{Y(t); t \geq 0\}$ un processus continu auto-similaire d'ordre H . Si le processus discret $X = \{X(t) = Y(t) - Y(t - 1); t \in \mathbb{N}\}$ est stationnaire alors il est autosimilaire d'ordre H .

Le processus continu Y est généralement interprété comme étant le volume de données cumulatif qui traversent le lien (exprimé en Mb(**mégabyte**) ou en paquets). Ainsi, le processus discret X peut être interprété comme étant le débit à l'intervalle $[t, t + 1[$, c'est à-dire la quantité de données qui a traversé le lien pendant cet intervalle (exprimée en Mbps ou en paquets par seconde).

D'autre part, la fonction d'autocorrélation du processus X , notée $r(k)$, vérifie pour $\frac{1}{2} < H < 1$:

$$r(k) = d^{k^{2H-2}}, k \rightarrow \infty$$

Où H est le même paramètre de *Hurst*.

L'une des caractéristiques du trafic autosimilaire est que le degré d'autosimilarité peut être mesuré en utilisant seulement le paramètre de Hurst. Plusieurs algorithmes ont été proposés afin d'estimer le paramètre de Hurst à partir d'un ensemble de données.

1.1.2.2 Dépendance à long terme :

La dépendance à long terme est l'une des propriétés les plus importantes des processus autosimilaires. Elle est définie par le comportement de l'autocorrélation du processus.

Définition 1 : Un processus présente une dépendance à long terme si sa fonction d'autocorrélation, notée $r(k)$, décroît hyperboliquement avec k et ne s'annule pas. Ceci est traduit mathématiquement par :

$$\sum_{K=-\infty}^{\infty} r(k) = \infty$$

Si $\sum_{K=-\infty}^{\infty} r(k) = \infty$ la somme est finie alors le processus présente une dépendance à court terme.

Ainsi, en examinant l'équation le quel, la valeur du paramètre H permet d'identifier si un processus présente une dépendance à long terme:

- Si $0 < H \leq \frac{1}{2}$, on a $\sum_{k=-\infty}^{\infty} r(k) = 0$ le processus n'est pas autosimilaire et il ne présente pas une dépendance à long terme, mais, plutôt une dépendance à court terme.
- Si $\frac{1}{2} < H < 1$, on a $\sum_{k=-\infty}^{\infty} r(k) = \infty$ le processus est autosimilaire avec une dépendance à long terme. Plus le paramètre H s'approche 0 de 1, plus le degré d'autosimilarité et de dépendance à long terme augmente.

Le trafic autosimilaire est responsable de beaucoup d'effets indésirables dans le réseau, notamment l'importante variance du débit au sein du réseau. Ceci engendre des tailles de files

d'attente très variables. Par conséquent, les temps d'attente au niveau des routeurs deviennent très variables.

Par contre, l'importante autocorrélation, qui est actuellement présente dans le trafic, peut être judicieusement exploitée pour prévoir le trafic avec précision. Plusieurs travaux recommandent d'utiliser des modélisations du trafic qui tiennent compte d'autosimilarité et de la dépendance à long terme [3].

1.1.3 Distribution à décroissance lente :

Il s'est avéré que plusieurs variables caractérisant le trafic Internet ont des distributions à décroissance lente. Parmi ces variables, on cite la taille des fichiers transmis et la durée des connexions. Intuitivement, une distribution à décroissance lente attribue des probabilités relativement élevées à des régions éloignées de la moyenne ou de la médiane. Ainsi, clic présente une variance très importante. Ceci montre que la distribution à décroissance lente est à l'origine de l'autosimilarité et de la dépendance à long terme du trafic dans le réseau .

Définition 1 : Une variable aléatoire Z est à décroissance lente si :

$$f(x) = P[Z > x] \sim cx^{-\alpha}, x \rightarrow \infty$$

Où α est appelé l'index de la distribution tel que $0 < \alpha < 1$ et c est une constante positive. Lorsque $0 < \alpha < 1$, la distribution à décroissance lente est caractérisée par une variance infinie et une moyenne finie. Lorsque $0 < \alpha \leq 1$ la moyenne devient aussi infinie .

La distribution de Pareto est une distribution à décroissance lente qui a été souvent utilisée pour représenter avec une bonne précision la distribution de la taille des fichiers et de la durée des connexions et des sessions http (*Hypertext Transfer Protocol*) .

1.1.4 La sporadicité du trafic :

La sporadicité (burstiness) du trafic signifie l'existence de périodes concentrées d'activités importantes et de périodes de faibles activités. Ainsi, le trafic se comporte en rafales et crée beaucoup de problèmes au niveau de la gestion du trafic (réservation des ressources) ainsi qu'au niveau des mémoires tampons dans les routeurs.

Dans des travaux précédents, le paramètre de *Hurst* a été considéré comme une métrique pour mesurer la sporadicité du trafic. Cette considération a été justifiée par l'observation visuelle du trafic et par l'hypothèse que le trafic se comporte comme un bruit Gaussien fractionnaire .En plus du paramètre de *Hurst*, nous allons utiliser un autre critère pour mesurer la sporadicité du trafic échantillonné.

Ce critère reflète directement les pics existants dans le trafic. Les critères traditionnellement utilisés sont le coefficient pic sur moyenne, le coefficient de variation ainsi que l'indice de dispersion.). Dans ce travail, nous mesurons la sporadicité avec le ratio pic par moyenne (Peak to Mean - PM) qui est obtenu divisant le pic de la bande passante par la moyenne de la bande passante. Le PM est donné par la formule suivante :

$$PM = \frac{m}{n} \sum_{k=1}^{n/m} \frac{\max_{i=(k-1)m+1}^{km} (y(i))}{\sum_{i=(k-1)m+1}^{km} \frac{y(j)}{m}}$$

OÙ $y(i)$ est une série temporelle, m le nombre de mesures considérées par intervalle d'étude et n est le nombre d'observations disponibles de la série $y(i)$.

Nous présentons, dans la section suivante, les principaux modèles de prévision utilisés dans les travaux précédents ainsi que la présentation des réseaux optiques à commutation de rafales [3].

1.2 Analyse de trafic internet :

L'analyse du trafic Internet consiste à analyser les différentes variables qui peuvent être mesurées à partir du réseau. Ces variables, notamment le débit au sein d'un lien, sont modélisées par des séries temporelles. Ainsi, il est primordial de présenter les différentes approches pour l'analyse des séries temporelles.

Une première approche est constituée par l'analyse spectrale directement importée du domaine de la physique. Elle se base principalement sur la représentation des données dans le domaine fréquentiel afin de rechercher les périodicités.

Une seconde approche est l'analyse des dépendances temporelles qui consiste à étudier directement les corrélations entre la valeur actuelle de la série et ses valeurs passées.

1.2.1 Analyse spectrale (recherche des périodicités) :

Généralement, on connaît la périodicité d'une série représentant le trafic Internet. Celle-ci s'appuie le plus souvent sur des cycles connus : mois, jour et trimestre

Cependant, dans certains cas, les cycles ne sont pas aisément identifiables. De plus, ils peuvent ne pas correspondre à une périodicité connue. L'analyse spectrale permet d'identifier ces périodicités. Ce type d'analyse s'appuie généralement sur la transformée de Fourier qui fournit une bonne description de la série. Cependant, la transformée de Fourier a de nombreuses limitations quand la série temporelle à analyser n'est pas stationnaire (Lin, 1976).

1.2.2 Analyse des dépendances temporelles :

Une série temporelle ne dépend pas seulement des facteurs externes, mais aussi de ses états passés. Dans ce qui suit, nous présentons les fonctions d'autocorrélation qui permettent d'analyser cette dépendance.

- *Fonction d'autocorrélation (ACF) :*

Définition 1 : Soit $r(k)$ l'autocorrélation d'ordre k de la série temporelle $y(t)$. Elle est obtenue en divisant la covariance des séries $y(t)$ et $y(t - k)$ par la variance de la série $y(t)$, c.-à-d. :

$$r(k) = \frac{\sum_{t=1}^{n-k} (y(t) - \bar{y})(y(t+k) - \bar{y})}{\sum_{t=1}^n (y(t) - \bar{y})^2}$$

Où $y(t)$ est la valeur de la série temporelle à l'instant t , n est la taille des données disponibles (c.-à-d. le nombre de valeurs de $y(t)$ disponibles) et $\bar{y}$ est la moyenne estimée sur l'ensemble des valeurs $y(t)$ de la série.

L'autocorrélation d'ordre k d'une série temporelle mesure la corrélation entre cette série et elle-même avec un retard de k mesures. Le graphe de la fonction d'autocorrélation (c.-à-d. $r(k)$ en fonction de k) est appelé *corrélogramme (ACF)*.

- **Fonction d'autocorrélation partielle (PACF) :**

Définition 1 : Soit $r^*(k)$ la corrélation partielle d'ordre k d'une série temporelle $y(t)$. Elle est calculée selon les formules suivantes :

$$r^*(1) = r(1)$$

$$r^*(2) = \frac{r(2) - r^2(1)}{1 - r^2(1)}$$

$$r^*(k) = \frac{r(k) - \sum_{j=1}^{k-1} \varphi_{k-1,j} r(k-j)}{1 - \sum_{j=1}^{k-1} \varphi_{k-1,j} r(j)}$$

Avec les équations de récurrences suivantes :
$$\begin{cases} \varphi_{k,j} = \varphi_{k-1,j} - \varphi_{k,k} \varphi_{k-1,k-j} \\ \varphi_{k,k} = r^*(k) \end{cases}$$

Où $r(k)$ est l'autocorrélation d'ordre k .

La signification des corrélations partielles est simple en dépit d'une forme analytique compliquée : elle mesure la corrélation entre la série $y(t)$ et son retard $y(t-k)$ d'ordre k lorsque l'influence des variables $y(t-k-i)$, $k < i$ a été retirée. Le graphe de la fonction d'autocorrélation partielle (c.-à-d. $r^*(k)$ en fonction de k) est appelé *corrélogramme partiel (PACF)*.

Les études de la décennie passée sur le trafic de réseau ont montré d'une façon convaincante que le trafic des réseaux locaux aussi bien que celui des réseaux étendus sont

caractérisés par une forte autocorrélation et une extrême variabilité . Dans ce qui suit, nous présentons les principales caractéristiques du trafic Internet.

1.3 Conclusion :

Dans ce chapitre, nous avons présenté une définition sur le trafic internet et le trafic de réseau. Ensuite les caractéristiques de trafic internet, Non-stationnarité et d'*autosimilarité* et de dépendance à long terme, Distribution à décroissance lente ainsi que leurs formulations mathématiques, finalement nous avons analysé le trafic internet et sa subdivision.

Chapitre 2

Modèles de prévision : Algorithmes de classification

2.1 Introduction

Aujourd'hui, des milliards de données sont collectées chaque jour dans le monde. En effet, les faibles coûts des machines en termes de stockage et de puissance ont encouragé les sociétés à accumuler toujours plus d'informations. Cependant, bien que la quantité de données à traiter ne cesse d'augmenter les spécialistes dans le domaine estiment que la quantité de données collectées dans le monde double tous les 20 mois. Les entreprises étaient jusqu'alors incapables de transformer leurs données en connaissance directement utilisable.

Dans cette optique, un ensemble d'architectures, de démarches et d'outils a été regroupé en une forme homogène sous le terme de fouille de données ou Data Mining.

2.2 Définition de Data Mining :

La fouille de données ou le Data Mining consiste essentiellement à extraire de l'information d'immenses bases de données de la façon la plus automatique possible.

Plus concrètement, le Data Mining est un processus de traitement informatique d'une très grande quantité de données afin de trouver des informations pertinentes, contrairement à la méthode statistique qui nécessite que l'on établisse une hypothèse de départ qu'il s'agira de vérifier. C'est des données elles-mêmes, que se dégageront les corrélations intéressantes. Le Data Mining se situe à la croisée des statistiques, de l'intelligence artificielle et des bases de données [11].

2.3 Les tâches de Data Mining :

Le choix des techniques du Data Mining à appliquer dépend de la tâche particulière à accomplir et des données disponibles pour l'analyse. La première étape consiste à traduire un objectif commercial en une ou plusieurs tâches.

Nous citons ci-dessous les tâches de base :

- La classification consiste à examiner des caractéristiques d'un objet afin de l'affecter à une classe d'un ensemble prédéfini.

- L'estimation est souvent utilisée pour effectuer une tâche de classification.
- La prédiction ressemble à la classification et à l'estimation, mais les enregistrements sont classés selon un certain comportement futur prédit ou à une valeur future estimée.
- Le regroupement par similitudes consiste à déterminer les objets qui vont naturellement ensemble.
- La description de qualité suffisante suggérera souvent une explication.

Une fois les tâches identifiées, elles sont utilisées pour restreindre la gamme des méthodes prises en compte. En termes généraux, notre but est de sélectionner la technique de Data Mining qui minimise le nombre et la difficulté des transformations de données qui doivent être effectuées pour produire de bons résultats. Les données brutes peuvent demander différentes manières d'être résumées, les valeurs manquantes doivent être traitées, etc... Ces transformations sont nécessairement indépendantes de la technique choisie [11].

2.4 Les méthodes de Data Mining :

Les outils de Data Mining utilisent les mêmes aspects théoriques que les techniques statistiques traditionnelles. En utilisant la puissance de ces méthodes et en introduisant le concept d'intelligence artificielle et celui de l'apprentissage automatique, le Data Mining constitue un outil très puissant dans le domaine de l'extraction des données au sein de l'entreprise. Cette combinaison technique facilite la résolution, la compréhension, la modélisation et l'anticipation des problèmes. Le schéma suivant positionne les différentes techniques du Data Mining [12].

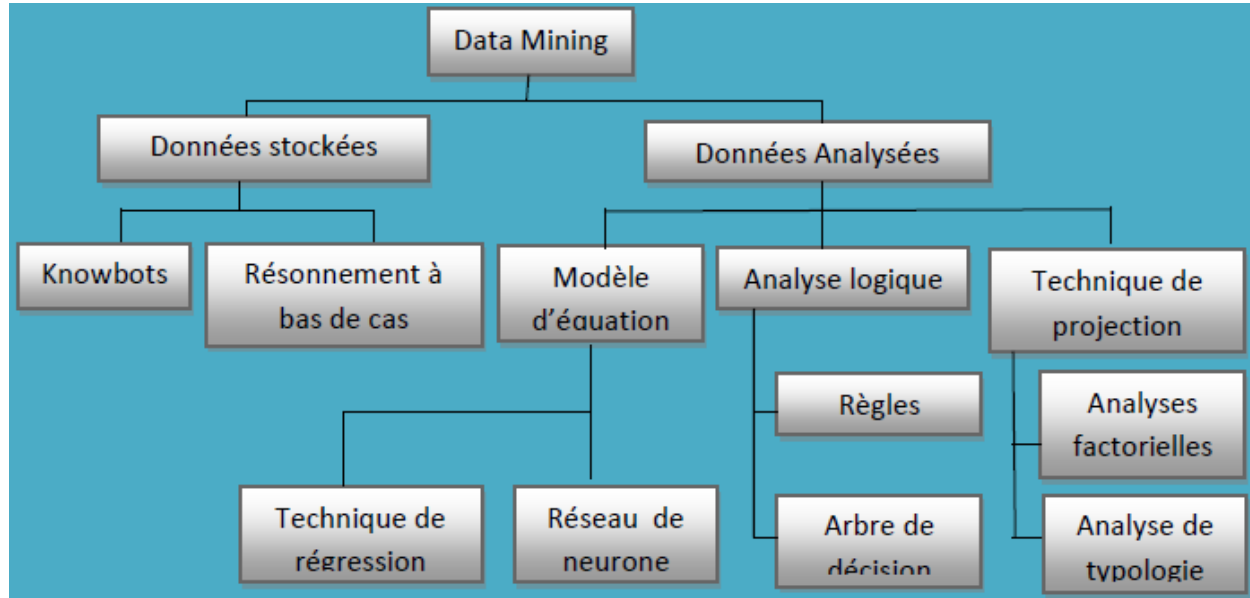


Figure 2-1 Les méthodes de Data Mining

2.5 Domaines d'utilisation de Data Mining :

Les domaines d'application du Data Mining sont vastes et variés. Cependant un trait commun les lie est le fait qu'ils traitent un volume important de données et tire des informations qui visent à améliorer la qualité du produit ou du service.

Parmi les domaines où l'utilisation du Data Mining est devenue monnaie courante :

Laboratoires pharmaceutiques et médicaux : La médecine représente un grand chantier pour les outils du Data Mining, notamment dans la détection des tumeurs et des maladies rares à partir des prélèvements d'autres maladies qui présentent des symptômes similaires dans le but d'identifier les meilleures thérapies.

Assurances : Le Data Mining est un outil très puissant pour l'analyse des sinistres et la recherche des critères explicatifs du risque ou de la fraude mais aussi il fournit des modèles de sélection et de tarification.

Banques et les grandes administrations : Le Data Mining fournit des modèles de prédiction de fraudes ainsi que des modèles de pré-autorisation de crédit automatique à partir des informations stockées dans leurs bases de données.

Automobiles et grandes industries : Afin d'améliorer la qualité du produit, on utilise le Data Mining pour anticiper les défauts de production. Par ailleurs, les méthodes du Data Mining servent pour la prévention des ventes.

Les transports à grandes échelles : On utilise le Data Mining pour l'optimisation des tournées et dans le Marketing afin de définir des programmes et des promotions selon les classes de clients et leurs caractéristiques.

Grande distribution et vente en correspondance : L'étude dans ce cas se base sur l'analyse des similarités des consommateurs en fonction de critères géographiques et sociodémographiques et l'analyse des comportements des individus [12].

2.6 Classification automatique :

L'objectif de la Classification automatique est de classer de façon automatique les documents dans des catégories qui ont été définies soit préalablement par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore clustering.

2.6.1 Apprentissage non supervisé (Clustering) :

L'apprentissage non supervisé consiste à apprendre à classer sans supervision. Au début du processus nous ne disposons ni de la définition des classes, ni de leurs nombres. C'est l'algorithme de classification qui va déterminer ces informations. Nous ne disposons pas non plus de données en entrée qui sont déjà classées, c'est aussi à l'algorithme de découvrir par lui-

même la structure plus ou moins cachée des données et de former des groupes d'individus dont les caractéristiques sont communes [13].

2.6.2 Apprentissage supervisé (Catégorisation) :

Contrairement à l'apprentissage non supervisé, nous commençons ici par un ensemble de classes connues et définies à l'avance. Nous disposons aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. La classification se fait par l'algorithme selon le modèle qu'il a appris.

La catégorisation de textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la classification supervisée pour l'apprentissage automatique et à la discrimination en statistiques alors que la recherche d'informations utilise des termes plus proches de l'application concernée : filtrage ou routage.

Cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve bayes, K-plus proches voisins, arbres de décision, machines à vecteurs support, réseaux de neurones, etc....) [13].

2.6.3 Avantages et inconvénients :

Parmi les avantages et inconvénients liés aux deux approches, on peut citer :

- Les groupes ou clusters obtenus par la technique supervisée est de meilleure qualité et plus précise que la technique non-supervisée.
- Dans la technique supervisée, on sait ce qui est attendu favorisant de meilleurs résultats par rapport au non supervisée.
- Un avantage des techniques non supervisées, est qu'elles accomplissent la tâche de similarité sans avoir besoin des données expertisées.
- Un inconvénient des approches supervisées, repose sur le fait qu'il peut être difficile de se procurer des données expertisées.
- L'inconvénient majeur des approches non supervisées qu'elle demande dans l'étape d'évaluation des résultats l'intervention d'un expert.

2.7 Algorithmes d'apprentissage :

En apprentissage automatique, différents types de classificateurs ont été mis au point, et cela dont le but d'atteindre un degré maximal de précision et d'efficacité, chacun ayant ses avantages et ses inconvénients. Mais, ils partagent toutefois des caractéristiques communes.

Parmi la panoplie de classificateurs existants, on peut faire des regroupements et distinguer des grandes familles.

Dans les pages qui suivent, nous allons exposer quelques algorithmes en détail, le classificateur bayésienne naïf algorithme qui nous avons utilisé dans notre étude, surpassé par d'autres mais il est souvent utilisé comme point de référence à cause de sa simplicité.

Il existe de nombreux algorithmes d'apprentissage supervisée, notamment :

- L'algorithme des K plus proches voisins (ou K-NN).
- Les arbres de décision.
- Machines à support de vecteurs (ou SVM).
- Les réseaux de neurones (RNA).
- L'algorithme de Naïve Bayes.

2.8 k-plus proches voisins (k-nearest neighbor):

L'idée de base de l'algorithme des *k-plus proches voisins* (PPV), traduction de (*nearest neighbor(k-NN)* en anglais). Sur une comparaison directe entre le vecteur caractéristique de l'instance à classer et les vecteurs des instances de la base d'apprentissage. La comparaison consiste en un calcul de distances entre ces instances. Puis à l'instance a classer est assignée la classe majoritaire parmi les classes des k instances les plus proches. Il existe plusieurs distances employées par l'algorithme k-NN pour comparer deux instances, parmi ces métriques nous pouvons citer :

Notons par $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ le vecteur caractéristique de l'instance i , avec n le nombre de variables et par p et q deux instances à comparer [1].

2.9 L'algorithme k-NN :

L'algorithme des k-plus proches voisins est l'un des algorithmes les plus simples de la classification supervisée, étant donné que l'ensemble des classes est défini à l'avance, cette méthode permet d'obtenir de très bons résultats de classification, dont l'algorithme est le suivant [6] :

Algorithme 5 Algorithme de classification par k-PPV

Paramètre : le nombre k de voisins

Contexte : un échantillon de l textes classés en $C = c_1, c_2, \dots, c_n$ classes

1 : **pour** chaque texte t **faire**

2 : transformer le texte t en vecteur $t = (x_1, x_2, \dots, x_m)$

3 : déterminer les k plus proches textes du texte t selon une métrique de distance

4 : combiner les classes de ces k exemples en une classe c

5 : **fin pour**

Sortie : le texte t associé à la classe c .

Le choix du paramètre K est primordial pour le bon fonctionnement de cette méthode. Une grande base d'apprentissage permet une plus grande valeur de K , et un K petit est nécessaire pour des petites bases d'apprentissage.

2.9.1 Principe de fonctionnement :

L'algorithme de K-NN compare avec ceux déjà classés en cherchant ses K plus proches voisins. Une fois ces derniers déterminés, le nouveau document est classé dans la catégorie qui inclut le maximum de voisins parmi les K trouvés [14].

Deux paramètres sont utilisés : le nombre K et la fonction de similarité pour comparer le nouveau document à ceux déjà classés telle que la distance euclidienne par exemple qui est donnée par l'équation suivante :

Distance euclidienne :

$$D(X_p, X_q) = \sqrt{\sum_{k=1}^n (x_{pi} - x_{qi})^2}$$

Distance de Manhattan :

$$D(X_p, X_q) = \sum_{i=1}^n |(x_{pi} - x_{qi})^2|$$

Distance de Minkowski :

$$D(X_p, X_q) = \left(\sum_{i=1}^n (x_{pi} - x_{qi})^r \right)^{\frac{1}{r}}$$

Distance de Tchebychev :

$$D(X_p, X_q) = \max_{i=1}^n |x_{pi} - x_{qi}|$$

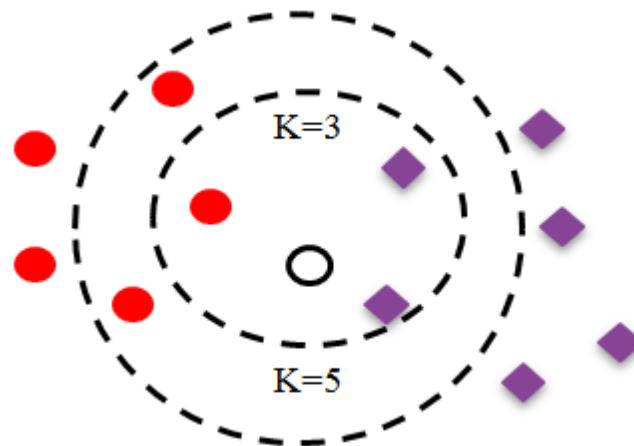


Figure 2-2 K-plus proche voisin

2.9.2 Critiques de la méthode :

L'avantage que présente cette méthode est sa simplicité et son efficacité qui fait d'elle une méthode très utilisée ; toutefois, on peut lui reprocher le fait qu'elle utilise un nombre important d'objets pour calculer la similarité avec un nouvel objet à classer et plus le nombre d'objets est grand plus le temps d'exécution est très important [14].

2.9.3 Les domaines d'application :

La méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Or, il est possible de définir des distances sur des champs complexes tels que des informations géographiques, des textes, des images, et du son. C'est parfois un critère de choix de la méthode K-PPV car les autres méthodes traitent difficilement les données complexes. On peut noter, également, que la méthode est robuste au bruit.

2.10 Arbre de décision:

Définition : Les arbres de décision sont plus populaires des méthodes d'apprentissage. Les Algorithmes connus sont ID3 (Quinlan 1986) et C4.5 (Quinlan 1993). Ils sont également populaires pour la classification de document.

Comme toute méthode d'apprentissage supervisée, les arbres de décision utilisent des exemples. Si l'on doit classer des documents dans des catégories, il faut construire un arbre de décision par catégorie. Pour déterminer à quelle(s) catégorie(s) appartient un nouveau document, on utilise l'arbre de décision de chaque catégorie auquel on soumet le document à classer. Chaque arbre répond Oui ou Non (il prend une décision).

Concrètement, chaque nœud d'un arbre de décision contient un test (un IF...THEN) et les feuilles ont les valeurs Oui ou Non. Chaque test regarde la valeur d'un attribut de chaque exemple. En effet, on suppose qu'un exemple est un ensemble d'attributs/valeurs. Pour des documents,

chaque attribut peut être un mot et la valeur sera par exemple 0 ou 1 selon que ce mot appartient ou non au document [12].

Le formalisme des arbres de décision permet de classer un nouvel objet en testant ses caractéristiques les unes à la suite des autres. Les exemples de l'ensemble d'apprentissage sont récursivement divisés par des tests définis sur les caractéristiques pour obtenir des sous-ensembles d'exemples ne contenant que des exemples appartenant tous à une même classe.

Les arbres de décision sont construits de haut en bas selon une structure d'arbre, où chaque nœud est soit :

- **Une feuille** : représente la classe des instances.
- **Un nœud de décision** : permettant de tester les instances sur la valeur d'un attribut, il possède deux ou plusieurs branches [1].

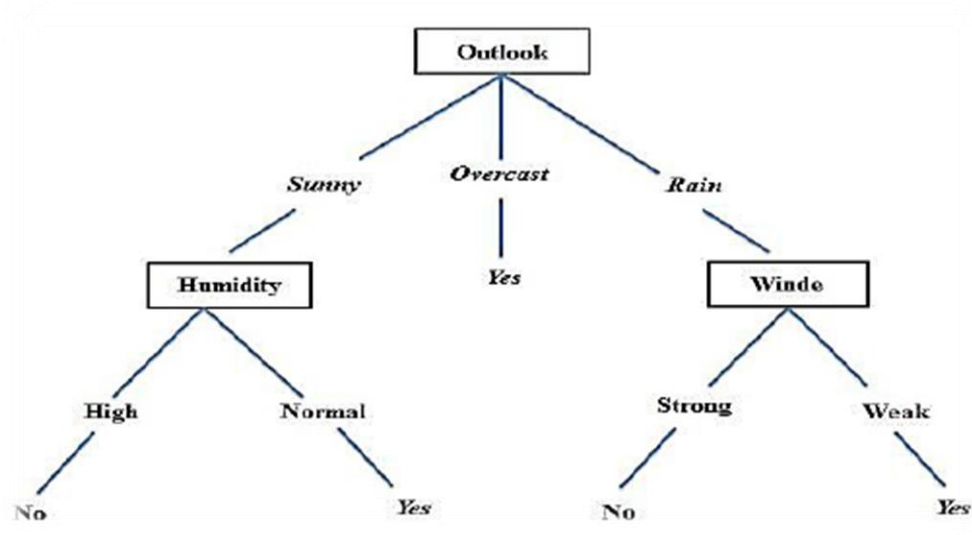


Figure 2-3 l'arbre de décision

Pour construire l'arbre de décision, il faut trouver quel attribut tester à chaque nœud. C'est un processus récursif. Pour déterminer quel attribut tester à chaque étape, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples Oui/Non.

On crée alors un nœud contenant ce test, et on crée autant de descendants que de valeurs possibles pour ce test.

Exemple : si on teste la présence d'un mot, les valeurs possibles sont Présent/Absent. A chaque fois, on aura donc deux descendants pour chaque nœud.

L'algorithme général d'un arbre de décision est le suivant :

Algorithme 1 Algorithme d'arbre de décision

Début

Entrées : base d'apprentissage S

Début

Initialiser à l'arbre vide ; // la racine est le nœud courant

Répéter

Décider si le nœud courant est terminal ;

Si le nœud est terminal alors affecter une classe ;

Sinon sélectionner un test et créer le sous-arbre ;

Passer au nœud suivant non exploré s'il existe ;

Jusqu'à obtenir un arbre de décision ;

Fin

Algorithme d'arbre de décision

Les algorithmes existants (CART, ID3, C4.5 ...) diffèrent essentiellement par leur façon de choisir, à une étape donnée, et parmi les caractéristiques disponibles, la caractéristique de segmentation et par le critère d'arrêt.

2.10.1 Algorithme ID3 :

Proposé par Quinlan l'algorithme Itérative Dichotomiser 3 (ID3) est construit en sélectionnant à chaque nœud l'attribut qui maximisera le gain d'information en d'autre terme l'attribut permettant la meilleure discrimination entre les classes à ce niveau de l'arbre. La fonction utilisée pour calculer le gain d'information est :

$$Gain(m ; T) = Entropie(p) - \sum_{i=1}^n (\rho_a * Entropie(\rho_i))$$

Avec Entropie, représentant l'entropie de Shannon :

$$Entropie(\rho) = - \sum_{i=1}^n \rho_i * \log(\rho_i)$$

2.10.2 Algorithme C4.5 :

Proposé par Quinlan , cet algorithme apporta plusieurs améliorations à l'arbre de décision ID3:

— Il peut traiter des données de types catégorielles et continues tandis qu'ID3 ne peut manipuler que des données catégorielles.

— C4.5, à la différence de ID3, utilise une phase d'élagage lui permettant d'éviter le sur-apprentissage en éliminant les branches superflues.

D'un autre côté, l'algorithme C4.5, utilise un test, nommé Gain Ratio, qui maximise le gain d'information :

$$GainRatio(p, T) = \frac{Gain(p, T)}{splitInfo(p, T)}$$

Avec

$$splitInfo(p, test) = - \sum_{j=1}^n p' \left(\frac{j}{p} \right) * \log \left(p' \left(\frac{j}{p} \right) \right)$$

Où $p' \left(\frac{j}{p} \right)$ est la proportion des éléments présents à la position p prenant la $j^{ème}$ valeur de test.

La valeur de la fonction *Splitinfo* est grande lorsqu'un attribut comporte peu d'instances pour chaque valeur. Donc, en choisissant le test qui maximise le *GainRatio*, on choisit l'attribut générant le moins de sous-arbres.

2.10.3 Algorithme Classification And Regression Trees (CART) :

L'algorithme CART fut proposé par Breiman. à la différence des arbres de décision cités précédemment, chaque nœud de l'arbre CART ne possède que deux fils et le critère utilisé pour la division est l'index de Gini [1].

2.10.4 Les domaines d'application :

Cette méthode peut être utilisée dans plusieurs domaines tels que :

Les études (pour comprendre les critères prépondérants dans l'achat d'un produit, l'impact des dépenses publicitaires), les ventes (pour analyser les performances par région, par enseigne, par vendeur), l'analyse de risques (pour détecter les facteurs prédictifs d'un comportement de non-paiement), Le domaine médical (pour étudier le rapport existant entre certaines maladies et des particularités physiologiques ou sociologiques) [6].

2.11 Naïve bayes :

La **classification naïve bayésienne** est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésienne naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs Linéaires.

Un terme plus approprié pour le modèle probabiliste sous-jacent pourrait être « modèle à Caractéristiques statistiquement indépendantes ».

En termes simples, un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classifieur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

Selon la nature de chaque modèle probabiliste, les classifieurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé.

Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayésiennes naïfs repose sur le maximum de vraisemblance. Autrement dit, il est possible de travailler avec le modèle bayésien naïf sans se préoccuper de probabilité bayésienne ou utiliser les méthodes bayésiennes.

Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classifieurs bayésienne naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue, Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats.

L'avantage du classifieur bayésienne naïf est qu'il requiert relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables. En effet, l'hypothèse d'indépendance des variables permet de se contenter de la variance de chacune d'entre elle pour chaque classe, sans avoir à calculer de matrice de covariance [13] .

2.11.1 Description du modèle Bayésien :

Le modèle probabiliste pour un classifieur est le modèle conditionnel $p(C|f_1, \dots, f_n)$

Où C est une variable de classe dépendante dont les instances ou *classes* sont peu nombreuses, conditionnée par plusieurs variables caractéristiques $p(f_1, \dots, f_n)$

Lorsque le nombre de caractéristiques n est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible.

Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble A l'aide du théorème de Bayes, nous écrivons :

$$p(C|F_1, \dots, F_n) = \frac{p(C) p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}.$$

En langage courant, cela signifie :

$$\text{postérieure} = \frac{\text{antérieure} \times \text{vraisemblance}}{\text{évidence}}.$$

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de C et les valeurs des caractéristiques F_i sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables $p(C, F_1, \dots, F_N)$

et peut-être factorisé de la façon suivante, en utilisant plusieurs fois la définition de la probabilité conditionnelle :

$$\begin{aligned} p(C, F_1, \dots, F_n) &= p(C) p(F_1, \dots, F_n|C) \\ &= p(C) p(F_1|C) p(F_2, \dots, F_n|C, F_1) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3, \dots, F_n|C, F_1, F_2) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3|C, F_1, F_2) p(F_4, \dots, F_n|C, F_1, F_2, F_3) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3|C, F_1, F_2) \dots p(F_n|C, F_1, F_2, F_3, \dots, F_{n-1}). \end{aligned}$$

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque F_i est indépendant des autres caractéristiques $F_{i \neq j}$, alors :

$$p(F_i|C, F_j) = p(F_i|C)$$

Pour tout $i \neq j$, par conséquent la probabilité conditionnelle peut s'écrire :

$$\begin{aligned} p(C, F_1, \dots, F_n) &= p(C) p(F_1|C) p(F_2|C) p(F_3|C) \dots \\ &= p(C) \prod_{i=1}^n p(F_i|C). \end{aligned}$$

Par conséquent, en tenant compte de l'hypothèse indépendance ci-dessus, la probabilité conditionnelle de la variable de classe C peut être exprimée par :

$$p(C|F_1, \dots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^n p(F_i|C)$$

Où Z (appelé "évidence") est un facteur d'échelle qui dépend uniquement de $F_1, \dots, F_n$, à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues.

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure $p(C)$ (probabilité *a priori* de C) et les lois de probabilité indépendantes $P(F_i|C)$. S'il existe k classes pour C et si le modèle pour chaque fonction $P(F_i|C) = C$ peut être exprimé selon r paramètres, alors le modèle bayésien naïf correspondant dépend de $(k - 1) + n r k$ paramètres.

Dans la pratique, on observe souvent des modèles où $k = 2$ (classification binaire) et $r = 1$ (les caractéristiques sont alors des variables de Bernoulli). Dans ce cas, le nombre total de paramètres du modèle bayésien naïf ainsi décrit est de $2n + 1$, avec n le nombre de caractéristiques binaires utilisées pour la classification [16].

2.11.2 Estimation de la valeur des paramètres :

Tous les paramètres du modèle (probabilités *a priori* des classes et lois de probabilités associées aux différentes caractéristiques) peuvent faire l'objet d'une approximation par rapport aux fréquences relatives des classes et caractéristiques dans l'ensemble des données d'entraînement. Il s'agit d'une estimation du maximum de vraisemblance des probabilités. Les probabilités *a priori* des classes peuvent par exemple être calculées en se basant sur l'hypothèse que les classes sont équiprobables (i.e. chaque antérieure = $1 / (\text{nombre de classes})$), ou bien en estimant chaque probabilité de classe sur la base de l'ensemble des données d'entraînement (i.e. antérieure de $C = (\text{nombre d'échantillons de } C) / (\text{nombre d'échantillons total})$).

Pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question ; sinon, il faut générer des modèles non-paramétriques pour les caractéristiques appartenant à l'ensemble de données d'entraînement. Lorsque l'on travaille avec des caractéristiques qui sont des variables aléatoires continues, on suppose généralement que les lois de probabilités correspondantes sont des lois normales, dont on estimera l'espérance et la variance.

L'espérance, μ , se calcule avec :

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

Où N est le nombre d'échantillons et x_i est la valeur d'un échantillon donné.

La variance, σ^2 , se calcule avec :

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2$$

Si, pour une certaine classe, une certaine caractéristique ne prend jamais une valeur donnée dans l'ensemble de données d'entraînement, alors l'estimation de probabilité basée sur la fréquence aura pour valeur zéro. Cela pose un problème puisque l'on aboutit à l'apparition d'un facteur nul lorsque les probabilités sont multipliées. Par conséquent, on corrige les estimations de probabilités avec des probabilités fixées à l'avance [16] .

2.11.3 Construire un classifieur à partir du modèle de probabilités :

Jusqu'à présent nous avons établi le modèle à caractéristiques indépendantes, à savoir le modèle de probabilités bayésien naïf. Le classifieur bayésien naïf couple ce modèle avec une règle de décision.

Une règle couramment employée consiste à choisir l'hypothèse la plus probable. Il s'agit de la règle du maximum a posteriori ou MAP. Le classifieur correspondant à cette règle est la fonction **classifieur** suivante :

$$\text{classifieur}(f_1, \dots, f_n) = \operatorname{argmax}_c p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c).$$

2.11.4 Analyse :

Fait étonnant, malgré les hypothèses d'indépendance relativement simplistes, le classifieur bayésien naïf a plusieurs propriétés qui le rendent très pratique dans les cas réels. En particulier, la dissociation des lois de probabilités conditionnelles de classe entre les différentes caractéristiques aboutit au fait que chaque loi de probabilité peut être estimée indépendamment en tant que loi de probabilité à une dimension. Cela permet d'éviter nombre de problèmes venant du fléau de la dimension, par exemple le besoin de disposer d'ensembles de données d'entraînement dont la quantité augmente exponentiellement avec le nombre de caractéristiques.

Comme tous les classifieurs probabilistes utilisant la règle de décision du maximum a posteriori, il classifie correctement du moment que la classe adéquate est plus probable que toutes les autres.

Par conséquent les probabilités de classe n'ont pas à être estimées de façon très précise.

Le classifieur dans l'ensemble est suffisamment robuste pour ne pas tenir compte de sérieux défauts dans son modèle de base de probabilités naïves.

La documentation citée en fin d'article détaille d'autres raisons pour le succès empirique des classifieurs bayésiens naïfs [16] .

2.12 Machines à support de vecteurs (ou SVM) :

2.12.1 Définition :

SVM (*Support Vector Machine* ou Machine à vecteurs de support) : Les SVMs sont une famille d'*algorithmes d'apprentissage automatique* qui permettent de résoudre des problèmes tant de *classification* que de *régression* ou de *détection d'anomalie*. Ils sont connus pour leurs solides garanties théoriques, leur grande flexibilité ainsi que leur simplicité d'utilisation même sans grande connaissance de *data mining*.

Les SVMs ont été développés dans les années 1990. Comme le montre la figure ci-dessous,

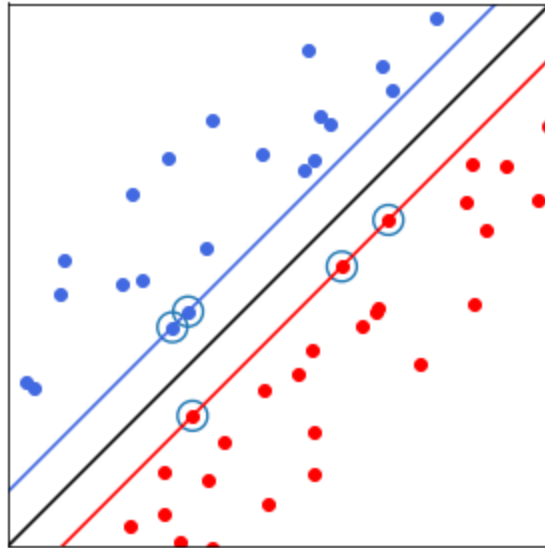


Figure 2-4 vecteurs de support

Dans cet espace à deux dimensions, la « frontière » est la droite noire, les « vecteurs de support » sont les points entourés (les plus proches de la frontière) et la « marge » est la distance entre la frontière et les droites bleue et rouge.

Avant d'aborder le principe de fonctionnement général des SVM voici quelques notions de base :

— **Hyperplan** : est un séparateur d'objets des classes. De cette notion, nous pouvons dire qu'il est évident de trouver une multitude d'hyperplans mais la propriété délicate des SVM est d'avoir l'hyperplan dont la *distance minimale* aux exemples d'apprentissage est maximale, cet hyperplan est appelé *l'hyperplan optimal*, et la distance appelée *marge*.

— **Vecteurs Support** : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier.

Voici un schéma représentatif de ces notions [4] :

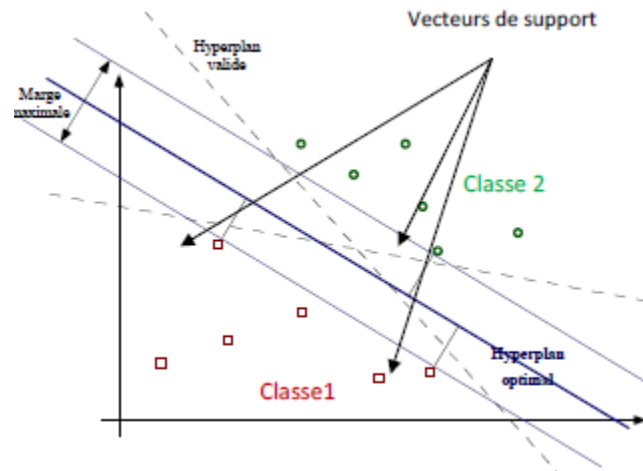


Figure 2-5 Les vecteurs à support

2.12.2 Principe de fonctionnement :

Le principe des SVM consiste en une stratégie de minimisation structurelle du risque mais le problème revient à trouver une frontière de décision qui sépare l'espace en deux régions, à trouver l'hyperplan qui classe correctement les données et qui se trouve le plus loin possible de tous les exemples. On dit qu'on veut maximiser la marge qui veut dire la distance du point le plus proche de l'hyperplan.

2.12.3 Domaine d'application :

Comme nous l'avons vu, les SVM dépendent d'algorithmes d'apprentissage supervisé. L'objectif de l'utilisation de SVM est de classer correctement les données non visibles. Les SVM ont de nombreuses applications dans plusieurs domaines. Certaines applications courantes de SVM sont :

- **Détection des visages :**
SVM classe les parties de l'image en visages et en non-visages et crée une bordure carrée autour du visage.
- **Catégorisation du texte et de l'hypertexte :**
Les SVM permettent la catégorisation du texte et de l'hypertexte pour les modèles inductifs et transductifs. Ils utilisent les données de formation pour classer les documents en différentes catégories. Il catégorise sur la base du score généré, puis compare avec la valeur de seuil.
- **Classification des images :**
L'utilisation de SVM offre une meilleure précision de recherche pour la classification des images. Il offre une meilleure précision par rapport aux techniques de recherche traditionnelles basées sur des requêtes.
- **Bio-informatique :**

Comprend la classification des protéines et la classification du cancer. Nous utilisons SVM pour identifier la classification des gènes, des patients sur la base de gènes et d'autres problèmes biologiques.

- **Détection de repli protéique et d'homologie à distance :**
Appliquez des algorithmes SVM pour la détection d'homologie à distance de protéines.
- **Reconnaissance de l'écriture manuscrite :**
Nous utilisons des SVM pour reconnaître les caractères manuscrits largement utilisés.

2.13 Réseau de neurones (Réseau neuronal) :

Un réseau de neurones est un ensemble de méthodes d'analyse et de traitements des données permettant de construire un modèle de comportement à partir de données qui sont des exemples de ce comportement. Un réseau de neurones est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones.

Ces neurones possèdent une fonction d'activation qui permet d'influencer les autres neurones du réseau. Les connexions entre les neurones, que l'on nomme liens synaptiques, propagent l'activité des neurones avec une pondération caractéristique de la connexion. On appelle poids synaptique la pondération des liens synaptiques.

Les neurones peuvent être organisés de différentes manières, c'est ce qui définit l'architecture et le modèle du réseau. L'architecture la plus courante est celle dite du perceptron multicouche [15].

2.13.1 Architecture du neurone formel :

McCulloch et Pitts (1943) ont modélisé le fonctionnement d'un neurone biologique et imaginé une reproduction artificielle mimétique du raisonnement humain.

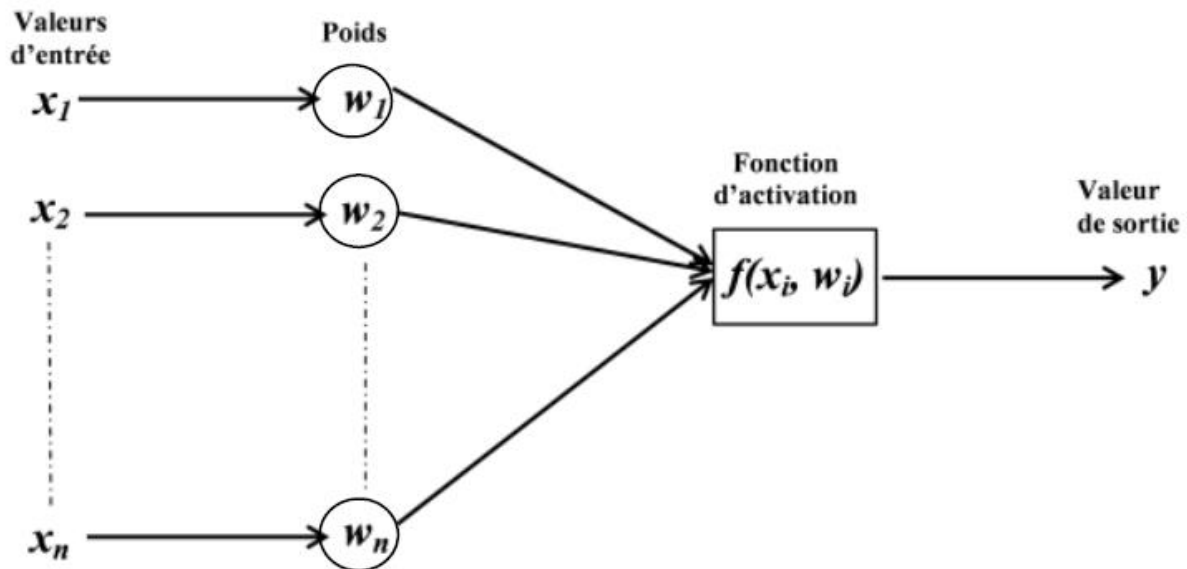


Figure 2-6 Architecture du réseau de neurone formel

2.13.2 Principe de fonctionnement d'un neurone formel :

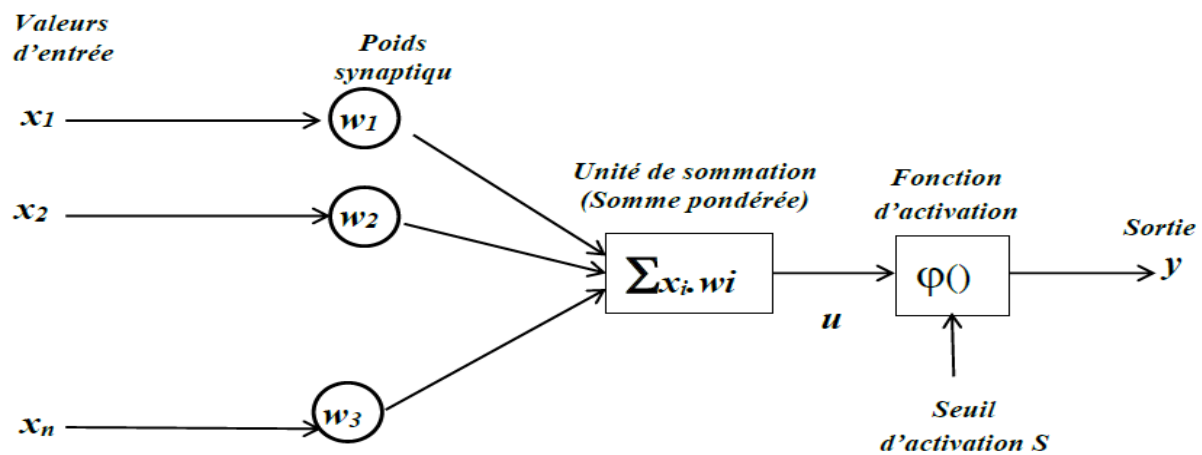


Figure 2-7 modèle de McCulloch et Pitts/Modèle linéaire

Ce modèle base sur deux équations :

$$u = \sum x_i \cdot w_i \quad (1)$$

$$y = \varphi(u - S) \quad (2)$$

Avec :

$x_1, x_2, \dots, x_n$: sont les entrées,

$w_1, w_2, \dots, w_n$: sont les poids synaptiques du neurone k,

u : la sortie de l'unité de sommation,

S : le seuil,

$\phi()$: la fonction d'activation,

y : le signal de sortie du neurone k.

Valeur de sortie : le neurone formel calcule la somme des entrées $x_1, x_2, \dots, x_n$ pondérées par les poids synaptiques $w_1, w_2, \dots, w_n$ et la comparer avec le seuil S , si le résultat est supérieur au seuil, alors la valeur renvoyée en sortie est +1 (activation du neurone), sinon la valeur renvoyée est 0.

$$y = \begin{cases} +1 & \text{Si } \sum x_i \cdot w_i \\ 0 & \text{Sinon} \end{cases}$$

2.13.3 Fonctionnement :

La construction de la structure du réseau (généralement empirique).

La constitution d'une base de données de vecteurs représentant au mieux le domaine à modéliser. Celle-ci est scindée en deux parties : une partie servant à l'apprentissage du réseau (on parle de base d'apprentissage) et une autre partie aux tests de cet apprentissage (on parle de base de test).

Le paramétrage du réseau par apprentissage. Au cours de l'apprentissage, les vecteurs de données de la base d'apprentissage sont présentés séquentiellement et plusieurs fois au réseau.

Un algorithme d'apprentissage ajuste le poids du réseau afin que les vecteurs soient correctement appris. L'apprentissage se termine lorsque l'algorithme atteint un état stable.

La phase de reconnaissance qui consiste à présenter au réseau chacun des vecteurs de la base de test. La sortie correspondante est calculée en propageant les vecteurs à travers le réseau. La réponse du réseau est lue directement sur les unités de sortie et comparée à la réponse attendue.

Une fois que le réseau présente des performances acceptables, il peut être utilisé pour répondre au besoin qui a été à l'origine de sa construction [15].

2.13.4 Domaine d'application :

Aujourd'hui, les réseaux de neurones ont de nombreuses applications dans des domaines très variés :

- **Traitement d'image** : compression d'images, reconnaissance de caractères et de signatures, reconnaissance de formes et de motifs, chiffrement, classification, ...

- **Traitement du signal** : traitement de la parole, identification de sources, filtrage, classification,
- **Traitement automatique des langues** : segmentation en mots, représentation sémantique des mots (plongements lexicaux), étiquetage morpho-syntaxique, traduction automatique,
- **Contrôle** : diagnostic de pannes, commande de processus, contrôle qualité, robotique, ...
- **Optimisation** : allocation de ressources, planification, régulation de trafic, gestion, finance
- **Simulation** : simulation boîte noire, prévisions météorologiques
- Classification d'espèces animales étant donnée une analyse ADN
- Modélisation de l'apprentissage et perfectionnement des méthodes de l'enseignement
- Approximation d'une fonction inconnue ou modélisation d'une fonction connue mais complexe à calculer avec précision.

2.14 Les points forts et les points faibles :

Dans le Tableau 2.1, et en s'appuyant sur l'étude présente dans (Nidhi & Gupta, 2011) nous résumons les avantages et les inconvénients de ces méthodes.

Algo	Points forts	Points faibles
K-NN	<i>K-NN ne fait pas d'apprentissage, il stocke tout simplement tous les exemples d'apprentissage. Il est également bien adapté à la catégorisation multi-classes puisque sa décision de classification est basée sur un voisinage de documents similaires.</i>	<i>Le temps nécessaire pour calculer la similarité est énorme. En pratique, il est impossible de mettre en œuvre l'algorithme pour des dimensions élevées et des corpus d'exemples énormes. En outre, le stockage mémoire augmente avec le nombre de documents d'entraînement.</i>
NB	<i>Rapide et facile à mettre en œuvre et donne de bons résultats. Il est caractérisé par sa robustesse vis-à-vis des données manquantes, sa vitesse de classification et d'apprentissage.</i>	<i>Une fois que l'espace d'apprentissage devient considérablement large, il est impossible d'interpréter le modèle construit. Son hypothèse naïve affecte la qualité des résultats, si les variables sont liés entre eux.</i>
AD	<i>Simple à comprendre et à interpréter. Ils ont besoin de peu de données et sont capables de gérer à la fois des données numériques et catégorielles. Cet algorithme, évolue bien, même lorsqu'il ya un nombre variable d'exemples d'apprentissage.</i>	<i>Un arbre complet peut être sujet au sur-apprentissage, comme certaines branches peuvent être trop spécifiques aux données d'apprentissage. La plupart des méthodes d'arbre de décision comprennent donc un procédé de construction de l'arbre et un autre pour l'élagage pour enlever les branches trop spécifiques.</i>
SVM	<i>les SVM sont moins sensibles au sur-apprentissage que d'autres méthodes</i>	<i>L'efficacité dépend du choix de la fonction noyau : Puisqu'il n'y a pas une fonction</i>

	<i>puisque la complexité du modèle est indépendante de la dimension de l'espace des attributs. Les approches basées sur les SVM peuvent gérer, un espace d'attributs de grande dimension, avec une précision excellente de classification. Le SVM produit de meilleurs résultats à la fois aux niveaux test et apprentissage, il est robuste par rapport au nombre d'attributs et il très rapide pendant la classification.</i>	<i>noyau supérieure aux autres et le temps requis pour l'apprentissage est relativement long parce qu'on est obligé d'expérimenter avec quelques fonctions noyaux candidates pour en trouver la meilleure qui nous content le plus. L'algorithme ne résiste pas aux valeurs manquantes puisqu'il a besoin de toutes ces derrières pour faire son calcul.</i>
RN	<i>Les réseaux de neurones sont des modèles non linéaires, ce qui les rend souples dans la modélisation des relations complexes du monde réel. Les réseaux de neurones sont en mesure d'estimer les probabilités a posteriori, qui fournissent la base pour établir la règle de classification et de l'analyse statistique.</i>	<i>Avec l'augmentation du nombre d'entrée et les nœuds caches, les paramètres nécessaires pour le réseau neuronal augmentent également, ceci provoque le sur-apprentissage.</i>

Tableau 2-1 Les points forts et les points faibles

2.15 Evaluation des méthodes de classification :

Évaluer les performances d'un système de classification est un enjeu de grande importance car ces performances peuvent être utilisées pour l'apprentissage en tant que tel ou pour optimiser les valeurs des hyper paramètres du classifieur.

Pendant longtemps, le critère retenu pour évaluer ces performances a été le taux de bonne classification, c'est-à-dire le nombre d'éléments d'une base de test correctement classés. Le problème d'un tel critère est qu'il n'est pas adapté à des environnements mal définis. Dans de nombreuses situations, toutes les erreurs n'ont pas les mêmes conséquences. Certaines erreurs ont un coût plus important que d'autres, par exemple, pour les diagnostics médicaux. Un mauvais diagnostic ou traitement peut, en effet, avoir différents coûts ou dangers selon le type d'erreur commise [8].

	Décision Positifs	Décision Négatifs	
Étiquette Positifs	Vrai Positifs, TP	Faux Négatifs, FN	pos^a
Étiquette Négatifs	Faux positifs, FP	Vrai Négatifs, TN	Neg^b
	p_{pos}^c	p_{Neg}^d	N

Tableau 2-2 Matrice de confusion

- ^a Nombre d'éléments étiquetés positifs dans la base.
- ^b Nombre d'éléments étiquetés négatifs dans la base.
- ^c Nombre d'éléments classés positifs.
- ^d Nombre d'éléments classés négatifs.

À partir de la matrice de confusion on peut dériver tout un tas de critères de performance. De manière générale, on préfère donner une fraction d'erreurs à un nombre total d'erreurs (5% d'erreurs, ça parle plus que 10 erreurs, quand on ne connaît pas le nombre de points dans le jeu de test) [10] .

– Le taux de vrais positifs ("True positive rate") :

$$tpr = \frac{TP}{POS} = \frac{TP}{TP+FN}$$

– Le taux de vrais négatifs ("True negative rate") :

$$tnr = \frac{TN}{Neg} = \frac{TN}{TN + FN}$$

– Le taux de faux positifs ("False positive rate") :

$$fpr = \frac{FP}{Neg} = \frac{FP}{FP + TN}$$

–Le taux de faux négatifs ("False negative rate") :

$$fnr = \frac{FN}{Pos} = \frac{FN}{FN + TN}$$

– Le taux de bonne classification ou l'exactitude (accuracy) :

$$\text{acc} = \text{tbc} = \text{Pos} * \text{tpr} + \text{Neg} * (1 - \text{fpr})$$

– En anglais on distingue « precision » et « accuracy », c'est-à-dire la proportion de prédictions correctes parmi les points que l'on a prédits positifs. C'est la capacité de notre modèle à ne déclencher d'alarme que pour un vrai incendie [10].

$$\text{prec} = \frac{TP}{PPos} = \frac{TP}{FP + TP}$$

– Le **rappel** ("recall" en anglais), ou **sensibilité** ("sensitivity" en anglais), est le **taux de vrais positifs**, c'est à dire la proportion de positifs que l'on a correctement identifiés. C'est la capacité de notre modèle à détecter tous les incendies [10] .

$$\text{rec} = \text{tpr} = \frac{TP}{Pos} = \frac{TP}{TP + FN}$$

Pour évaluer un compromis entre rappel et précision, on peut calculer la "*F-mesure*", qui est leur moyenne harmonique [10] .

$$F - \text{mesure} = 2 \times \frac{\text{prec} * \text{rec}}{\text{prec} + \text{rec}} = \frac{2TP}{2TP + FP + FN}$$

Pour finir cette longue liste, on s'intéresse aussi souvent à la **spécificité** ("specificity" en anglais), qui est le **taux de vrais négatifs**, autrement dit la capacité à détecter toutes les situations où il n'y a pas d'incendie. C'est une mesure complémentaire de la sensibilité [10] .

$$\text{specificity} = \frac{TN}{FP + TN}$$

Le taux d'erreur de l'algorithme d'apprentissage automatique est [10] :

$$\text{error} = \frac{(FP + FN)}{(TP + FP) + (TN + FN)}$$

2.16 Conclusion :

Dans ce chapitre, Nous avons présenté une généralité sur les conceptions des méthodes de classification et l'apprentissage automatique, les différents types d'apprentissages quelques algorithmes de classification les plus répandus comme : **les arbres de décision ,naïve bayes et l'algorithme des k-plus proche voisins, SVM, RN** , ainsi que plusieurs méthodes d'évaluation et différentes mesures de performance, comme le **rappel**, la **précision**, la **spécificité** et le **F-mesure**, qui reflètent différents aspects du modèles.

Chapitre 3

Etat de l'art

3.1 Introduction :

Ce chapitre a pour objectif la présentation du contexte de notre projet. Dans un premier temps, les étapes de développement de trafic internet et résumé des articles universitaires utilisés dans le domaine du trafic internet.

3.2 Revue de la littérature :

Internet évolue sans cesse, avec un nombre d'appareils connectés toujours croissant, dans un trafic mondial en pleine expansion. L'analyse de ces transferts de données dans leur globalité représente un réel défi pour les chercheurs qui pourraient en apprendre beaucoup. Avec ces informations, ils pourraient aider à créer un réseau plus efficace, prévenir les pannes et améliorer les défenses contre les cyberattaques.

L'analyse et la prévision du trafic sur le réseau ont des applications dans un large éventail de domaines et ont récemment attiré un nombre important d'études. Différents types d'expériences sont conduits et résumés pour identifier divers problèmes dans les applications de réseau informatique existantes.

L'analyse et la prévision du trafic réseau constituent une approche proactive pour assurer une communication réseau sécurisée, fiable et qualitative. Diverses techniques sont proposées et expérimentées pour analyser le trafic réseau, y compris les techniques basées sur les réseaux neuronaux et les techniques d'exploration de données. De même, divers modèles linéaires et non linéaires sont proposés pour la prévision du trafic sur le réseau. Plusieurs combinaisons intéressantes d'analyse de réseau et de techniques de prédiction sont mises en œuvre pour obtenir des résultats efficaces et efficaces.

Dans l'année 2015, La mesure et l'analyse du trafic Internet génèrent des ensembles de données qui sont des indicateurs de l'utilisation les tendances, et cet ensemble de données peut être utilisé pour la prévision de la circulation au moyen de diverses analyses statistiques. Dans cette étude, une analyse approfondie a été effectuée sur les données quotidiennes du trafic Internet dans l'université *School of (Computer Sciences, North Maharashtra University, Jalgaon (M.S))* sous le titre : **"A Review of Network Traffic Analysis and Prediction Techniques"**

Dans cet article, ils ont examiné les études précédentes sur l'analyse du trafic réseau. Ils ont demandé et discuté de diverses approches proposées pour analyser et prévoir le trafic réseau, y compris les techniques d'exploration de données, l'analyse des réseaux neuronaux et Ses composants, et les modèles de séries chronologiques linéaires et non linéaires. L'agencement tabulaire de tous les documents étudiés doit donner un aperçu des travaux de recherche dans l'analyse et la prévision du trafic réseau. Ensembles de données utilisés, algorithmes et paramètres mis en œuvre pour évaluer les résultats [19].

- **Dans Janvier 2016**, Oliveira, T.P., Barbar, J.S. and Soares, A.S.(2016) publié un article de titer : ***‘Computer network traffic prediction: a comparison between traditional and deep learning neural networks’***, Int. J. Big Data Intelligence, Vol. 3, No. 1, pp.28–37. Cet article compare quatre différentes méthodes de réseau de neurones artificiels pour prévision du trafic sur le réseau informatique, telles que :

- 1) perceptron multicouche (MLP) utilisant la rétropropagation comme algorithme d'entraînement
- 2) MLP avec rétropropagation résiliente (Rprop);
- 3) réseau neuronal récurrent (RNN);
- 4) Auto encodeur empilé d'apprentissage profondeur (SAE).

Le trafic du réseau informatique est échantillonné à partir du trafic des périphériques réseaux connectés à Internet. Il est montré ici comment un modèle de réseau neuronal plus simple, comme le RNN et le MLP, peut fonctionner encore mieux qu'un modèle plus complexe, comme le SAE. La prévision du trafic Internet est une tâche importante pour de nombreuses applications, telles que les applications adaptatives, le contrôle de la congestion, le contrôle des entrées, la détection des anomalies et l'attribution de la bande passante.

De plus, des méthodes efficaces de gestion des ressources, comme la bande passante, peuvent être utilisées pour améliorer le rendement et réduire les coûts, améliorant ainsi la qualité du service (QoS). La popularité des méthodes les plus récentes d'apprentissage profond a augmenté dans plusieurs domaines, mais il y a un manque d'études concernant la prédiction des séries temporelles, comme le trafic Internet [18].

- **Dans l'année 2017** university au Nigeria Department of Electrical and Information Engineering, Covenant University, Ota, Ogun State, Nigeria a publié un article sous titre: ***“Data mining approach for predicting the daily Internet data traffic of a smart university”*** Dans journal of big data.

Dans cette étude, une analyse approfondie a été réalisée sur les données quotidiennes de trafic Internet généré de janvier à décembre 2017 dans une université intelligente au Nigéria. Le jeu de données analysé contient sept éléments clés : le mois, la semaine, le jour de la semaine, le trafic IP quotidien de la veille, le trafic IP quotidien moyen des deux jours précédents, La classification de l'état du trafic (TSC) pour le téléchargement et la TSC pour le téléchargement Internet données de trafic. L'analyse d'exploration de données a été réalisée à l'aide de quatre algorithmes d'apprentissage : l'arbre de décision, l'ensemble d'arbres, la forêt aléatoire et l'algorithme Naïve Bayes sur l'application d'exploration de données KNIME (Konstanz Information Miner) et k-NN, Neural Algorithmes Network, Random Forest, Naïve Bayes et CN2

Rule Inducer sur l'Orange Plate-forme. Une analyse comparative des performances des modèles est présentée à l'aide de la matrice de confusion, valeur Kappa de Cohen, précision de chaque modèle, aire sous ROC Courbe, etc. Une précision minimale de 55,66% a été observée tant pour le téléchargement que pour le télécharger les données IP sur la plateforme KNIME avec une précision minimale de 57,3% et 51,4% respectivement ont été observés sur la plateforme Orange [5].

Dans l'année 2018, Department of Electronics and Telecommunications, Politecnico Di Torino, Torino, Italy, publié un article sous-titre « **An Application of Internet Traffic Prediction with Deep Neural Network** » par: Sanam Narejo and Eros Pasero

Dans cet article « Le document présente une application de la prévision du trafic Internet avec DBN, également l'étude des critères de sélection de la largeur de couche cachée dans les réseaux neuronaux profonds. Afin d'identifier la structure topologique appropriée, trois modèles de prédiction différents de l'architecture profonde ont été développés. Selon leur analyse comparative, il a été constaté que le nombre de neurones dans les couches cachées est crucial pour les réseaux plus profonds. Le modèle avec la largeur décroissante des couches cachées, bien qu'avec suffisamment d'unités cachées dans les couches supérieures, s'est avéré être un modèle prédictif compétent. Les résultats indiquent que les modèles I et II sont capables de saisir avec précision le point saillant caractéristiques du trafic réseau. Pour justifier davantage la faisabilité de notre approche pour la stratégie de taille de couche cachée, expériences ont été effectuées pour chaotique prédiction des séries chronologiques qui a également donné des résultats prometteurs Le modèle adopté pour produire des prévisions à long terme est conçu comme un travail futur [20].

- **Dans l'année 2019**, Un groupe de chercheurs (*Muhammad Faisal Iqbal, Muhammad Zahid, Durdana Habib, and Lizy Kurian John*) de diverses universités de *Islamabad, Pakistan* a publié un article sous le titre « **Efficient Prediction of Network Traffic for Real-Time Applications** »

Cet article explore un certain nombre de prédicteurs et recherche un prédicteur qui a une grande précision et une faible complexité des calculs et consommation d'énergie. De nombreux prédicteurs de trois classes différentes, y compris des séries chronologiques classiques, Les réseaux de neurones artificiels et les prédicteurs basés sur la transformée en ondelettes sont comparés. Ces prédicteurs sont évalués à l'aide de traces de réseau réel. Une comparaison de la précision et du coût, à la fois en termes de complexité de calcul et de consommation d'énergie, est présentée. On observe qu'un double prédicteur de lissage exponentiel fournit un compromis raisonnable entre les performances et les frais généraux.

Ils ont fourni une comparaison des performances et de la puissance de trois classes différentes de prédicteurs en utilisant un grand nombre de traces de réseau réel. Leurs résultats indiquent que le trafic réseau est généralement prévisible. De plus, le choix du prédicteur dépend des caractéristiques du réseau, ils ont trouvé différents prédicteurs adaptés aux traces provenant de différentes sources.

Le même prédicteur fonctionne toujours bien pour toutes les traces provenant de la même source. En outre, dans les applications en ligne critiques en énergie, DES et ARMA affichent une précision prometteuse avec une surcharge énergétique minimale. Le prédicteur basé sur l'ANN a toujours bien fonctionné, mais a une puissance et une surcharge de calcul élevées. Ils ont proposé une nouvelle métrique pour combiner précision et consommation d'énergie en un seul chiffre. Sur la base de cette métrique, le DES est devenu le prédicteur de choix lorsque la précision et la consommation d'énergie sont considérées collectivement [15].

3.3 Conclusion :

Dans ce chapitre, Nous avons étudié des articles dans ce domaine et leur avons donné une synthèse appropriée. de 2015 à 2021.

Chapitre 4

Implementation et realization

4.1 Introduction :

Au cours de ce chapitre, nous allons présenter notre travail qui consiste à concevoir et à réaliser la prévision de trafic internet dans l'université de m'sila. Celui-ci est basé sur les algorithmes d'apprentissages : les naïve bayésienne et arbre de décision et la méthode de K-NN. Nous commençons par présenter les différents outils utilisés : logiciel de WEKA et logiciel de R. Le principe d'analyse de donnée seront implémentés et intégrés dans la plateforme de data mining WEKA et R. WEKA est un plateforme open source qui rassemble une série de méthodes de fouilles de données. **R** est un langage de programmation et un logiciel libre destiné aux statistiques pour le développement de logiciels statistiques et l'analyse des données. Nous présentons aussi acquisition de données et méthodologie, puis nous montrons les résultats obtenus pour chaque partie et quelques interfaces de notre application.

4.2 Acquisition des données et méthodologie :

La mesure et l'analyse du trafic Internet génèrent des ensembles de données qui sont des indicateurs des tendances d'utilisation, et ces ensembles de données peuvent être utilisés pour la prédiction du trafic via diverses analyses statistiques. Dans cette étude, une analyse approfondie a été réalisée sur les données quotidiennes de trafic Internet générées de février à avril 2021 extrait dans un centre des réseaux et des systèmes d'information et de communication au université de M'sila.

L'ensemble de données analysé contient quatre caractéristiques clés : le jour, la semaine et mois et classification de trafic internet. L'analyse de l'exploration des données a été réalisée à l'aide de quatre algorithmes d'apprentissage : l'arbre de décision, l'algorithme de Naïve Bayes et les algorithmes kNN .

4.2.1 Fichier LOG :

Le fichier LOG est un fichier Texte, appelé aussi journal des connexions, qui conserve les traces des requêtes et des opérations traitées par le serveur. Généralement il est de la forme

suivante, pour laquelle les différents champs de ce fichier seront importés dans une base de données définie comme suit :

```
From 172.16.0.43 - 3 packets to tcp(800)
From 172.16.0.109 - 151 packets to tcp(800)
From 172.16.0.170 - 19 packets to udp(53) tcp(800)
From 172.16.0.171 - 4 packets to tcp(800)
From 172.16.0.172 - 44 packets to udp(53) tcp(800)
From 172.16.0.173 - 106 packets to udp(53) tcp(80)
From 172.16.0.176 - 7 packets to udp(53)
From 172.16.0.187 - 12 packets to udp(53) tcp(800)
From 172.16.0.196 - 5 packets to tcp(800)
From 172.16.0.198 - 10 packets to udp(53) tcp(800)
From 172.16.0.199 - 47 packets to udp(53) tcp(800,8829)
From 172.16.0.200 - 40 packets to tcp(445)
From 172.16.0.222 - 314 packets to tcp(800)
From 172.16.0.231 - 6 packets to udp(53)
From 172.16.0.245 - 10 packets to tcp(80)
From 172.16.1.21 - 78 packets to tcp(800)
From 172.16.1.111 - 6 packets to udp(53)
From 172.16.1.129 - 6 packets to tcp(443,800)
```

Tableau 4-1 Extrait de fichier log.

Selon ce format quatre informations sont enregistrées :

- Le nom du domaine ou l'adresse de Protocole Internet (IP) de la machine appelante,
- La date et l'heure de la requête.
- Taille de paquet.
- Les protocoles (TCP.UDP) et les ports utilisés.

Tout d'abord, il faut remarquer que les lignes arrivent dans un ordre chronologique des différentes requêtes et non pas regroupées par visiteur. Chaque ligne a un format bien défini. La première ligne servira d'exemple pour commenter les différents blocs de données.

- **172.16.0.43** : La première série de chiffres est l'adresse de Protocole Internet ou adresse IP.
- **3_packets** : la taille de paquet.
- **TCP (800)** : Transmission Control Protocol.
- **UDP (53)** : User Datagram Protocol.

4.2.1.1 Les informations dans un Fichier LOG :

Le comportement de l'utilisateur sur un site Web réside en une suite de clics de souris et de saisies sur un clavier. Ces informations déclenchent des requêtes qui ont pour résultat l'affichage de certaines pages du site. Ces requêtes sont enregistrées dans un fichier texte à mesure qu'elles sont déclenchées par les utilisateurs. Ces données sont stockées de manière standardisée de façon à ce qu'il soit possible de procéder à des analyses. Cette base de données est communément appelée fichier log. Son analyse permet en principe de savoir quelles sont les requêtes qui n'aboutissent pas (page manquante, lien erroné...) ou encore quelle est la fréquentation de chaque

page. Cependant la structure et le contenu de ce fichier permettent d'obtenir de plus amples informations après certains traitements.

Le fichier Log se transforme en une table composée de plusieurs colonnes. Chaque colonne correspond à un champ spécifique du fichier Log comme suit :

	A	B	C	D
42	lundi	troisième	mars	MDT
43	mardi	troisième	mars	MDT
44	mercredi	troisième	mars	LDT
45	jeudi	troisième	mars	MDT
46	vendredi	troisième	mars	MDT
47	samedi	troisième	mars	HDT
48	dimanche	troisième	mars	HDT
49	lundi	quatrième	mars	HDT
50	mardi	quatrième	mars	MDT
51	mercredi	quatrième	mars	MDT
52	jeudi	quatrième	mars	MDT
53	vendredi	quatrième	mars	MDT
54	samedi	quatrième	mars	MDT
55	dimanche	quatrième	mars	HDT
56	lundi	quatrième	mars	MDT
57	mardi	quatrième	mars	MDT
58	mercredi	quatrième	mars	HDT

Tableau 4-2 Les données de trafic internet

- La colonne « jour » correspond au jour d'accès
- La colonne « semaine » correspond à la semaine d'accès.
- La colonne « mois » correspond au mois d'accès.
- La colonne « classification » correspond au classification de Traffic internet, Nous l'avons classé sur la base suivante :

Classification	Taille de Traffic internet
Low data Traffic (LDT)	$0 < T < 1009$
Moderate data traffic (MDT)	$1010 < T < 9999$
Heavy data Traffic (HDT)	$10000 < T < 38444$

Tableau 4-3 classification de la taille du données de trafic internet .

4.3 Présentation de l'espace de travail :

Le choix de l'environnement de programmation convenable est très important pour le développement des projets. Nous commençons par présenter les différents outils utilisés : logiciel de WEKA et logiciel de R et RStudio.

4.3.1 Présentation de la plateforme WEKA :

WEKA (*Waikato Environment for Knowledge Analysis*) est un outil de fouille de données (*licence GNU*) développé en Java. Il a été créé à l'université de Waikato, en Nouvelle-Zélande, par un groupe de chercheurs issus de l'apprentissage automatique, de la reconnaissance de formes et de la fouille de données.

L'espace de travail **WEKA** contient une collection d'outils de visualisation et d'algorithmes pour l'analyse des données et la modélisation prédictive, allié à une interface graphique pour un accès facile de ses fonctionnalités

WEKA permet de prétraiter des données (*onglet Preprocess dans l'interface graphique*), faire de la classification supervisée (*Classify*) et non-supervisée (*Cluster*), des régressions (*Select Attributes*), rechercher des règles d'association (*Associate*), et de visualiser différentes représentations graphiques des données (*Visualize*).

C'est un logiciel « *open source* » gratuit dédié à la fouille de données. Il s'adresse à deux types de publics. D'un côté, il présente une interface graphique, le rendant ainsi accessible à une utilisation de type « *chargé d'études* » sur des données réelles. De l'autre, du fait que le code source est librement disponible et l'architecture interne très simplifiée, il se prête à une utilisation de chercheurs qui veulent avant tout expérimenter de nouvelles techniques en améliorant celles déjà implémentées ou en introduisant de nouvelles [13].

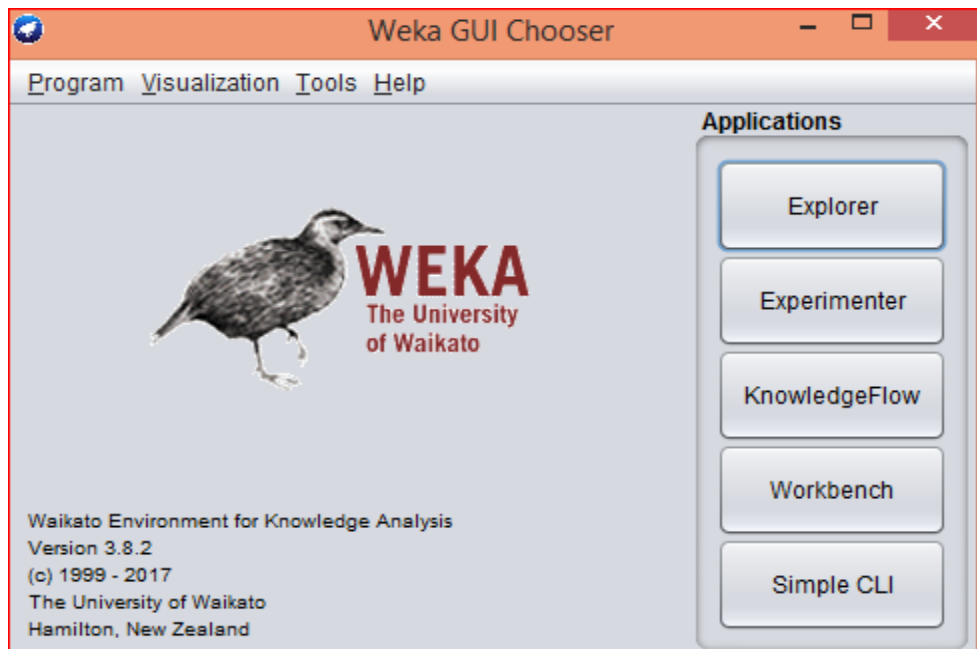


Figure 4-1 interface de logiciel WEKA .

4.3.2 Présentation de la plateforme R :

R est un langage de programmation et un logiciel libre destiné aux statistiques et à la science des données soutenu par la R .

R est un environnement permettant de faire des analyses statistiques et de produire des graphiques. C'est également un langage de programmation complet.

Le langage R est largement utilisé par les statisticiens, les *data miners*, *data scientists* pour le développement de logiciels statistiques et l'analyse des données.

L'objectif de logiciel R est simplement de nous familiariser avec R :

- Apprendre les bases du langage
- Apprendre à manipuler des données
- Apprendre à faire un graphique
- Apprendre à utiliser la documentation et le système d'aide

Interfaces : RStudio est un environnement de développement intégré qui permet de travailler en R, développer de nouvelles bibliothèques et travailler avec des notebooks.

L'interface RStudio est généralement composée de quatre fenêtres :

- La **première fenêtre** est dédiée à l'**édition** (ouverture, création, modification) des scripts.
- La **seconde fenêtre** contient la **console du logiciel**, qui permet l'exécution du code. Les lignes de commandes peuvent y être entrées directement, ou envoyées depuis la fenêtre supérieure en se positionnant en début de ligne et à l'aide des touches **CTRL+ENTER**
- la **troisième fenêtre** est dédiée à la **gestion du Workspace**, il s'agit de tous les objets (données, tableau, constante etc...) créés par l'exécution du code. Cette fenêtre permet également d'avoir accès à l'historique des commandes exécutées, ou encore de faire du versionnage.
- La **quatrième fenêtre** dispose également de plusieurs onglets qui permettent entre autres:
 - D'accéder à un **explorateur de fichier**.
 - La **visualisation** des plots réalisés.
 - L'importation **de packages**.
 - D'accéder à l'**aide** sur les fonctions.

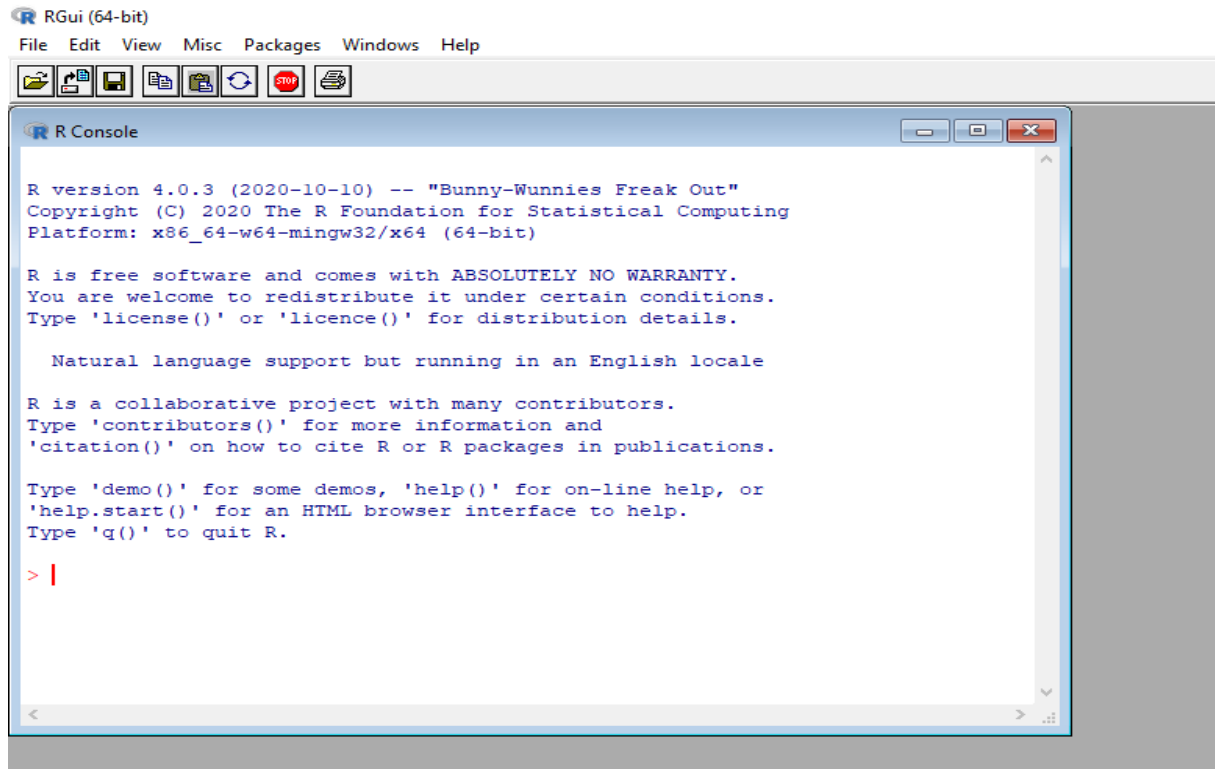


Figure 4-2 Interface de R sous Windows

RStudio est un environnement de développement intégré libre, gratuit, et qui fonctionne sous **Windows**, **Mac OS X** et **Linux**. Il complète **R** et fournit un éditeur de script avec coloration syntaxique, des fonctionnalités pratiques d'édition et d'exécution du code (comme l'autocomplétions), un affichage du code, de la console **R**, des fichiers, graphiques et pages d'aide, une gestion des extensions une intégration avec des systèmes de contrôle de versions comme **git**, etc. Il intègre de base divers outils comme par exemple la production de rapports au format **Rmarkdown**. Il est en développement actif et de nouvelles fonctionnalités sont ajoutées régulièrement. Son principal défaut est d'avoir une interface uniquement Anglophone.

4.4 Implémentation et Réalisation :

4.4.1 Le processus de classification par la méthode arbre de décision avec WEKA :

Pour faire la classification par des méthodes arbre de décision par WEKA, ce dernier a besoins d'un fichier de format spéciale qui s'appelle ARFF, ARFF définit des attributs par les utilisateurs, ces attributs sont de type classe.

La représentation externe d'une instance de classe se compose de :

- Un en-tête : Décrit les types d'attributs.
- Section de données : Liste CSV des données séparé.

WEKA traite des données contenues dans des fichiers respectant le format ARFF Attribute_Relation. Il s'agit de fichiers de type texte, décrivant des ensembles des tuples caractérisés par un certain nombre d'attributs communs.

```
@relation fichier
@attribute jour{samedi,dimanche,lundi,mardi,mercredi,jeudi,vendredi}
@attribute semaine {première,deuxième,troisième,quatrième}
@attribute mois {février,mars,avril}
@attribute classification {LDT,MDT,HDT}
@data
mercredi    première    février    HDT
jeudi       première    février    MDT
vendredi    première    février    MDT
samedi      première    février    HDT
dimanche    première    février    LDT
```

Tableau 4-4 extrait en fichier ARFF

Après le lancement du logiciel WEKA on obtient la fenêtre principale (**Figure 4-4**) En choix l'application que l'on désirée (pour notre travail Explorer), et sélectionner le fichier ARFF qui contient les données, Cliquer sur le bouton Explorer disponible dans le menu Application. Après l'ouverture de la fenêtre explorer le seul onglet visible est l'onglet Preprocess, Cliquez sur « Open file ». Pour notre exemple nous allons choisir le fichier « data1.arff »

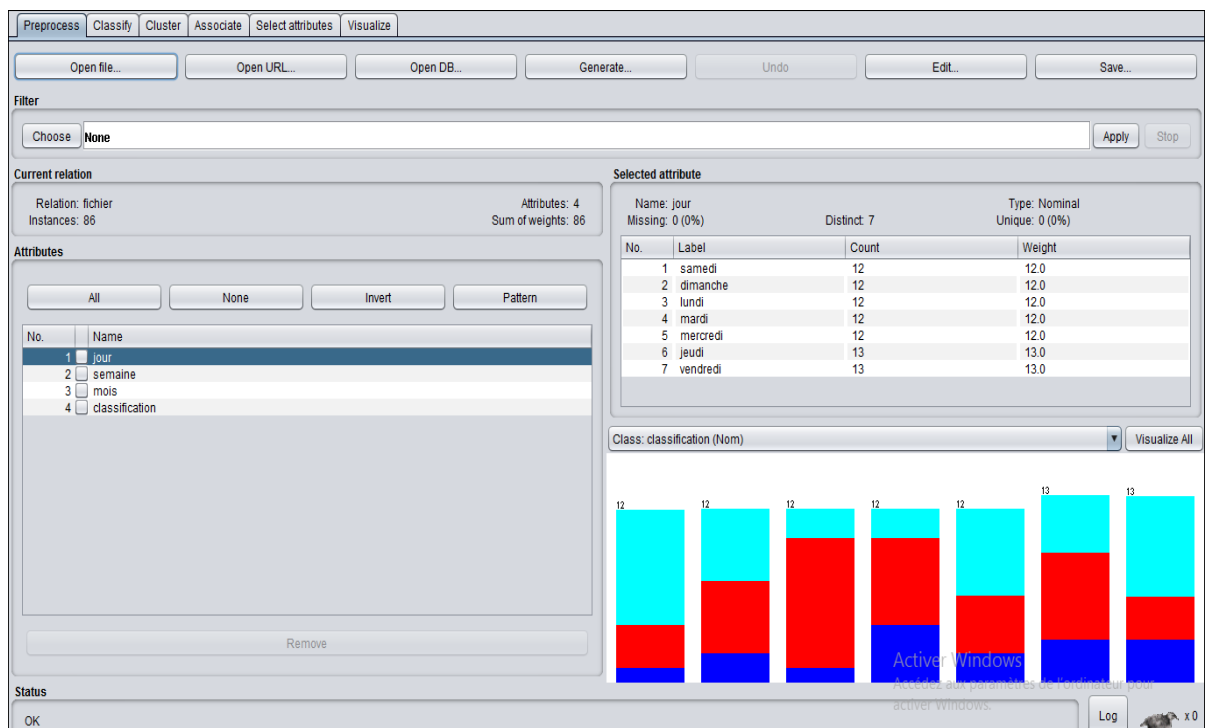


Figure 4-3 Sélection d'un fichier dans WEKA

Après la sélection du fichier, nous obtenons la fenêtre sous dessus qui affichée les attributs de fichier ARFF (Jour, Semaine, Mois, classification).

Pour lancer la classification, on clique sur le bouton **choose**, après on choisit la méthode **Trees.J48** parmi les méthodes de classification a base proposés dans WEKA.

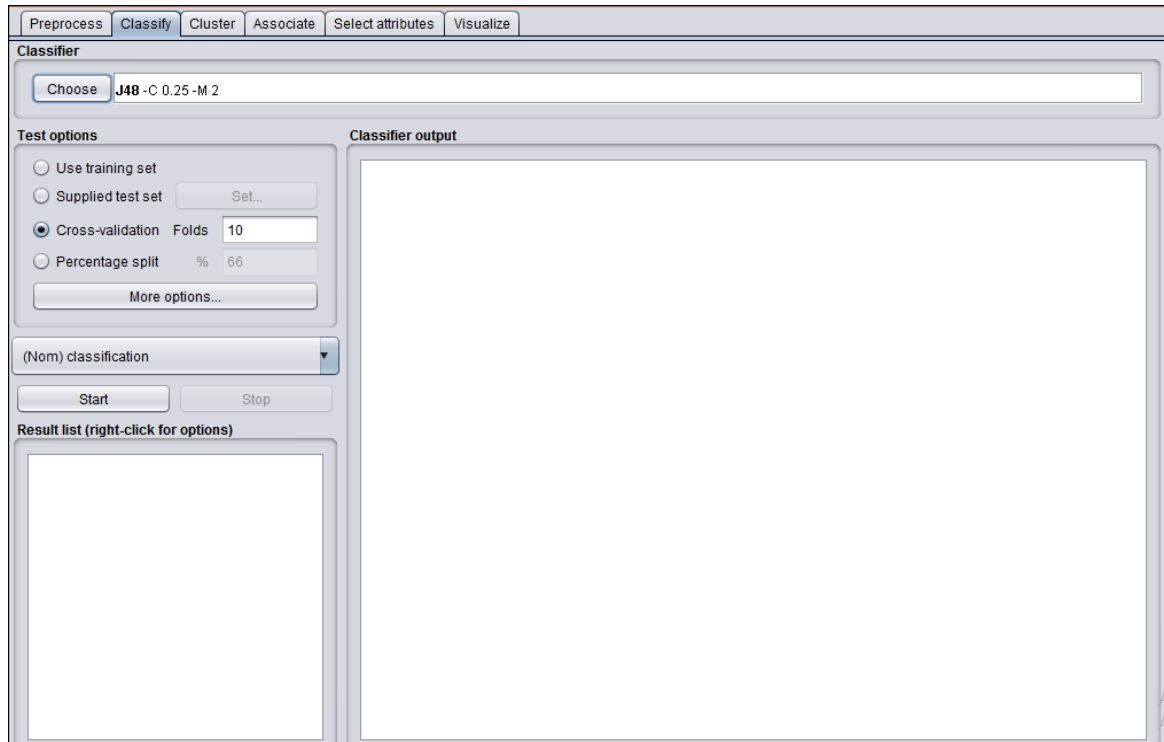


Figure 4-4 Choix de classifieur WEKA

Une fois que la méthode de classification est choisie on appuie sur le bouton **Start**, pour lancer l'exécution et prend les résultats.

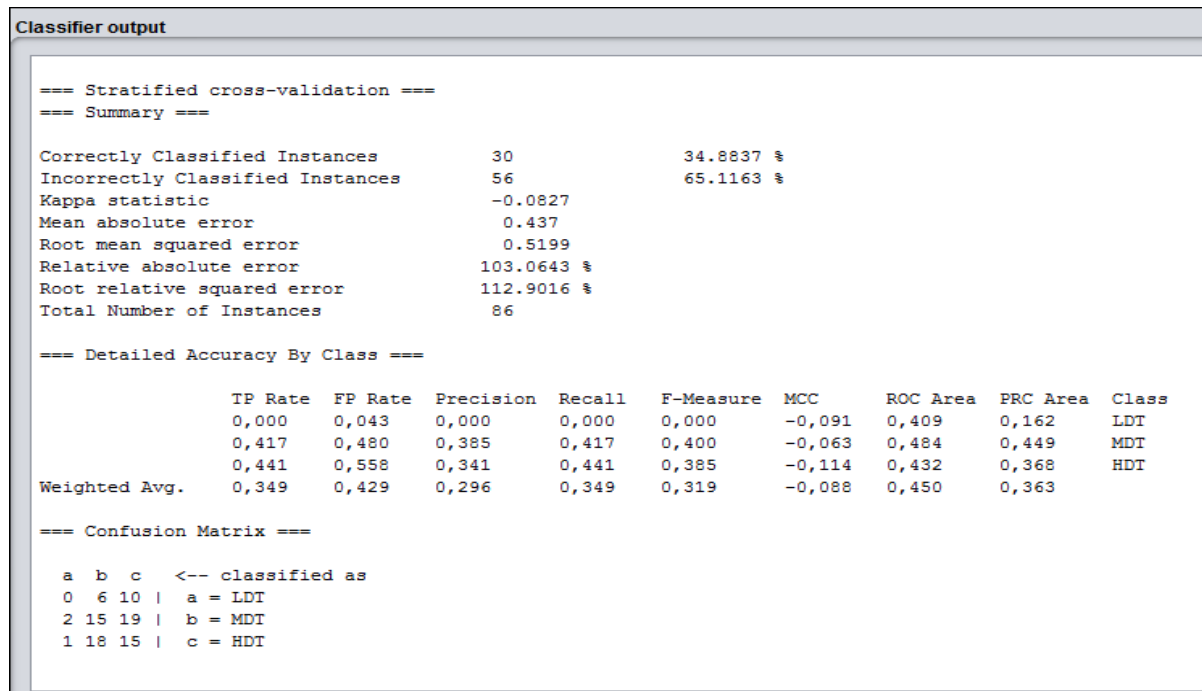


Figure 4-5 résultat des mesures de classification

Dans la partie **Classifier output** vous avez des statistiques sur le fichier exploité, à savoir le nombre d'instances **Total Number of Instances** de votre fichier est 86 instances, le nombre d'instances correctement classifiées **Correctly Classified Instances** est 30 instances et incorrectement classifiées **Incorrectly Classified Instances** est 56 instances avec la valeur de Kapp **kappa statistic** est 0.0827 et autres statistiques à découvrir. Sur le même écran, vous avez aussi **la matrice de confusion** présentée dans cette tableau :

	LDT	MDT	HDT
LDT	0	6	10
MDT	2	15	19
HDT	1	10	15

Tableau 4-5 matrice de confusion de l'arbre WEKA

Pour afficher l'arbre de décision, cliquer droit dans la partie **Result list (right-click for options)**. Lors de ce clic droit, un menu d'options s'affiche Choisir l'option **Visualize tree** comme suit :

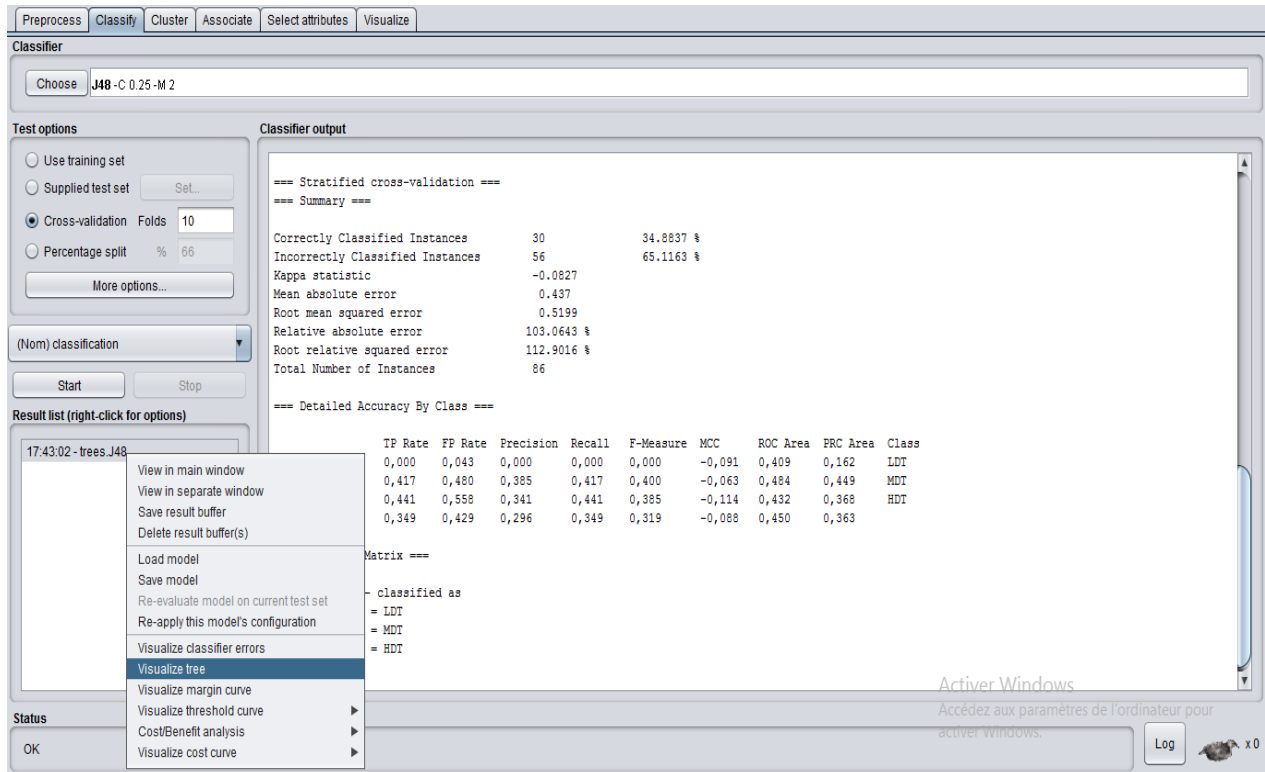


Figure 4-6 la méthode pour afficher l'arbre WEKA .

L'arbre de décision s'affiche ressemblant à ceci :

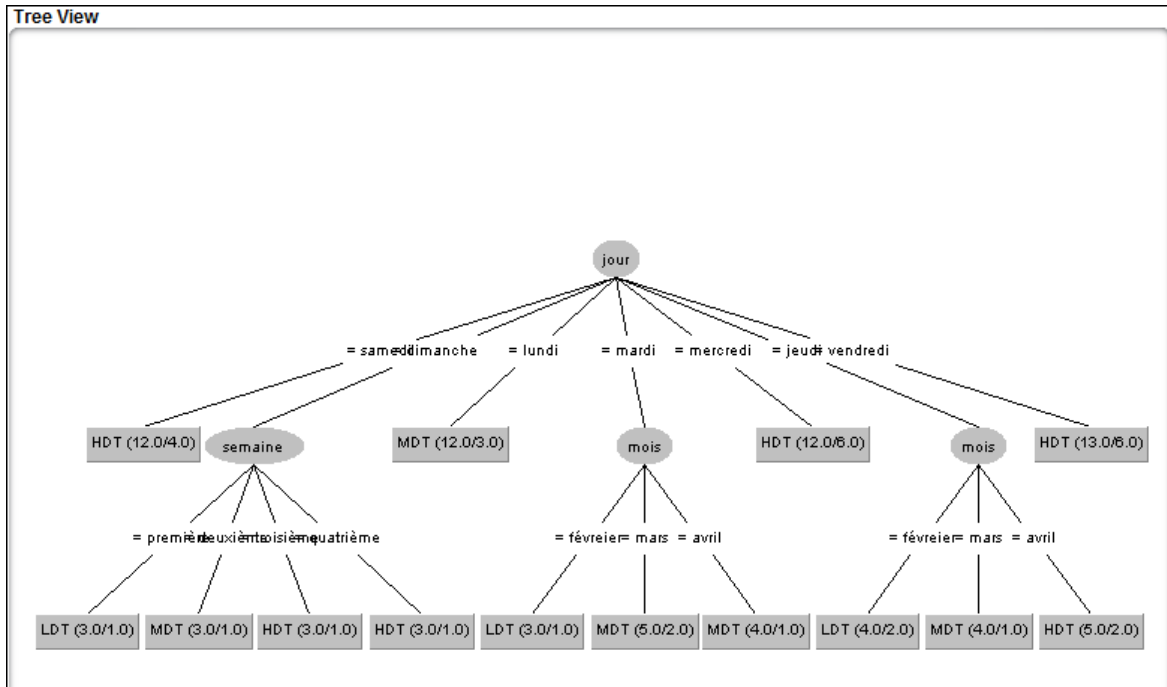


Figure 4-7 l'arbre résultat WEKA .

Dans cet exemple on peut analyser le résultat de la manière suivante.

Premièrement, nous avons Cinque catégories principales de jour :

- Tous les jours de samedi, mercredi et vendredi classifiées sur **HDT** parce que le Traffic internet de nos jours est rapide et à un taux élevé.
- Tous les jours de lundi classifié sur **MDT**, Cela signifie que le trafic Internet est à un taux moyen.
- Le trafic Internet varie les dimanches des quatre semaines de **février, mars** et **avril**, comme suit :
 - La première semaine, classifié sur **LDT**.
 - Le deuxième semaine, classifié sur **MDT**.
 - La troisième et quatrième semaine classifiée sur **HDT**.
- Les jours de mardi dans :
 - Le mois **février** classifié sur **LDT**.
 - Les mois **mars** et avril classifiés sur **MDT**.
- Le trafic Internet varie les jeudis au cours des trois mois, comme suit :
 - Le mois **février** classifié sur **LDT**.
 - Le mois **mars** classifié sur **MDT**.
 - Le mois **avril** classifié sur **HDT**.

4.4.2 Le processus de classification par arbre de décision avec R

Pour faire la classification par des méthodes arbre de décision par R , Après le lancement du logiciel WEKA on obtient la fenêtre principale (Figure 4-2) et cliquer sur le bouton « R-script » pour taper l’algorithme de la méthode

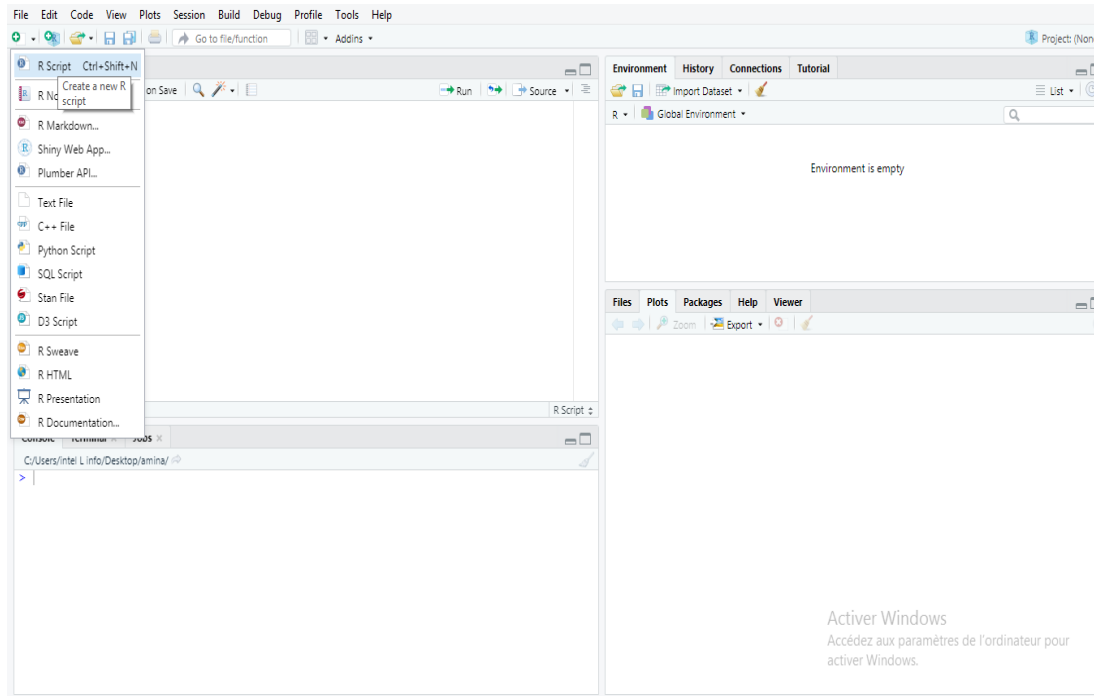


Figure 4-8 la page Principale de R.

On appuie sur le bouton «RUN», pour lancer l'exécution et prend les résultats comme suit :

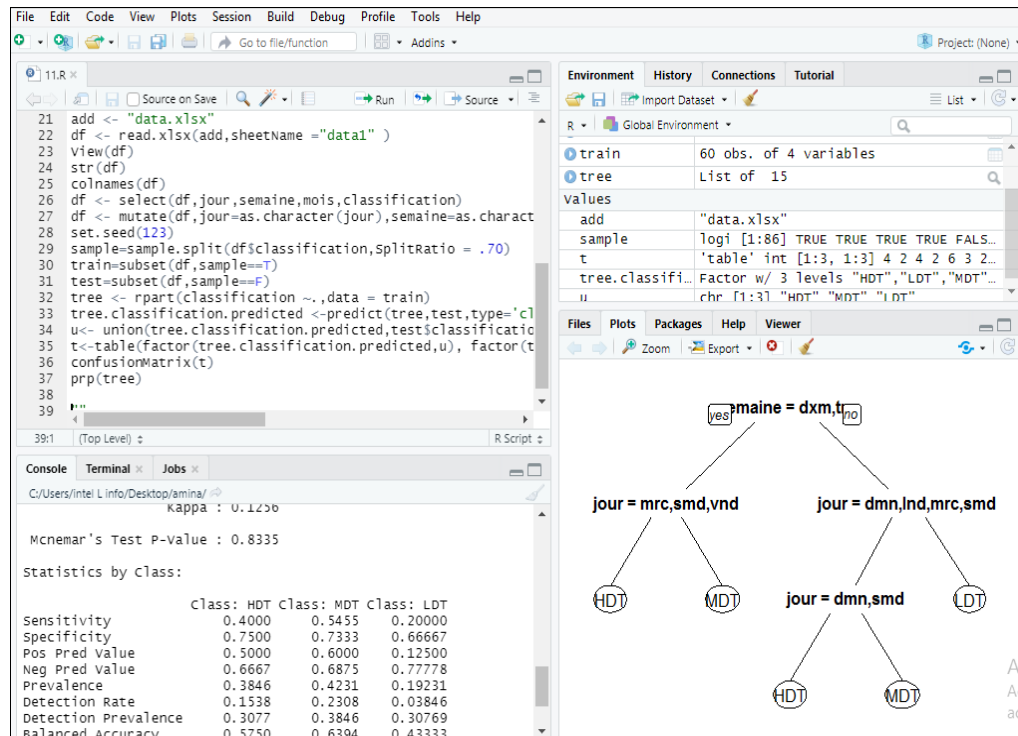


Figure 4-9 résultat de L'exécution de l'algorithme arbre de décision en R.

Dans la partie **Console** vous avez des statistiques et la matrice de confusion sur le fichier exploité

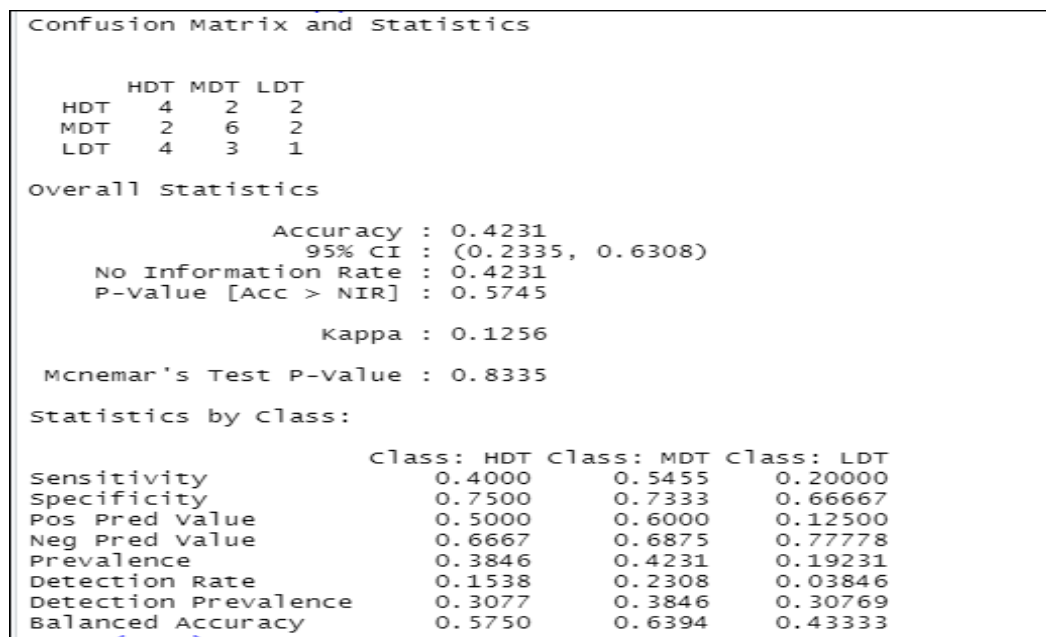


Figure 4-10 résultat des mesures de classification en R

	LDT	MDT	HDT
LDT	1	3	4
MDT	2	6	2
HDT	2	2	4

Tableau 4-6 matrice de confusion en R

Sur le même écran, vous avez aussi l'arbre dans la coté **Plot** comme suit :

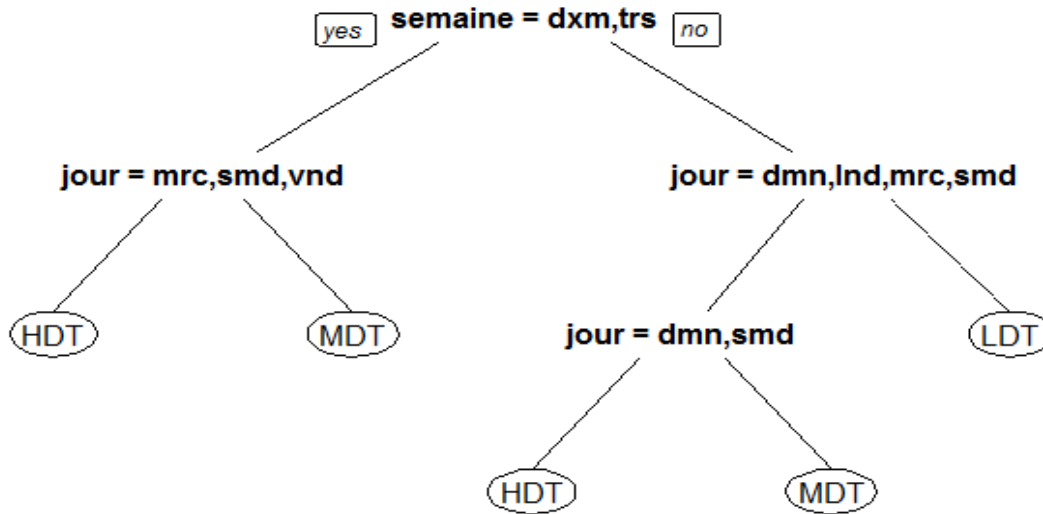


Figure 4-11 L'arbre de décision en R

Dans cet exemple on peut analyser le résultat de la manière suivante.

- Au cours de la deuxième et de la troisième semaine, le trafic Internet est parfois élevé ou moyen les mercredis, samedis et vendredis
- Comme pour la première et la quatrième semaine, le trafic internet est parfois élevé ou moyen les samedis et dimanches, mais les lundis et mercredis il est très faible.

Table : Comparaison de performance de algorithme de l'arbre de décision sur la plateforme WEKA et R .

	WEKA	R
Correct classified	54	42
Accuracy	62.7907 %	42.31 %
Wrong classified	32	58
Cohen's Kappa (k)	0.4012	0.1256
Error	37.2093 %	57.69%

Tableau 4-7 Comparaison des statistiques de Precision ,Recall et F_measure

precision	Recall	F_measure
WEKA		0
LDT 0	0	
MDT 0,714	0,556	0,625
HDT 0,583	0,824	0,683
R		1.53846
LDT 0.12500	0.20000	
MDT 0.6000	0.5455	0.57145
HDT 0.5000	0.4000	0.44444

Tableau 4-8 les résultats de R et WEKA

4.4.3 Le processus de classification par la méthode de Naïve Bayes avec WEKA

Maintenant, nous répétons les mêmes étapes que dans la méthode précédente mais ci, vous devez appuyer sur le bouton Choisir un classificateur et, dans le menu a « bayes », sélectionner NaiveBayes. Assurez-vous que l'attribut Play est sélectionné en tant que sélecteur de classe, puis appuyez sur le bouton « start » pour créer un modèle.

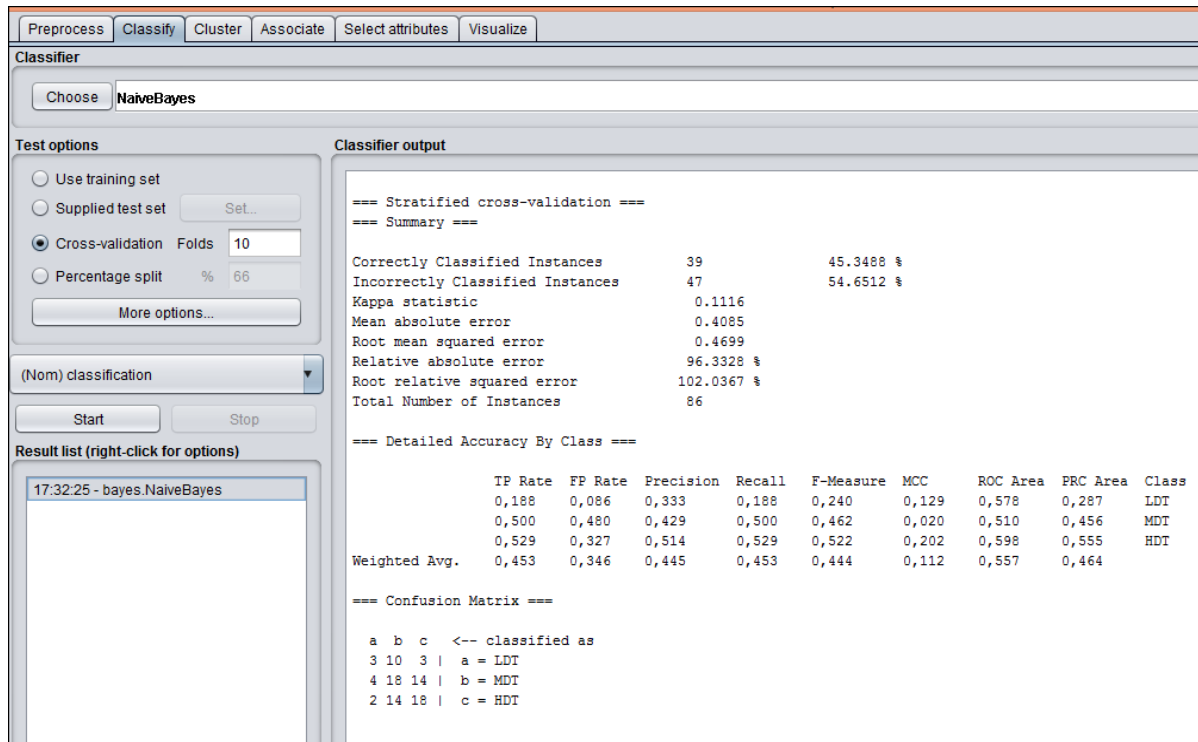


Figure 4-12 résultat du classificateur Naivebayes.

Le modèle génère des informations sur la précision de sa classification et d'autres paramètres.

Correctly Classified Instances 45.3488 %, Incorrectly Classified Instances 54.6512% avec la valeur de Kapp **kappa statistic est 0.1116** et autres statistiques à découvrir. Sur le même écran, vous avez aussi la **matrice de confusion** présentée dans cette tableau :

	LDT	MDT	HDT
LDT	2	10	3
MDT	4	18	14
HDT	2	14	18

Tableau 4-9 matrice de confusion de Naivebayes en WEKA.

4.4.4 Le processus de classification par des méthodes naïves bayes à travers R :

Nous créons un autre R_script pour l'algorithme Naïve bayes

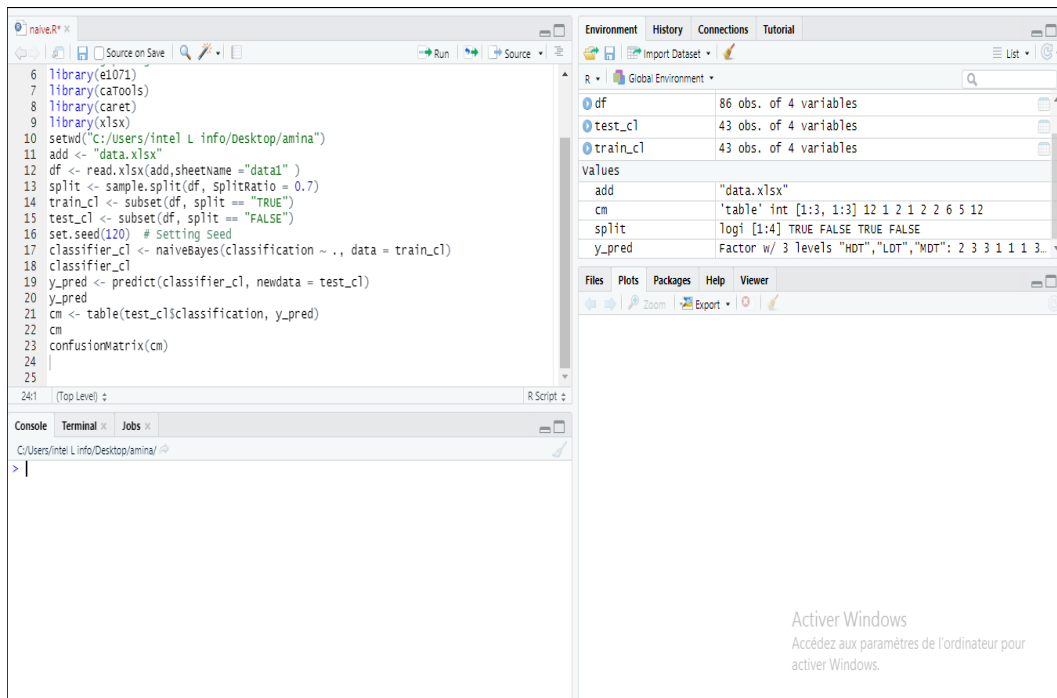


Figure 4-13 l'algorithme de Naivebayes sur R

Le résultat d'exécution « RUN » sera comme ça :

Confusion Matrix and Statistics				
	y_pred			
	HDT	LDT	MDT	
HDT	12	1	6	
LDT	1	2	5	
MDT	2	2	12	
Overall statistics				
	Accuracy : 0.6047			
	95% CI : (0.4441, 0.7502)			
	No Information Rate : 0.5349			
	P-Value [Acc > NIR] : 0.2230			
	Kappa : 0.3676			
	McNemar's Test P-Value : 0.3496			
Statistics by Class:				
	Class: HDT	Class: LDT	Class: MDT	
Sensitivity	0.8000	0.40000	0.5217	
Specificity	0.7500	0.84211	0.8000	
Pos Pred Value	0.6316	0.25000	0.7500	
Neg Pred Value	0.8750	0.91429	0.5926	
Prevalence	0.3488	0.11628	0.5349	
Detection Rate	0.2791	0.04651	0.2791	
Detection Prevalence	0.4419	0.18605	0.3721	
Balanced Accuracy	0.7750	0.62105	0.6609	

Figure 4-14 résultat des mesures de classification par Naivebayes

	LDT	MDT	HDT
LDT	2	5	1
MDT	2	12	2
HDT	1	6	12

Tableau 4-10 matrice de confusion de Naivebayes en R

	WEKA	R
Correct classified	39	60
Accuracy	45.3488 %	60.47%
Wrong classified	47	40
Cohen's Kappa (k)	0.1116	0.3676
Error	54.6512	39.53%

Tableau 4-11 comparaison de performance de algorithme de Naivebayes sur plateforme WEKA et R .

	precision	Recall	F_measure
WEKA			0.240
LDT	0.333	0.188	
MDT	0.429	0,500	0,462
HDT	0,514	0,529	0,522
R			0.307
LDT	0.25000	0.40000	
MDT	0.7500	0.5217	0.6152
HDT	0.6316	0.80000	0.7058

Tableau 4-12 Comparaison des statistiques de Precision ,Recall et F_measure en R

4.5 Conclusion :

Ce chapitre présente la description et la mise en œuvre des étapes implémentées pour notre approche. Après la génération de données de fichier ARFF dans WEKA et la génération de fichier Excel dans R pour faire la classification et comparaison des statistiques, on conclurons que les résultats obtenir on peut dire que la méthode d'arbre de décision donnée les bonnes résultats.

CONCLUSION GENERALE

La surveillance et la mesure du trafic de données Internet sont vitales pour les opérations des fournisseurs de services Internet, et cela peut être réalisé en utilisant une approche de surveillance du trafic basée sur les flux. Les données de trafic Internet enregistrées acquises grâce à la surveillance du trafic contiennent des informations et des connaissances utiles accessibles via l'analyse des données. Dans cette étude, les données de trafic Internet générées à l'Université Mohamed Boudiaf, en Algérie pour l'année 2021, ont été analysées statistiquement et des modèles prédictifs basés sur WEKA et R ont été développés pour prévoir le trafic de données Internet un jour donné en utilisant les données de trafic des mois précédents. L'arbre de décision et les algorithmes de data mining de Naïve Bayes ont été appliqués sur les modèles WEKA et R

Les algorithmes ont été efficacement entraînés avec 43% des échantillons d'ensembles de données tandis que les 43% restants ont été appliqués pour l'évaluation du modèle. Le résultat de l'évaluation des performances du modèle montre que le prédicteur Naïve Bayes avait la meilleure précision tandis que le prédicteur Arbre de décision avait le moins de précision sur R. Le prédicteur Naïve Bayes avait le moins de précision tandis que le prédicteur Arbre de décision avait la meilleure précision sur WEKA. La précision la plus faible enregistrée pour tous les cas considérés est de 42,31 % tandis que la précision maximale est de 62,7907 %. Cela montre que l'approche d'exploration de données utilisant des workflows de pipeline de données interactifs et visuels est raisonnablement précise pour prédire les tendances du trafic Internet dans une université intelligente, mais des études supplémentaires seront nécessaires afin d'améliorer les performances des modèles.

Bibliographie

- [1] Meryem, M. S. (s.d.). " Traitement des données biologiques par les méthodes ensemblistes ". *thèse de doctorat*. Informatique: université de telemcen.
- [2] fatiha, b. (2013). «contribution à la gatégorisation de textes et l'extraction d'information». *thèse de doctorat*. informatique: université de oran.
- [3] ZHANI, M. F. (2011, juin). " PRÉVISION DU TRAFIC INTERNET -MODÈLES ET APPLICATIONS ". *THESE DU DOCTORAT EN INFORMATIQUE*. UNIVERSITÉ DU QUÉBEC À MONTRÉAL.
- [4] Mezghani, N. (2019, janvier). "*Arbres de décision*". TELUQ, INFORMATIQUE.
- [5] Aderibigbe Israel Adekitan*, J. A. (2019). " Data mining approach for predicting the daily Internet data traffic of a smart university ". *journal of big data*.
- [6] Hanane, T. (2018). «Classification automatique de textes». *mémoire de master*. m'sila: université mohamed boudiaf.
- [7] JALAM, R. (2003, juin). « Apprentissage automatique et catégorisation de textes multilingues ». *Thèse de doctorat*. Université Lumière Lyon 2, France.
- [8] OUFELLA, Y. (2008, septemeber). " ÉVOLUTION DU CONCEPT DE FRONT ROC ET COMBINAISON DE CLASSIFIEUR ". *MEMOIRE DE MASTER*. UNIVERSITÉ DE ROUEN.
- [10] Azencott, p. C.-A. (2020, 12 15). Récupéré sur openclassrooms: <https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308256-evaluez-un-algorithme-de-classification-qui-retourne-des-valeurs-binaires>
- [11] B.Agard, A. (2005, Avril). « Exploration des bases de données industrielles à l'aide du Data Mining – Perspectives ». *9ème colloque national AIP PRIMECA*.
- [12] R. Lefébure, G. (1998). « *Le Data Mining* ». Edition EYROLLES.
- [13] O.CHOAYB. (2013-2014). «Classification automatique de textes ». *Mémoire de Master*. Université Mohamed BOUDIAF– Msila, Algérie.

- [14] S.Raheel. (2010). « L'Apprentissage Artificiel pour la Fouille de Données Multilingues: Application à la Classification Automatique des Documents Arabes ». *Thèse de doctorat en Sciences de l'Information et de la Communication*. Université Lumière Lyon 2.
- [15] Muhammad Faisal Iqbal, M. Z. (2019). "Efficient Prediction of Network Traffic for Real-Time Applications". *Journal of Computer Networks and Communications*.
- [16] (s.d.). Récupéré sur techno-science.net: <https://www.techno-science.net/glossaire-definition/Classification-naive-bayesienne.html>
- [18] "Computer network traffic prediction: A comparison between traditional and deep learning neural networks". (January 2016). *International Journal of Big Data Intelligence*.
- [19] Manish R. Joshi, T. H. (s.d.). "A Review of Network Traffic Analysis and Prediction Techniques". School of Computer Sciences, North Maharashtra University.
- [20] Sanam Narejo, E. P. (2018). "An Application of Internet Traffic An Application of Internet Traffic". *Springer International Publishing AG*, DOI 10.1007/978-3-319-56904-8_14.

ملخص

إن قياس وتحليل حركة الإنترنت يولد بيانات تُستخدم كمؤشرات لاتجاه الاستخدام. يمكن أيضًا استخدام هذه البيانات للتنبؤ بحركة المرور على الإنترنت. في هذا المشروع، قمنا بإجراء تحليل متعمق لحركة الإنترنت اليومية المتولدة في أراضي جامعة المسيلة يتم استخدام البيانات التي تم إنشاؤها للتنبؤ بحركة المرور على الإنترنت داخل الجامعة نستخدم منصات التنقيب عن البيانات مثل تطبيق R و weka. العديد من خوارزميات التعلم بما في ذلك أشجار القرار وتصنيف بايز .

الكلمات المفتاحية: التنقيب عن البيانات، التنبؤ ، حركة الإنترنت ، جامعة المسيلة

Abstract:

The measurement and analysis of Internet traffic generates data that is used as an indicator of usage trends. This data can also be used for Internet traffic forecasting. In this project, an in-depth analysis of the daily Internet traffic generated on the campus of M'sila University was conducted. The data generated is used to predict Internet traffic within the university. We use data mining platforms like weka and R as applications Several learning algorithms, including decision trees and Bayesian classification.

Keywords: data mining, forecasting, Internet traffic, M'sila University.

Résumé :

La mesure et l'analyse du trafic internet génèrent des données utilisées comme indicateurs de tendance d'utilisation. Ces données peuvent être également utilisées pour la prédiction du trafic internet. Dans ce projet, l'objectif est de réaliser une analyse approfondie du trafic quotidien internet généré dans l'enceinte de l'université de M'sila. Les données générées sont utilisées pour la prédiction du trafic internet au sein de l'université. Nous utilisons des plateformes de data mining comme weka et R pour l'application de plusieurs algorithmes d'apprentissage dont les arbres de décision et la classification bayésienne.

Mots clés : data mining, prédiction, trafic internet, université de M'sila.