

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université Mouloud Mammeri de Tizi Ouzou  
Faculté de Génie électrique et d'informatique  
Département informatique



# Mémoire de fin d'études

En vue de l'obtention du diplôme de

Master en Informatique

Spécialité : Conduite de Projet Informatique

Thème :

Utilisation d'une méthode d'apprentissage supervisée pour la  
detection de l'attrition d'un client Djezzy

Présenté par :

BOUALI Hakima-Cyilia

Dirigé par :

Mme BOUARAB Farida

Membres de jury :

---

- Mr. SADI Sami
- Melle SEGHIRI Narimène
- Mme. BELLATAF Samia
- Mme BOUARAB Farida

Président de jury  
Examinatrice  
Examinatrice  
Promotrice

---

## *Remerciements*

---

*Je remercie Allah de m'avoir donné le courage, la santé et la motivation pour finir ce projet de fin d'études.*

*Je tiens à remercier particulièrement ma promotrice Madame BOUARAB Farida pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.*

*Il est essentiel pour moi de remercier les professionnels rencontrés au sein de l'entreprise DJEZZY qui m'ont accordé leur temps pour répondre à mes questions et je remercie plus particulièrement mon encadreur Monsieur HANNACHI Radhouane pour ses directives précieuses et son appui considérable dans ma démarche.*

*Je tiens à remercier les membres du jury d'avoir bien accepté de consacrer leur temps pour évaluer ce modeste travail.*

*Je souhaite vivement remercier ma famille pour leur soutien et leur compréhension tout au long de mes années d'études.*

*Enfin, je n'oublie pas de remercier mes amis d'études et tous ceux qui ont contribué de près ou de loin à ma formation et à l'aboutissement de ce projet.*

---

## *Dédicaces*

---

*A la mémoire de mon père.*

*A ma chère maman.*

*A mes deux chères frères Bélaïd et Aghilès.*

---

## *Résumé*

---

La perte de clientèle est un problème majeur des grandes entreprises en raison de son effet direct sur les revenus de celles-ci, en particulier dans le secteur de télécommunication. Les entreprises cherchent à mettre au point des moyens pour prédire l'attrition des clients. La principale contribution de notre travail est de développer un modèle de prévision du taux de désabonnement qui aide les opérateurs de télécommunications à prédire les clients les plus susceptibles de quitter. Pour développer un modèle performant répondant à cet effet, nous mettons en œuvre divers algorithmes d'apprentissage supervisé et choisissons le meilleur selon son taux de précision.

**Mots clés :** apprentissage automatique, apprentissage supervisé, prédiction de l'attrition des clients, classification supervisée.

## Table des matières

|  |    |
|--|----|
| Introduction générale .....  | 8  |
| Chapitre I.....  | 11 |
| Apprentissage automatique.....   | 11 |
| I.1. Introduction .....  | 12 |
| I.2. Apprentissage automatique.....  | 12 |
| I.1.1. Définition.....   | 13 |
| I.1.2. Types d'apprentissage automatique .....                                       | 13 |
| I.1.3. Objectifs des méthodes d'apprentissage.....                                   | 15 |
| I.1.4. Types de données.....   | 16 |
| I.3. Algorithmes d'apprentissage automatique .....                                   | 18 |
| I.3.1. Algorithmes d'apprentissage supervisé.....                                    | 18 |
| I.3.2. Algorithmes d'apprentissage non supervisé.....                                | 37 |
| I.4. Quelques exemples d'application des algorithmes de l'apprentissage supervisé .. | 42 |
| I.5. Conclusion .....  | 44 |
| Chapitre II .....  | 45 |
| Domaine d'étude.....   | 45 |
| II.1. Introduction .....   | 46 |
| II.2. Etude de l'existant.....   | 46 |
| II.2.1. Fiche signalétique de Djezzy.....  | 46 |
| II.2.2. Historique .....   | 47 |
| II.2.3. Structure fonctionnelle de Djezzy .....                                      | 48 |
| II.2.4. Domaine d'étude : .....  | 50 |
| II.2.5. Besoin de l'application à réaliser .....                                     | 51 |
| II.3. Conclusion.....  | 51 |
| Chapitre III.....  | 52 |
| Conception .....   | 52 |
| III.1. Introduction.....   | 53 |
| III.2. Approche proposée .....   | 53 |
| III.3. Attrition (churn en anglais).....   | 53 |
| III.4. Cycle de vie de notre projet .....  | 54 |
| III.5. Architecture du projet .....  | 55 |

|   |    |
|---|----|
| III.6. Algorithmes à utiliser.....                            | 56 |
| III.7. Conclusion .....                                       | 56 |
| Chapitre IV.....  | 57 |
| Implémentation .....  | 57 |
| IV.1. Introduction.....                                       | 58 |
| IV.2. Logiciels et librairies .....                           | 58 |
| IV.2.1. Logiciels .....                                       | 58 |
| IV.2.2. Librairies.....                                       | 58 |
| IV.3. Mise en œuvre.....                                      | 59 |
| IV.3.1. Compréhension du problème métier.....                 | 59 |
| IV.3.2. Compréhension des données .....                       | 59 |
| IV.3.3. Préparation des données .....                         | 63 |
| IV.3.4. Modélisation .....                                    | 67 |
| IV.3.5. Evaluation .....                                      | 71 |
| IV.4. Résultats et discussion .....                           | 75 |
| IV.5. Implémentation du meilleur modèle dans notre étude..... | 77 |
| IV.6. Perspectives .....                                      | 78 |
| IV.7. Conclusion .....  | 78 |
| Conclusion générale.....                                      | 79 |
| Bibliographie.....  | 81 |

---

## *Table des figures*

---

|  |    |
|--|----|
| FIGURE I.1 : EXEMPLE D'UN ARBRE DE DECISION .....  | 26 |
| FIGURE I.2 : ILLUSTRATION DU FONCTIONNEMENT D'UNE FORET ALEATOIRE. ....  | 29 |
| FIGURE I.3 : ARCHITECTURE GENERALE D'UN RESEAU DE NEURONE.....   | 33 |
| FIGURE I.4 : SCHEMA D'UN PERCEPTRON. ....  | 34 |
| FIGURE I.5 : SCHEMA DU PERCEPTRON MULTI COUCHES.....   | 34 |
| FIGURE I.6 : EXEMPLE DE CLUSTERING HIERARCHIQUE ASCENDANT POUR QUELQUES VILLES DE<br>FRANCE QUI ONT DES PROFILS METEO SIMILAIRES. .... | 39 |
| FIGURE I.7 : EXEMPLE DE DENDROGRAMME.....  | 40 |
| FIGURE I.8 : ALGORITHME K-MEANS.....   | 42 |
| FIGURE III.1 : CYCLE DE VIE D'UN PROJET MACHINE LEARNING.....  | 54 |
| FIGURE III.2 : ARCHITECTURE DE L'APPLICATION.....  | 55 |
| FIGURE IV.1 : SELECTION DES VARIABLES AVEC CHI-SQUARED. ....   | 65 |
| FIGURE IV.2 : CODE POUR PARTITIONNER LES DONNEES.....  | 67 |
| FIGURE IV.3 : DISTRIBUTION DE LA CLASSE CIBLE.....   | 68 |
| FIGURE IV.4 : IMPLEMENTATION DE L'ALGORITHME SMOTE SOUS R.....   | 69 |
| FIGURE IV.5 : CREATION D'UN DATA FRAME D'APPRENTISSAGE EQUILIBRE. ....   | 69 |
| FIGURE IV.6 : DISTRIBUTION DE LA CLASSE CIBLE APRES SUR-ECHANTILLONNAGE.....   | 69 |
| FIGURE IV.7 : ALGORITHME DE L'ARBRE DE DECISION. ....  | 70 |
| FIGURE IV.8 : ARBRE DE DECISION OBTENU.....  | 70 |
| FIGURE IV.9 : ALGORITHME DE FORET DES ARBRES DECISIONNELS. ....  | 70 |
| FIGURE IV.10 : ALGORITHME DE REGRESSION LOGISTIQUE.....  | 71 |
| FIGURE IV.11 : ALGORITHME DE RESEAU DE NEURONES.....   | 71 |
| FIGURE IV.12 : GRAPHIQUE DES VARIABLES IMPORTANTES PAR ORDRE CROISSANT. ....   | 76 |
| FIGURE IV.13 : APERÇU DES DONNEES DE LA COLONNE « SCORE ».....   | 77 |

---

## *Table des tableaux*

---

|  |    |
|--|----|
| TABLEAU I.1 : EXEMPLE DE PREDICTION DE LA TAILLE D'UN TEE-SHIRT .....  | 20 |
| TABLEAU I.2 : EXEMPLE A PREDIRE. ....  | 21 |
| TABLEAU I.3 : CALCUL DE DISTANCE POUR CHAQUE EXEMPLE. ....   | 21 |
| TABLEAU I.4 : TABLEAU ILLUSTRANT LES 3-PLUS PROCHES VOISINS. ....  | 22 |
| TABLEAU I.5 : EXEMPLES DE DETECTION DE POURRIELS.....  | 24 |
| TABLEAU I.6 : PROBABILITE A PRIORI DES CLASSES. ....   | 24 |
| TABLEAU I.7 : PROBABILITE DE CHAQUE VARIABLE $x_i$ SACHANT Y. ....   | 25 |
| TABLEAU I.8 : RESULTATS DE PREDICTION AVEC LE NAÏF BAYESIEN. ....  | 25 |
| TABLEAU II.1 : FICHE SIGNALÉTIQUE DE DJEZZY .....  | 47 |
| TABLEAU IV.1: DESCRIPTION DES VARIABLES DE LA TRAME DE DONNEES.....  | 63 |
| TABLEAU IV.2 : RESULTATS DE DETECTION DES VALEURS EXTREMES. ....   | 64 |
| TABLEAU IV.3 : LISTE DES VARIABLES PERTINENTES RETOURNEE PAR LA CHI-SQUARED.....   | 66 |
| TABLEAU IV.4 : LISTE DES VARIABLES PERTINENTES FINALES SELECTIONNEE.....   | 67 |
| TABLEAU IV.5 : EXEMPLE DE MATRICE DE CONFUSION.....  | 72 |
| TABLEAU IV.6 : MATRICE DE CONFUSION DE L'ARBRE DE DECISION.....  | 73 |
| TABLEAU IV.7 : RESULTATS DE L'EVALUATION DE L'ARBRE DE DECISION. ....  | 73 |
| TABLEAU IV.8 : MATRICE DE CONFUSION DE FORET DE L'ARBRE DECISIONNEL.....   | 73 |
| TABLEAU IV.9 : RESULTATS DE L'EVALUATION DES FORETS ALEATOIRES.....  | 74 |
| TABLEAU IV.10 : MATRICE DE CONFUSION DU MODELE REGRESSION LOGISTIQUE. ....   | 74 |
| TABLEAU IV.11 : RESULTATS DE L'EVALUATION LA REGRESSION LOGISTIQUE.....  | 74 |
| TABLEAU IV.12 : MATRICE DE CONFUSION DES RESEAUX DE NEURONES.....  | 75 |
| TABLEAU IV.13 : RESULTATS DE L'EVALUATION DU RESEAU DE NEURONES. ....  | 75 |
| TABLEAU IV.14 : COMPARAISON DES RESULTATS DES MESURES D'EVALUATION DES POUR<br>CHAQUE ALGORITHME D'APPRENTISSAGE SUPERVISE. .... | 75 |

# *Introduction générale*

## Contexte du travail

La perte de la clientèle est l'un des problèmes de plus en plus préoccupants du secteur des télécommunications. Le progrès technique et le nombre croissant d'opérateurs ont accru le niveau de concurrence. En raison du coût élevé de l'acquisition de nouveaux clients, le secteur des télécommunications est passé de la stratégie de l'acquisition à celle de la fidélisation des clients déjà existants. Pour appliquer cette stratégie, les entreprises doivent réduire le potentiel de l'attrition des clients (customer churn en anglais). D'autre part, la prévision des clients susceptibles de quitter l'entreprise représentera une source de revenus supplémentaire potentiellement importante si cela est fait à un stade précoce.

De nombreuses recherches ont confirmé que la technologie d'apprentissage automatique est très efficace pour prévoir cette situation. Cette technique est appliquée en tirant partie des données contenant des informations historiques sur les clients de l'entreprise.

L'apprentissage automatique est l'art de l'analyse prédictive dans lequel un système est formé sur un ensemble de données pour en apprendre à partir des modèles, puis testé pour effectuer des prédictions sur un nouvel ensemble de données. Plus les prédictions sont précises, plus le modèle est performant.

## Problématique

La perte des clients communément appelé « attrition » constitue un vrai problème dans les entreprises évoluant dans les différents secteurs d'activités surtout en situation de concurrence où l'industrie de la télécommunication en fait partie.

Djezzy est une entreprise qui évolue dans ce secteur et souffre de ce phénomène et est mise en concurrence dans le marché de la télécommunication en Algérie.

Afin de parvenir à réduire la rétention des clients, on devrait commencer par connaître qui sont les clients susceptibles de partir ? Quelle est la probabilité qu'un client parte ? Et pouvons-nous les prédire à l'avance ?

## Contributions

Pour y parvenir au problème cité auparavant, nous pensons mettre en place des techniques d'apprentissage supervisé en regroupant des données sur les clients Djezzy à une période avec leurs différentes caractéristiques.

## Organisation du mémoire

Ce mémoire est organisé en 4 chapitres en plus de l'introduction et de la conclusion générale.

- Dans le premier chapitre nous aborderons les deux techniques les plus populaires de l'apprentissage automatique ainsi que quelques généralités sur les algorithmes de celui-ci ;
- Dans le second chapitre nous étudierons le domaine de notre recherche ;
- Dans le troisième chapitre nous présentons notre approche pour concevoir notre système de prédiction de l'attrition.
- Dans le quatrième et dernier chapitre nous présentons l'implémentation de notre solution ainsi que les résultats expérimentaux obtenus sur notre base de données en nous basant sur les critères d'évaluation que nous définirons

# *Chapitre I*

## *Apprentissage automatique*

## I.1. Introduction

Dans sa nature, l'Homme a toujours tendance à vouloir apprendre de ses expériences passées, de ses échecs et de ses réussites, afin de pouvoir contrôler et prédire au mieux possible sa vie future. Ainsi, chacune de nos expériences est représentée par ses caractéristiques et son contexte, qui sont emmagasinées dans notre mémoire au fur et à mesure. Ces informations sont alors utilisées pour nous aider à prendre une décision lorsque l'on rencontre une nouvelle situation. Dans un contexte scientifique, cet apprentissage humain est plutôt connu sous le nom de l'apprentissage automatique (Machine Learning en anglais).

L'apprentissage automatique s'appuie sur des observations ou sur des expériences pour produire une procédure qui permettra de classer des individus, des objets et à prédire de nouvelles entrées.

Après avoir donné quelques définitions, nous présentons dans ce chapitre l'apprentissage supervisé et nous citons quelques algorithmes les plus populaires pour la classification supervisée. Ensuite, nous élaborons un tableau qui montre les différents avantages et inconvénients des algorithmes cités.

## I.2. Apprentissage automatique

L'apprentissage automatique est un domaine de recherche à la croisée de l'intelligence artificielle et de la statistique. Ce domaine a émergé dans les années 1960 en ayant pour objectif de fournir à la machine un ensemble d'algorithmes lui permettant d'apprendre à partir de données de façon automatique.

Dans les années 1990, grâce aux statistiques, l'apprentissage automatique est devenu très célèbre. L'intersection de l'informatique et de la statistique a donné naissance à des approches probabilistes de l'IA. Cela a déplacé le champ vers des approches basées sur des données. Disposant de données à grande échelle, les scientifiques ont commencé à mettre en place des

systèmes intelligents capables d'analyser de grandes quantités de données et d'en tirer des enseignements.

Les progrès de l'IA se sont poursuivis jusqu'aux années 2000, plusieurs contributions scientifiques à l'IA firent, telles que le concept du deep learning.

### I.1.1. Définition

(En anglais machine learning, littéralement « apprentissage machine » ou apprentissage statistique).

«L'apprentissage automatique est la science qui consiste à faire en sorte que les ordinateurs agissent sans être explicitement programmés.» - Stanford [1]

### I.1.2. Types d'apprentissage automatique

Pour commencer, un algorithme d'apprentissage est représenté par une fonction qui prend en entrée un ensemble d'apprentissage ou d'entraînement  $D$  et fournit un classifieur  $h$ . L'objectif de l'algorithme  $A$  est de généraliser l'information contenue dans  $D$  afin de produire une fonction prédictive.

En fonction du type de problème que l'on se pose, on peut mettre en place plusieurs types d'apprentissage.

On distingue deux grands types d'apprentissage :

- **Apprentissage supervisé**

L'objectif de cette approche est de concevoir un modèle reliant des données d'apprentissage à un ensemble de valeurs de sortie (un comportement) [3]. Par exemple on montre plusieurs images de visage et de non-visage et l'algorithme apprendra et sera capable de prédire si une image est un visage ou non.

L'apprentissage supervisé nécessite une base d'apprentissage où chaque instance est préalablement étiquetée avec sa classe respective. Supposons que nous disposons d'un échantillon de données formulés de  $N$  variables, nous aurons alors deux types d'informations :

- a. Un vecteur de valeurs  $X = (x_1, x_2, \dots, x_N)$  prises par chaque variable ;
- b. Et une valeur de sortie  $Y$  appelée valeur supervisée (qui peut être une classe pour le cas d'une classification).

Nous pouvons représenter le problème décrit précédemment par un ensemble d'entrée-sortie  $(X_i, Y_i)$ , avec  $i = 1..n$ ,  $n$  le nombre d'observations disponibles.

On appelle alors fonction d'apprentissage la fonction notée  $l: X \rightarrow Y$  qui associe un résultat supervisé à toute nouvelle entrée qui pourrait lui être présentée.

- **Apprentissage non-supervisé**

Il vise à concevoir un modèle structurant l'information. La différence ici est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver. [3]

Comprenons cela par un exemple. Supposons qu'un chef d'un magasin de location souhaiterait comprendre les préférences de ses clients pour développer son activité. Il pourra

regrouper tous ses clients en 10 groupes en fonction de leurs habitudes d'achat et utiliser une stratégie distincte pour les clients de chacun de ces 10 groupes.

### I.1.3. Objectifs des méthodes d'apprentissage

- **La classification**

La classification consiste à inférer, à partir d'un échantillon d'objets classés, une procédure de classification. Donc les classes sont connues et l'on dispose d'exemple de chaque classe [3], par exemple, classer si un email est un spam ou non ou classer une image de scanographie comme étant maligne ou bénigne.

Plus formellement, soit  $X \subset \mathbb{R}^d$  un ensemble de données représentant un espace à  $d$  dimensions. La donnée  $x \in X$  est appelé une instance et représente un point dans l'espace  $X$ . L'instance  $x$  est présentée sous forme d'un vecteur de taille  $d, x = (x(1), \dots, x(d))$ , où chaque chaque composante  $x(i) \in \mathbb{R}$  est une valeur discrète ou continue. Soit  $Y$  un ensemble fini de classes où chaque classe  $y \in Y$  est présentée sous forme de valeur discrète appelée étiquette ou cible de classe. Le classifieur se présente alors sous forme d'une fonction de classification  $h$  (appelé aussi modèle de classification) permettant d'associer une donnée  $x \in X$  à une étiquette de classe  $y \in Y$ .

(Équation I.1)

$$h \begin{cases} X \rightarrow Y \\ x \mapsto y = h(x) \end{cases}$$

Notez que pour le problème de classification, l'espace des réponses  $Y \subset \mathbb{N}$  est discret et fini, vu que chaque  $y \in Y$  représente une classe. Lorsque  $Y$  est continu (c.à.d.  $Y \subset \mathbb{R}$ ), on parle alors du problème de régression, qui sert à estimer la relation entre une à plusieurs variables  $x(i) \in \mathbb{R}$  et une autre variable  $y \in \mathbb{R}$ . Dans le cadre de ce mémoire, nous nous intéressons au problème de classification qu'au problème de régression.

Pour mesurer la qualité d'un classifieur  $h$ , on considère généralement une fonction de coût que l'on cherche à minimiser.

Exemples de fonction de coût :

- coût 0/1 : vaut 0 lorsque les étiquettes prévues et observées coïncident (c.à.d. lorsque la classe de  $x$  est correctement prédite), 1 sinon. Ce type de coût est utilisé pour la classification.
- Erreur quadratique : (équation I.2)  $(h(x) - y)^2$ . Elle est utilisée particulièrement en régression.
- **La régression**

Similaire à la classification, sauf que la classification se rapporte à des événements discrets. L'estimation porte sur des variables continues. Par exemple en prévoyant les prix d'un logement en fonction de l'âge de la maison, du nombre de pièces, de l'emplacement du quartier, etc.

- **La segmentation**

Consiste à former des sous-groupes (clusters) relativement homogènes à l'intérieur d'une population hétérogène. Dans ce cas, les classes ne sont pas prédéfinies et pour cette tâche, il n'y a pas de classe à expliquer, il appartient donc à un expert du domaine de déterminer l'intérêt et la signification des sous-groupes ainsi constitués.

#### I.1.4. Types de données

Par définition, selon Stanford, le machine learning est une technologie de l'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement [1]. Pour apprendre et se développer, les ordinateurs ont besoin de données à analyser et sur lesquelles s'entraîner. Les données, donc, sont très importantes pour l'apprentissage.

Il existe différents types de données, on cite :

- **Données structurées**

Elles résident généralement dans des bases de données relationnelles (SGBDR) et sont constituées de champs avec des types de données prédéfinis. Le langage SQL (Structured Query Language) permet d'effectuer des requêtes sur ce type de données structurées dans des bases de données relationnelles. Les données structurées sont facilement analysées d'ailleurs, la plupart des algorithmes d'apprentissage automatique fonctionnent avec des données structurées.

- **Données semi-structurées**

Elles conservent un type de données contenant des balises sémantiques qui identifient des éléments de données distincts, ce qui permet le regroupement d'informations et les hiérarchies. Le courrier électronique est un exemple très courant de type de données semi-structuré ainsi que XML et d'autres langages de balisage.

- **Données non-structurées**

Elles sont composées de types de fichiers, tels que des fichiers texte, audio, vidéo, et des publications sur les réseaux sociaux, qui peuvent être plus difficiles à analyser et à exploiter.

Ces données sont également stockées dans une base de données non relationnelle telle que NoSQL (Not only SQL).

Les données non structurées sont généralement pré-traitées pour en extraire des données structurées, avant que le modèle ne puisse être construit.

## I.3. Algorithmes d'apprentissage automatique

### I.3.1. Algorithmes d'apprentissage supervisé

Parmi les méthodes de classification supervisées les plus populaires, on peut citer :

#### I.3.1.1. k-plus proche voisins

L'algorithme des plus proches voisins (K-PPV ou K-NN pour Nearest Neighbor) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, et le nombre de voisins à prendre en compte soit  $k$ .

Afin de trouver les  $k$  plus proches d'une donnée à classer, on peut choisir plusieurs mesures de distance, la plus souvent utilisée est la distance Euclidienne, cependant, en fonction du problème, on peut encore utiliser les distances de Manhattan, de Mahalanobis, de Hamming, etc.

#### - *Quelques mesures de distance utilisées*

Soient 2 points  $X$  et  $Y$  de coordonnées respectives  $(x_1, \dots, x_n)$ ,  $(y_1, \dots, y_n)$  dans un espace de paramètres  $\mathbb{R}^n$ .

✚ Distance Euclidienne :

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

✚ Distance Manhattan :

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

✚ Distance de Hamming :

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

$$x = y \Rightarrow d(x, y) = 0$$

$$x \neq y \Rightarrow d(x, y) = 1$$

- **Choix de k**

L'algorithme nécessite de connaître la valeur du paramètre k, le nombre de voisins à considérer. Une méthode classique pour avoir cette valeur est la validation croisée (cross validation en anglais). Cette méthode consiste à diviser l'ensemble des données D en k sous-ensembles, utiliser k-1 sous-ensembles de D comme données d'apprentissage et un sous-ensemble comme données de validation. Par la suite, calculer un score de performance, puis on répète l'opération en sélectionnant un autre sous-ensemble de validation parmi les k-1 qui n'ont pas encore été utilisés pour la validation du modèle. L'opération se répète k fois, pour qu'en fin de compte chaque sous-ensemble ait été utilisé exactement une fois comme ensemble de validation.

- **Algorithme**

---

**Algorithme des k-PPV**

---

1. Initialisation, choix de :
    - Nombre de classes, valeur de k, exemples initiaux, mesure de similarité ;
  2. Pour chaque vecteur d'objet à classer :
    - Mesurer la distance du vecteur avec tous les autres déjà classés ;
    - Déterminer la liste des k vecteurs les plus proches de lui (k-ppv) ;
    - Déterminer la classe la plus représentée dans la liste des k-ppv et affecter notre vecteur à cette classe.
- 

- **Exemple**

Considérons l'ensemble de données suivant ayant trois variables : la taille, le poids et la taille du t-shirt de certains clients. Et que nous ayons besoin de prédire la taille du t-shirt d'un nouveau client en fonction des informations de taille et de poids dont nous disposons.

| <b>HAUTEUR EN<br/>CM)</b> | <b>POIDS (EN KG)</b> | <b>TAILLE DU T-<br/>SHIRT</b> |
|---------------------------|----------------------|-------------------------------|
| 158                       | 58                   | M                             |
| 158                       | 59                   | M                             |
| 158                       | 63                   | M                             |
| 160                       | 59                   | M                             |
| 160                       | 60                   | M                             |
| 168                       | 62                   | L                             |
| 160                       | 64                   | L                             |
| 163                       | 64                   | L                             |
| 165                       | 61                   | L                             |

Tableau I.1 : Exemple de prédiction de la taille d'un tee-shirt.

Sur la base de données fournie, nous devons prédire la taille du t-shirt du client suivant :

| <b>HAUTEUR (EN CM)</b> | <b>POIDS (EN KG)</b> | <b>TAILLE DU T-SHIRT</b> |
|------------------------|----------------------|--------------------------|
| 161                    | 61                   | ?                        |

Tableau I.2 : Exemple à prédire.

Pour trouver les plus proches voisins, nous allons calculer la distance euclidienne :

| <b>HAUTEUR EN CM)</b> | <b>POIDS (EN KG)</b> | <b>TAILLE DU T-SHIRT</b> | <b>DISTANCE</b> |
|-----------------------|----------------------|--------------------------|-----------------|
| 158                   | 58                   | M                        | 4.2             |
| 158                   | 59                   | M                        | 3.6             |
| 158                   | 63                   | M                        | 3.6             |
| 160                   | 59                   | M                        | 2.2             |
| 160                   | 60                   | M                        | 1.4             |
| 168                   | 62                   | L                        | 7.1             |
| 160                   | 64                   | L                        | 3.2             |
| 163                   | 64                   | L                        | 3.6             |
| 165                   | 61                   | L                        | 4               |

Tableau I.3 : calcul de distance pour chaque exemple.

Calculons maintenant le plus proche voisin pour  $k=3$  :

| HAUTEUR EN CM) | POIDS (EN KG) | TAILLE DU T-SHIRT | Distance   |
|----------------|---------------|-------------------|------------|
| 158            | 58            | M                 | 4.2        |
| 158            | 59            | M                 | 3.6        |
| 158            | 63            | M                 | 3.6        |
| 160            | 59            | M                 | <b>2.2</b> |
| 160            | 60            | M                 | <b>1.4</b> |
| 168            | 62            | L                 | 7.1        |
| 160            | 64            | L                 | <b>3.2</b> |
| 163            | 64            | L                 | 3.6        |
| 165            | 61            | L                 | 4          |

Tableau I.4 : tableau illustrant les 3-plus proches voisins.

2/3 des voisins se dirigent vers la taille « M ». Par conséquent, selon l'algorithme k-nn la classe de (161,61) est la taille « M ».

### I.3.1.2. Classifieur Naïf Bayésien

Un classifieur probabiliste simple basé sur le théorème de Bayes qui suppose que les descripteurs qui décrivent les objets de l'ensemble d'apprentissage sont indépendants (d'où le nom naïf).

Théorème de Bayes est donné par : (équation I.3)  $P(x|y) = \frac{P(x|y)P(y)}{P(x)}$ , avec  $P(x|y)$  est la probabilité conditionnelle d'un événement  $x$  sachant qu'un autre événement  $y$  de probabilité non nulle s'est réalisé.

#### - Formulation

Soient  $X = (x_1, \dots, x_n)$  l'ensemble des descripteurs,  $Y$  la variable à prédire (l'attribut classe comportant  $k$  modalités). En apprentissage supervisé, pour un nouvel individu  $w$  à classer le classifieur bayésien va choisir la classe «  $y_k$  » qui a la plus grande probabilité, on parle de règle MAP (maximum a posteriori) c.-à-d.

(Équation I.4)

$$y = \operatorname{argmax}_k P[Y = y_k | X]$$

Cette dernière formule peut s'écrire d'une manière différente, d'après la règle de Bayes.

(Équation I.5)

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1)P(x_2)\dots P(x_n)}$$

Comme l'objectif est de détecter le maximum de cette quantité selon  $y_k$  et que le dénominateur n'en dépend pas, alors détecter le maximum de cette quantité selon  $y_k$  revient à détecter le maximum de (équation I.6)  $P(y|x_1, \dots, x_n) = P(y) \prod_{i=1}^n P(x_i|y)$ . D'où la formule suivante :

(Équation I.7)

$$y = \underset{k}{\operatorname{argmax}} (P(y) \prod_{i=1}^n P(x_i|y)).$$

- *Exemple*

Nous disposons des 10 exemples de courriels bancaires suivants afin de prédire si un mail est un spam ou pas.

| <b>gagner</b> | <b>million</b> | <b>compte</b> | <b>code</b> | <b>Classe</b> |
|---------------|----------------|---------------|-------------|---------------|
| 1             | 1              | 0             | 0           | Spam          |
| 0             | 0              | 1             | 1           | Spam          |
| 0             | 1              | 1             | 0           | Non spam      |
| 1             | 1              | 0             | 0           | Spam          |
| 0             | 0              | 0             | 0           | Non spam      |
| 1             | 0              | 0             | 0           | Spam          |
| 1             | 0              | 0             | 0           | Non spam      |
| 0             | 0              | 0             | 1           | Spam          |
| 1             | 0              | 1             | 1           | Spam          |
| 0             | 1              | 1             | 1           | Non spam      |

Tableau I.5 : exemples de détection de pourriels.

Nous souhaitons prédire la classe des messages suivants : « gagner million », « million compte » et « code compte ».

Déterminons les probabilités à priori de chaque classe :

| <b>P (spam)</b> | <b>P (¬spam)</b> |
|-----------------|------------------|
| 6/10            | 4/10             |

Tableau I.6 : Probabilité à priori des classes.

Calculons les probabilités de chaque variable sachant la classe :

| Variables $x_i$ | $P(x_i \text{spam})$ | $P(x_i \neg\text{spam})$ |
|-----------------|----------------------|--------------------------|
| gagner          | 4/6                  | 1/4                      |
| million         | 2/6                  | 2/4                      |
| compte          | 2/6                  | 2/4                      |
| code            | 3/6                  | 1/4                      |

Tableau I.7 : Probabilité de chaque variable  $x_i$  sachant  $y$ .

Appliquons la règle de Bayes : (équation I.6).

Par conséquent, nous devons trouver la classe  $y$  avec la probabilité maximale (Équation I.7).

Nous obtenons les résultats suivants :

| Message        | Score si le message est un spam | Score si le message est non un spam | prédiction |
|----------------|---------------------------------|-------------------------------------|------------|
| gagner million | 0.133                           | 0.05                                | Spam       |
| compte million | 0.067                           | 0.1                                 | Non spam   |
| code compte    | 0.1                             | 0.05                                | spam       |

Tableau I.8 : Résultats de prédiction avec le naïf bayésien.

### I.3.1.3. Arbre de décision

#### - Définition

Un arbre de décision est une structure qui est souvent utilisée pour représenter des connaissances. Il permet de remplacer ou d'assister un expert humain dans la détermination des propriétés d'un objet, c'est l'opération de classification. Un arbre de décision est une représentation d'une procédure de décision pour déterminer la classe d'un objet donné.

En général à chaque nœud interne de l'arbre, il y a un test (question) qui correspond à un attribut dans la base d'apprentissage, et une branche correspondant à chacune des valeurs possibles de l'attribut. A chaque nœud de feuille, il y a une valeur de classe. Les arbres de décision sont construits à partir d'un ensemble d'apprentissage. Un chemin de la racine à un nœud correspond à une série d'attributs (questions) avec leurs valeurs (réponses). [4]

Classer un nouvel individu se fait par une descente dans l'arbre, de la racine vers une des feuilles (qui encode la décision ou la classe). A chaque niveau de la descente on passe un nœud intermédiaire où une variable est testée pour décider du chemin (ou sous arbre) à choisir pour continuer la descente.

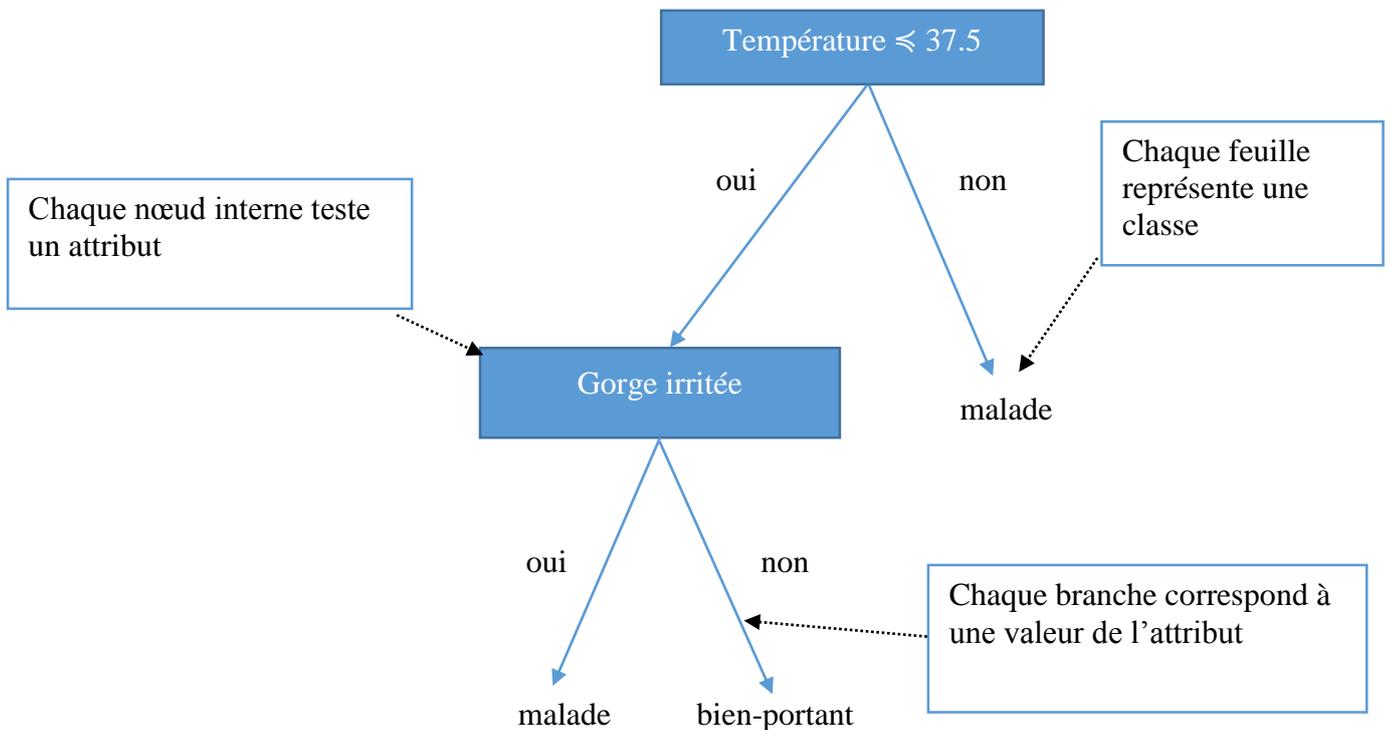


Figure I.1 : Exemple d'un arbre de décision

- **Principe de construction d'un arbre de décision**

Au départ, toutes les instances d'apprentissage sont à la racine de l'arbre. Ensuite, on sélectionne un attribut et on choisit un test de séparation sur un attribut, qui sépare le mieux les

instances. La sélection des attributs est basée sur une heuristique ou une mesure statistique. Enfin, partitionner les instances entre les nœuds fils suivant la satisfaction des tests.

- **Sélection de l'attribut discriminant**

Le choix des attributs à tester est une étape cruciale pour la construction de l'arbre car elle permet de choisir l'attribut qui permettra de répartir au mieux le jeu d'apprentissage. Pour ce faire, une mesure statistique est souvent utilisée. L'objectif principal est de construire des arbres de décision relativement simples. On cherche un arbre petit et simple plutôt qu'un arbre grand qui est complexe difficile à interpréter par un non-expert.

Dans la pratique, les deux mesures les plus connues sont : l'entropie et l'indice de Gini.

Pour une partition de l'ensemble A en C<sub>1</sub>, C<sub>2</sub>,..., C<sub>n</sub> parties on a :

- Entropie de l'information (de Shannon) :

(Équation 1.8)

$$E(A) = - \sum_{i=1}^n \frac{|C_i|}{|A|} * \left( \log_2 \left( \frac{|C_i|}{|A|} \right) \right)$$

- Indice de Gini :

(Équation 1.9)

$$I(A) = - \sum_{i=1}^n \frac{|C_i|}{|A|} * \left( 1 - \frac{|C_i|}{|A|} \right)$$

- ✓ |C<sub>i</sub>| représente le cardinal de la classe C<sub>i</sub>
- ✓ |A| représente le cardinal de l'ensemble A

- **Algorithme générique [5]**

---

### Algorithme d'apprentissage d'arbre de décision

---

**Données** : Echantillon A avec m enregistrements étiquetés ;

**Initialisations** : Arbre : vide ; nœud courant : racine ; échantillon courant : A ;

**Répéter**

✓ Décider si le nœud courant est terminal ;

✓ **Si** le nœud courant est terminal **alors**

Etiqueter le nœud courant par une feuille

**Sinon**

Sélectionner un test et créer le sous arbre ;

**FinSi** ;

Nœud courant : un nœud non encore étudié ;

Echantillon courant : échantillon atteignant le nœud courant ;

✓ Passer au nœud suivant non exploré s'il en existe ;

**Jusqu'à** obtenir un arbre de décision h ;

**Sortie** : arbre de décision h.

---

#### I.3.1.4. Forêts aléatoires (Random Forest)

Les forêts aléatoires sont introduites par Breiman en 2001, basés sur le principe de l'algorithme expliqué précédemment. L'idée derrière leur conception été d'améliorer la précision des modèles obtenus, les forêts aléatoires combinent plusieurs modèles pour obtenir une meilleure performance. Dans ce contexte deux méthodes sont particulièrement utilisées : le bagging et le boosting.

**- Le bagging**

Le « bagging » est un algorithme qui se base sur le « Bootstrap » qui consiste à créer de nouveaux échantillons par tirage au hasard dans l'ancien échantillon, avec remise.

L'idée de base du « bagging » est d'entraîner un algorithme d'apprentissage sur plusieurs bases d'apprentissage obtenues par tirage avec remise. Alors, à partir de chaque échantillon un

classifieur est construit, le résultat de la classification est ensuite obtenu par rapport au vote majoritaire.

- *Le boosting*

Le boosting consiste à appliquer le même algorithme de classification, de façon successive à des versions de l'échantillon initial d'apprentissage qui sont modifiés à chaque étape pour tenir compte des erreurs de classification de l'étape précédente, puis à combiner les classifieurs (éventuellement faibles) ainsi construits pour obtenir un classifieur plus fort.

- *Algorithme*

---

**Algorithme d'apprentissage des forêts aléatoires**

---

**Entrées :**

- D l'échantillon d'apprentissage ;
- T le nombre d'arbres ;

**Pour  $i = 1 \dots T$  faire**

1. Tirer un échantillon bootstrap dans D ;
2. Pour chaque échantillon  $i$  : Tirer au hasard  $q$  attributs parmi  $p$  existants et construire un arbre de décision  $G_i(x)$  sur ces attributs ;

**Sortie :** Classification : agrégation par vote  $G(x) = \text{vote majoritaire}(G_1(x), \dots, G_T(x))$ .

---

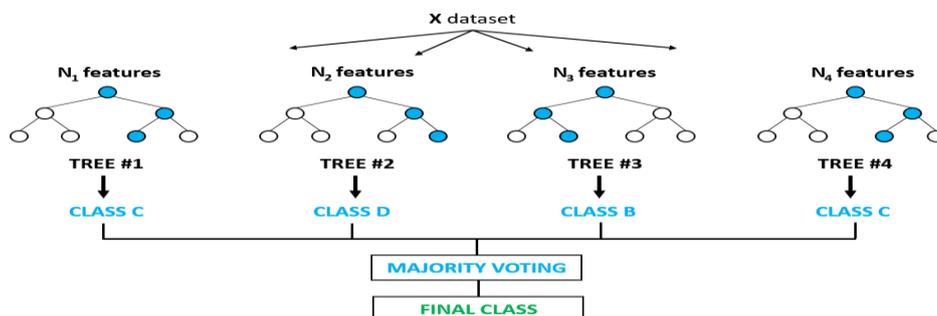


Figure I.2 : Illustration du fonctionnement d'une forêt aléatoire.

### I.3.1.5. Régression logistique binaire

La régression logistique binaire est une technique prédictive qui vise à construire un modèle permettant de prédire les valeurs prises par une variable cible qualitative à partir d'un ensemble de variables explicatives quantitatives ou qualitatives.

Cette méthode a été utilisée dans plusieurs domaines de recherche tels que la biologie humaine, la sociologie, la finance ..., dans ce type de modèle le but est de décrire la relation entre une variable expliquée ou réponse  $Y$  dichotomique codée traditionnellement en 0 pour absence du phénomène et 1 pour présence du phénomène que nous cherchons à expliquer à travers  $p$  variables explicatives  $X = (X_1, \dots, X_p) \in \mathbb{R}^p$ .

#### - *Formulation*

La régression logistique nous permet donc d'exprimer la probabilité que l'événement se réalise en fonction des variables explicatives, à l'aide de la fonction Logit qui est la suivante :

(Équation I.10)

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

La formulation mathématique de la régression logistique est la suivante :

(Équation I.11)

$$\text{logit}(p(Y = 1|X)) = \ln\left(\frac{p(Y = 1|X)}{1-p(Y = 1|X)}\right) = \beta_0 + \beta_1 x_1 + \dots + \beta_j x_j$$

Avec  $X$  représentant les variables explicatives et les  $\beta_i$  étant les coefficients de la régression à estimer. Les coefficients  $\beta_i$  permettent de mesurer l'influence de chaque variable et ainsi de déterminer les plus discriminantes.

### I.3.1.6. SVM (Support Vector Machine)

#### - *Principe*

La technique des séparateurs à vastes marges est utilisée pour résoudre des problèmes de discrimination, c.à.d. décider à quelle classe appartient un échantillon de données ou de régression, prédire la valeur numérique d'une variable.

Les SVMs ont été développés pour but de séparer les données en deux classes à l'aide d'une frontière aussi simple que possible, de telle façon que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge », « les vecteurs de support » étant les données les plus proches de la frontière.

L'idée principale des SVMs est de reconsidérer le problème dans un espace de dimension supérieure, éventuellement de dimension infinie. Dans ce nouvel espace, il est alors probable qu'il existe un hyperplan séparateur linéaire. Si c'est le cas, les SVMs cherche parmi l'infinité des hyperplans séparateurs celui qui maximise la marge entre les classes.

Les SVMs font appel à une fonction implicite qui transforme les entrées  $X$  en un vecteur dans un espace de Hilbert  $F$ . Dans le cas d'un classement en 2 classes, on détermine un hyperplan dans cet espace  $F$ . la solution optimale repose sur la propriété que les objets sont les plus éloignés possibles de l'hyperplan, on maximise ainsi les marges.

Soit  $x$  le vecteur associé. On définit  $f(x) = \omega x + \beta$  et l'hyperplan a pour équation  $\omega x + \beta = 0$ .

La distance d'un point au plan est donnée par  $d(x) = |\omega x + \beta|/|\omega|$ .

En résumé, pour savoir si un point  $x$  se trouve d'un côté ou de l'autre de l'hyperplan. La fonction  $f$  nous permet nous permet de répondre à cette question, grâce à la classification suivante :

$$\begin{cases} f(x) \geq 0 \Rightarrow x \in \text{classe1} \\ f(x) > 0 \Rightarrow x \in \text{classe2} \end{cases}$$

- **Exemple :**

Supposons que nous disposons des exemples suivants :

Classe «-1» :  $\{(1,1), (2,0)\}$  et classe «1» :  $\{(2,3)\}$

L'objectif est de tracer un hyperplan qui classe tous les vecteurs d'apprentissage en deux classes  $\{-1,1\}$ .

Un hyperplan  $f(x)$  donne une fonction discriminante linéaire qui divise l'espace d'origine en deux demi-espaces :  $f(x) = \varpi x + \beta$  où  $\varpi$  est le vecteur de poids et  $\beta$  est le biais.

Le vecteur de poids est  $\vec{\varpi} = (2,3) - (1,1) = (\alpha, 2\alpha)$ .

Nous avons  $\begin{cases} f(1,1) = -1 \Leftrightarrow \alpha + 2\alpha + \beta = -1 \text{ (équation I.12)} \\ f(2,3) = +1 \Leftrightarrow 2\alpha + 6\alpha + \beta = +1 \text{ (équation I.13)} \end{cases}$

De l'équation I.13 on aura :  $\beta = 1 - 8\alpha$  (équation I.14)

En remplaçant  $\beta$  dans l'équation I.12 :  $\alpha = \frac{2}{5}$

On remplace la valeur de  $\alpha$  dans l'équation I.14 :  $\beta = -\frac{11}{5}$

Donc  $\vec{\beta} = (\frac{2}{5}, \frac{4}{5})$

Alors, la fonction de l'hyperplan qui divise le mieux l'échantillon est :

(Équation I.15)

$$f(x) = \frac{2}{5}x_1 + \frac{4}{5}x_2 - \frac{11}{5} \Leftrightarrow f(x) = x_1 + 2x_2 - 5$$

Maintenant, pour décider à quelle classe appartient l'exemple suivant (4,5). Il suffit de remplacer les variables  $x_1$  et  $x_2$  par les valeurs 4 et 5 respectivement dans l'équation I.15 :  $f(4,5) = 8,5 > 1$  du coup on va le classer dans la classe +1.

### I.3.1.7. Réseau de neurone

Les réseaux de neurones artificiels (Artificial Neural Networks en anglais) sont des modèles de calculs dont la conception initiale est très schématiquement inspirée du fonctionnement des neurones biologiques.

Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes entre un grand nombre de variables, sans intervention externe. Les tâches particulièrement adaptées au traitement par réseau de neurones sont : l'association, la classification, l'estimation, et la commande de processus complexes.

Les réseaux de neurones sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone. L'architecture générale d'un réseau de neurones est subdivisée en plusieurs niveaux, ainsi, on retrouve trois différentes couches: une couche d'entrée, une couche de sortie, et entre ces deux une couche cachée. Ces couches peuvent être composées de un ou plusieurs neurones. La figure suivante représente l'architecture générale d'un réseau de neurones :

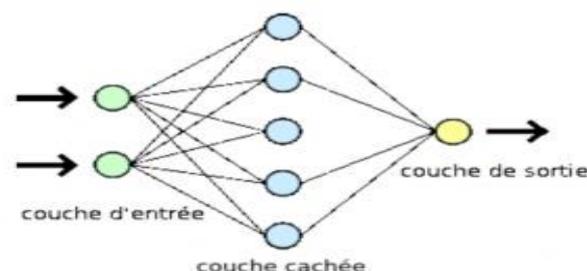


Figure I.3 : Architecture générale d'un réseau de neurone.

Les notions générales à connaître pour comprendre le fonctionnement des réseaux de neurones sont les suivantes :

- *Le perceptron*

Le premier modèle de réseau de neurones complet a été nommé Perceptron. Ce modèle est un réseau mono-couche et n'a qu'une seule sortie à laquelle toutes les entrées sont connectées. Une fonction de combinaison associe les entrées du neurone (par exemple en les sommant), puis compare ce résultat à une valeur seuil, via une fonction d'activation.

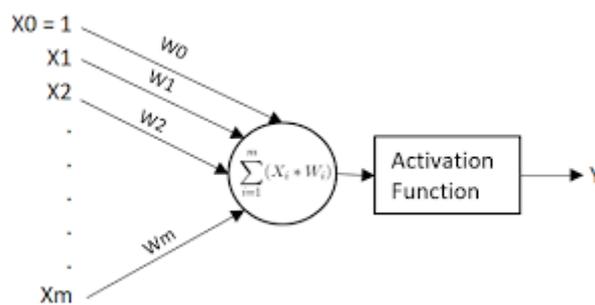


Figure I.4 : Schéma d'un perceptron.

Les Perceptrons MultiCouches (PMC) constituent une amélioration du perceptron comprenant une ou plusieurs couches intermédiaires dites cachées, afin d'augmenter les combinaisons possibles. Les neurones entre les couches cachées sont reliés par des liaisons pondérées.

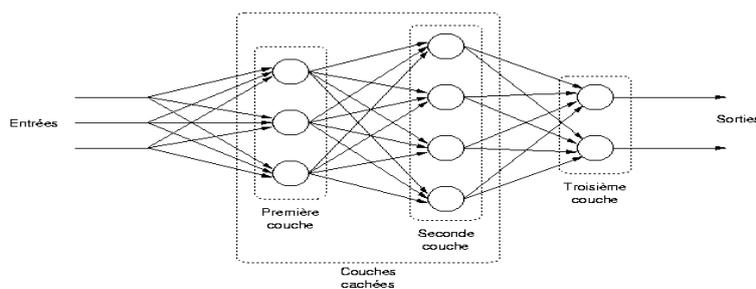


Figure I.5 : Schéma du perceptron multi couches.

- *Les entrées / sorties*

Les entrées sont les valeurs que reçoit le réseau en entrée, il peut s'agir de valeurs numériques, alphabétiques, alphanumériques, des images, des signaux et autres. Par exemple dans le domaine médical ce sera les symptômes du malade, dans notre domaine il s'agit des différentes variables de notre base de données contenant des informations historiques des clients. Les valeurs sortantes de cette couche iront directement dans la couche cachée.

Les sorties sont les valeurs que retourne le réseau de neurones, tout comme les entrées, elles peuvent être de différents types. Par exemple dans le domaine médical ce sera la maladie et pourquoi pas aussi le traitement à prescrire au malade, dans notre domaine il s'agira de valeurs binaires 0 ou 1 (0 pour dire que le client est dans la classe non churner et 1 pour dire que le client est classé churner). Les valeurs retournées par la couche cachée seront prises par la couche de sortie.

- *Les couches cachées*

C'est l'ensemble de neurones séparant la couche d'entrée et celle de sortie. C'est à ce niveau que ce fait le passage de l'information acquise par la couche d'entrée pour faire un apprentissage dessus, et retourner les résultats à la couche de sortie. Cet apprentissage est effectué en faisant une rétropropagation à chaque fois que le résultat ne correspond pas au résultat souhaité, ceci en modifiant les différents poids à chaque itération.

- *Les poids et la fonction de transfert*

Les poids représentent des valeurs réelles comprises entre 0 et 1, permettant d'affecter un degré d'importance à une donnée en entrée. Ces poids sont initialisés avec des valeurs aléatoires. Ils sont modifiés à chaque itération et propagés dans le réseau.

Et la fonction de transfert est le modèle mathématiques qui relie les entrées et les sorties du système. Dans cette étape les valeurs des données  $x_i$  sont associées à leurs poids respectifs  $w_i$  afin de calculer une nouvelle valeur qui sera passée à la couche suivante.

Il existe plusieurs fonction de transfert, on peut citer : les fonctions qui utilisent le logarithme, le cosinus, l'exponentiel ...etc, chacune varient selon le temps d'exécution, l'espace mémoire, ...etc.

- ***La rétropropagation (Backpropagation)***

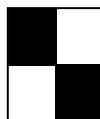
La rétropropagation permet de corriger et d'améliorer les erreurs de prédictions qui représentent la différence entre les résultats obtenus et les sorties désirées, avec un simple retour arrière qui permet de modifier les poids et ceci à chaque fois que la sortie ne correspond au résultat voulu. L'objectif principal de la rétropropagation est de minimiser le taux d'erreurs.

- ***Exemple***

Nous allons utiliser un perceptron simplifié. Notre perceptron sera un réseau de neurone comportant deux couches dont tous les neurones seront connectés à la couche suivante. Il s'agit de reconnaître un motif géométrique sur un élément de quatre pixels.

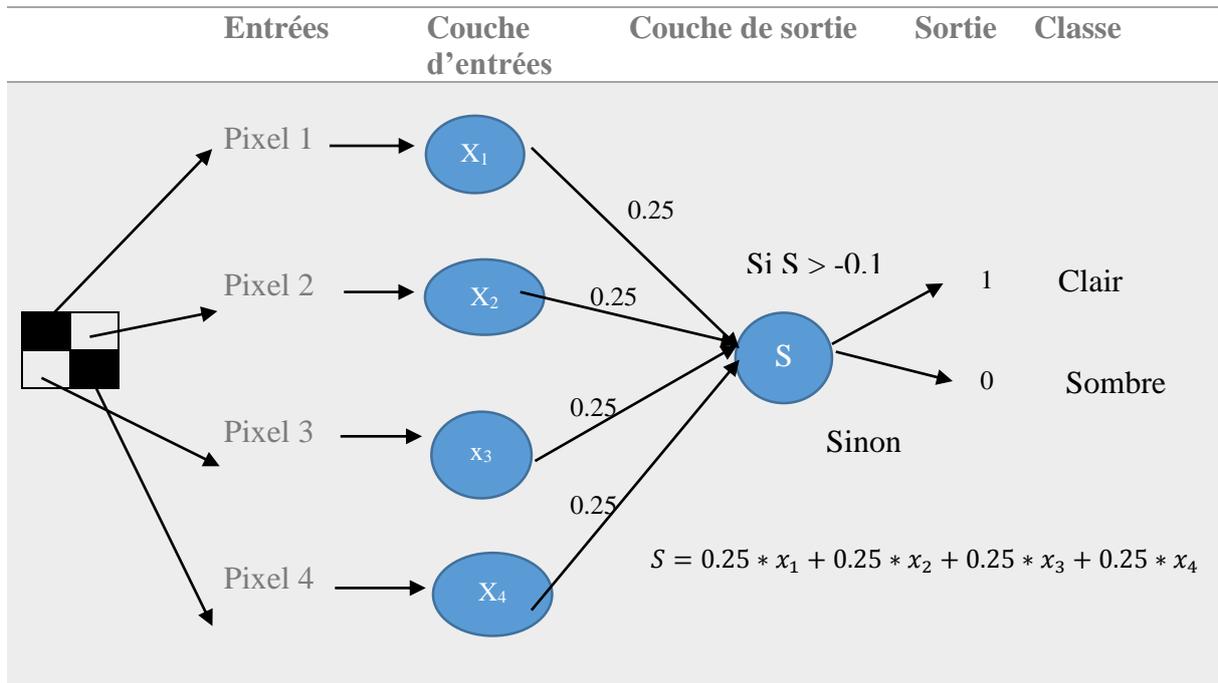
Nous obtiendrons 0 lorsque le pixel sera clair et 1 lorsqu'il sera sombre.

Si l'image contient deux, trois ou quatre pixels blancs sera classée comme « claire ». Si elle contient 0 ou 1 pixel blanc, elle sera classée comme « sombre ».



Le motif géométrique à reconnaître « 1001 ».

Le réseau va se charger de classer une image de 2\*2 pixels soient donc quatre entrées. Il y a en tout seize motifs possibles soient donc un échantillon de seize exemples.



Pour prédire le motif (1001) :

$$S = 0.25 * 1 + 0.25 * 0 + 0.25 * 0 + 0.25 * 1 = 0.5$$

Nous avons  $0.5 > -0.1$  donc le résultat du réseau est 1, l'image est classé comme « claire ».

### I.3.2. Algorithmes d'apprentissage non supervisé

L'objectif d'une méthode de classification non supervisée (ou Clustering) est la segmentation ou la répartition des individus dans des classes homogènes, ou catégories. Ceci est fait en optimisant un critère visant à regrouper les individus dans des classes les plus homogènes possibles, et les plus distinctes entre elles.

Il existe de nombreuses méthodes de classification non supervisée, seule une sélection est décrite en dessous. Cette sélection représente les méthodes les plus fréquemment utilisées.

Le clustering se catégorise en plusieurs familles de méthodes selon la stratégie suivie pour construire les clusters. Parmi ces méthodes on trouve :

### I.3.2.1. Le clustering hiérarchique

Les clusters formés dans cette méthode forment une structure de type arborescence basée sur la hiérarchie appelée dendrogramme. Il s'agit d'un arbre dans lequel la racine est le cluster unique qui regroupe tous les échantillons, les feuilles étant les individus, et les nœuds intermédiaires de l'arbre représentent les clusters.

Les méthodes permettant de construire un dendrogramme de ce type se décomposent en deux familles :

- *Les méthodes hiérarchiques ascendantes (ou agglomératives)*

Commence par une solution spécifique aux données pour arriver à une autre solution plus générale. Avec ces méthodes, chaque observation est initialement considérée comme un cluster distinct (feuille), c'est-à-dire on démarre avec autant de clusters que de données. Ensuite, les clusters les plus similaires sont fusionnés selon un critère donné jusqu'à ce qu'il reste qu'un seul grand cluster (racine) englobant tous les individus.

- **Algorithme HCA (Hierarchical Agglomerative Clustering) [5]**

---

#### Algorithme HCA

---

**Entrée:** une collection A de N objets;

**Sortie :** un dendrogramme

- Initialiser N clusters chacun réduit à un objet de A ( $\mathcal{C} := \{\mathcal{C}_1 \dots \mathcal{C}_N\}$ );
  - Initialiser la matrice de distance entre les pairs de cluster ;
  - **Tant que**  $|\mathcal{C}| > 1$  ou (condition d'arrêt) **faire**
-

- ✓ Récupérer la paire  $C_{n1}, C_{n2}$  de clusters vérifiant le critère d'agrégation ;
- ✓ Oter de C ces deux clusters ;
- ✓ Ajouter à C le cluster correspondant à leur fusion ;
- ✓ Mettre à jour la matrice de distance ;

**fait**

### ▪ Illustration

La classification hiérarchique commence par traiter chaque observation comme un cluster distinct. Ensuite il exécute à plusieurs reprises les deux étapes suivantes : (1) identifier les deux clusters les plus proches l'un de l'autre et (2) fusionner les deux clusters les plus similaires. Cela continue jusqu'à ce que tous les clusters soient fusionnés. Ceci est illustré dans la figure ci-dessous.

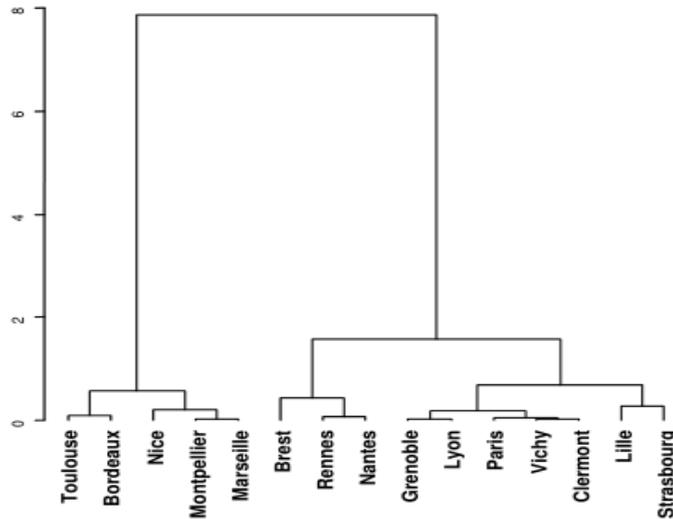


Figure I.6 : Exemple de clustering hiérarchique ascendant pour quelques villes de France qui ont des profils météo similaires.

- *Les méthodes hiérarchiques descendantes (ou divisives)*

À l'inverse des méthodes ascendantes, ces méthodes commencent par une solution générale pour arriver à une autre plus spécifique. Les méthodes descendantes démarrent d'un seul cluster contenant la totalité des données, puis, elles divisent à chaque étape les clusters selon un critère jusqu'à l'obtention de n clusters.

Il existe différentes méthodes pour mesurer la distance entre les clusters. On peut citer :

- *Centroid Method* : définit la distance entre les centres des clusters.
- *Single Link Method* : définit la distance minimale entre toutes les paires d'objets de 2 clusters.
- *Complete Link Method* : définit la distance maximale entre toutes les paires d'objets de 2 clusters.
- *Average Linkage* : définit la distance moyenne entre toutes les paires d'objets.

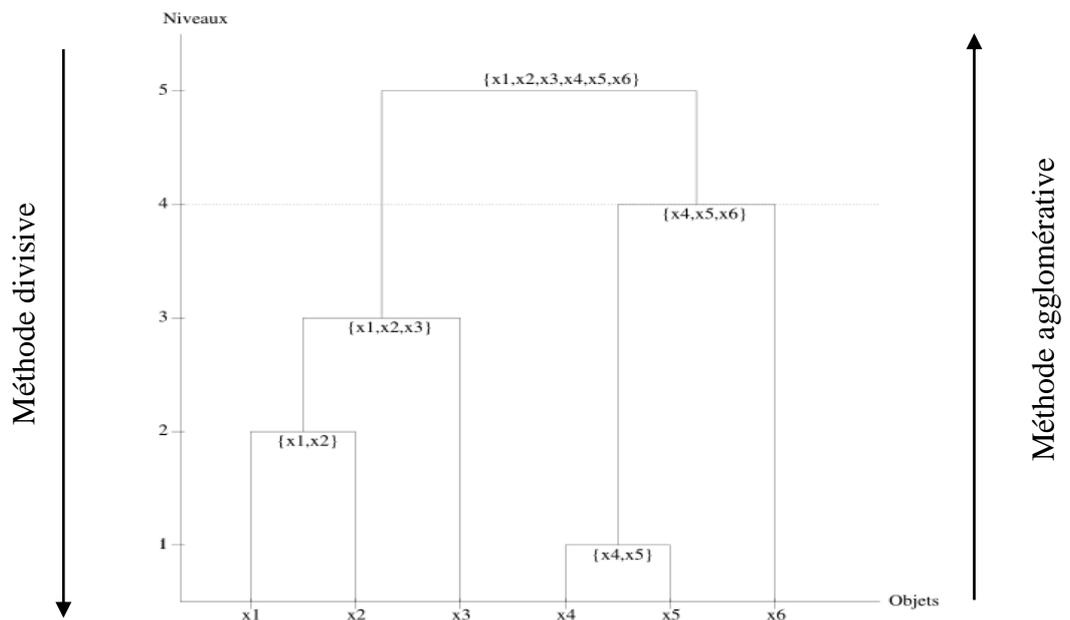


Figure I.7 : Exemple de dendrogramme

### I.3.2.2. Le clustering par partitionnement

Contrairement aux méthodes hiérarchiques précédentes, les algorithmes de partitionnement proposent, en sortie, une partition de l'espace des objets plutôt qu'une structure organisationnelle de type « dendrogramme ».

Ces méthodes consistent à diviser de manière optimale l'ensemble des instances en un groupe fini de clusters  $k$  étant fixé à priori. Les approches par partitionnement utilisent un processus itératif en fonction du nombre  $k$  qui consiste à affecter chaque individu à la classe la plus proche au sens d'une distance – ou d'un indice de similarité-. Il existe plusieurs algorithmes utilisant cette méthode comme K-Means, K-Médoids, CLARANS.

#### - *K-Means*

Cet algorithme fut longtemps utilisé sur les grands jeux de données en raison de sa rapidité. On a  $n$  objets à partitionner en  $k$  clusters. Cet algorithme construit les  $k$  clusters et les corrige jusqu'à obtenir une similarité satisfaisante, ou qu'il n'y ait plus de changement.

#### ▪ **Algorithme**

---

Algorithme K-Means

---

**Entrée** :  $k$  le nombre de clusters ;

**Sortie** : ensemble de  $k$  clusters ;

Choisir aléatoirement les centres de groupes et le nombre de clusters ;

#### **Répéter**

✓ affecter chaque objet  $O$  au cluster  $C_i$  dont il est le plus proche à son centre ;

✓ recalculer le centre de chaque cluster ;

**jusqu'à** stabilisation des centres

---

#### ▪ **Illustration**

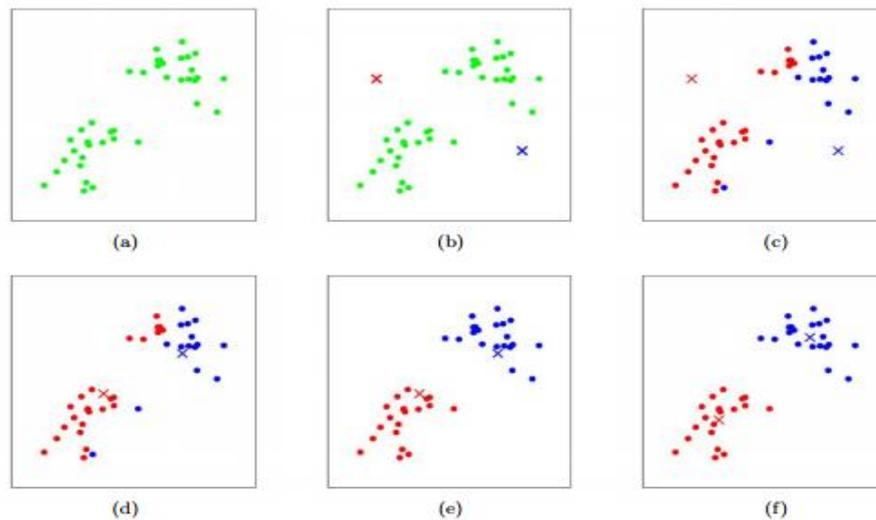


Figure I.8 : Algorithme K-means.

Dans la figure I.8, les points représentent les exemples de formation et les croix représentent les centroïdes des clusters. (a) jeu de données initial. (b) centroïdes de clusters initiaux aléatoires. (c), (d), (e) et (f) Illustration de l'exécution de deux itérations de k-means. Dans chaque itération, nous attribuons chaque exemple d'apprentissage au centroïde du cluster le plus, nous déplaçons ensuite chaque centre de cluster vers la moyenne des points qui lui sont attribués.

### I.3.2.3. Le clustering par densité

Consiste à identifier dans l'espace de description des objets les régions de forte densité, entourées par des régions de faible densité pour former les clusters.

## I.4. Quelques exemples d'application des algorithmes de l'apprentissage supervisé

- K-plus proches voisins

Concernant un espace géométrique :

- Probabilité de collision, par exemple dans le cas d'un jeu vidéo, on calcule la distance entre le personnage et les objets les plus proches pour savoir si le personnage peut être bloqué par un obstacle ;
- Interaction à distance : en physique, l'intensité des interactions diminue avec la distance, il est donc utile de déterminer les plus proches voisins d'une particule, par exemple, pour limiter les calculs d'interactions.

- Classifieur Naïf Bayes

Il est particulièrement utile pour les problématiques de classification de texte comme le filtre anti-spam et catégorisation de texte en thème.

- Arbre de décision

- Prédire le désabonnement des clients dans de nombreux secteurs ;
- Détection de fraude dans le secteur des assurances ;
- Diagnostic médical ;
- ...

- Forêts aléatoires

- Aide au diagnostic médical ;
- La prévision de la pollution ;
- Reconnaissance des positions et des mouvements ;
- Reconnaissance d'actions (exemple : boire, téléphoner, écrire,...)
- ...

- Régression logistique : très utilisée dans :

- le domaine médical : prédire la guérison d'un patient ;
- le marketing quantitatif : achat ou non de produits ou service suite à une action ;
- en finance pour modélisation de risque (scoring).

- SVM

- Classification d'images : reconnaissance de visages, de chiffres manuscrits ;

- Interprétation textuelle : détection de spam ;
- Réseaux de neurone
  - La détection des cellules cancéreuses ;
  - Classifications d'espèce animales selon leur ADN ;
  - Analyse de l'ADN d'un individu pour extraire ses caractéristiques, et déterminer l'existence de maladies génétiques ;
  - Estimations boursières ;
  - Reconnaissance de formes.

## I.5. Conclusion

Nous avons présenté une vue générale sur les méthodes de classification supervisée et qu'elles peuvent être basées sur des hypothèses probabilistes (classifieur naïf bayésien) ou sur des notions de proximité (plus proche voisins) ou bien encore sur des recherches dans des espaces d'hypothèses (arbre de décision, réseau de neurone et SVM).

En résumé, les algorithmes k-ppv et le classifieur naïf bayes sont très simple à mettre en œuvre en revanche, en pratique, il est impossible de les implémenter pour des dimensions plus élevées et un grand nombre de variables. Les arbres de décisions sont faciles à comprendre, à interpréter et ont besoin de peu de données pour apprendre contrairement aux réseaux de neurones qui ont besoin de beaucoup de données. Les résultats obtenus par ces derniers ne sont pas interprétables car le réseau généré par cet algorithme est une boîte noire tout comme la régression logistique aussi. Les forêts aléatoires sont très efficaces dans le cas de grande dimension et sont robustes contre les valeurs aberrantes, par contre sont souvent un peu lent durant l'apprentissage. Enfin, tout comme les réseaux de neurones, les SVMs ont une capacité de manipuler de grandes quantités de données par contre, ils utilisent des fonctions mathématiques complexes pour la classification et pour trouver les meilleures paramètres, cet algorithme demande un temps énorme pendant la phase d'entraînement.

Dans le prochain chapitre, nous allons voir comment le machine learning est utilisé dans le domaine du marketing en télécommunication.

## *Chapitre II*

### *Domaine d'étude*

## II.1. Introduction

La rétention des clients est devenue aujourd'hui très importante avec l'accroissement de la concurrence et la diversité des offres sur le marché. Dans l'industrie de la télécommunication par exemple, le taux du churn (ou attrition) annuel chez l'opérateur OTA varie de 8% à 11% près. Avec une telle compétitivité, la stratégie mise en place par l'opérateur, au lieu de tenter d'acquérir de nouveaux clients ou d'attirer les abonnés loin de la concurrence, il s'intéresse plutôt à la réduction des départs de ces clients surtout que le coût d'acquérir un nouveau client est cinq fois plus coûteux que d'en garder un abonné.

La rétention d'un client est relativement lié à son insatisfaction sur les offres ou la qualité des services, il faut donc détecter au plutôt ces motifs d'insatisfaction afin d'y apporter une réponse appropriée au plus vite.

L'analyse Big Data avec le Machine Learning et le Data Mining s'est révélé être un moyen efficace d'identifier et de prédire le désabonnement des clients afin d'anticiper la rupture des clients avant qu'elle se produise.

Dans ce chapitre, nous présenterons le profil de l'entreprise Djazzy.

## II.2. Etude de l'existant

### II.2.1. Fiche signalétique de Djazzy

La fiche signalétique ci-dessous résume l'identité de l'entreprise :

|                            |                                      |
|----------------------------|--------------------------------------|
| <b>Nom de l'entreprise</b> | <b>Djazzy</b>                        |
| Date de création           | 11 juillet 2001.                     |
| Dates clés                 | 2002 : lancement officiel de Djazzy. |

|                   |  |
|-------------------|--|
|                   | 2003 : le réseau est disponible dans les 48 wilayas.   |
|                   | 2004 : lancement d'Allô OTA.   |
|                   | 2014 : lancement de la 3G.   |
|                   | 2016 : lancement de la 4G.   |
| Forme juridique   | SPA (Société Par Actions).   |
| Siège sociale     | Lot n°37/4, Route de la wilaya Dar El Beïda, Alger  Algérie |
| Direction         | Vincenzo Nesci (président exécutif)<br>Matthieu Galvani (directeur général)  |
| Actionnaires      | État algérien (51 %)<br>VEON ex VimpelCom (49 %)   |
| Activité          | Opérateur de Télécommunications.   |
| produits          | Go, Good, Line, Play, Liberty, Millenium, Imtiyaz, Allô OTA, OTAxiphone, @migo, Speed.   |
| Société mère      | Global Telecom Holding<br>VimpelCom  |
| Sociétés sœurs    | Beeline, Kyivstar, Wind, Mobilink, Banglalink  |
| Effectif          | +4000  |
| Site web          | www.djezzy.dz  |
| Chiffre d'affaire | 164 milliards de DA [7]  |

Tableau II.1 : Fiche signalétique de Djezzy. [6]

### II.2.2. Historique

C'est en juillet 2001 que le groupe Orascom Telecom remporte la deuxième licence de téléphonie mobile en Algérie, et ce pour un montant de 737 millions de dollars. Avec 2,5

milliards USD d'investissement depuis 2001 à ce jour, fort d'un capital humain de plus de 4000 employés, et plusieurs offres et solutions (3G, 2G et VSAT), OTA (membre du groupe GTH/VimpelCom). Les numéros de téléphones des abonnés de Djezzy commencent par (07)7x xx xx xx , (07)9x xx xx xx ou (07)8x xx xx xx pour les numéros plus récents.

Le Chairman de cette entreprise est Vincenzo NESCI depuis juillet 2012. Le groupe égyptien Orascom Telecom a annoncé, mercredi 18 novembre 2009, avoir été informé officiellement, le mardi 17 novembre, par la Direction générale des impôts (DGI) d'un redressement fiscal de 596,6 millions de dollars pour sa filiale algérienne Djezzy. Le parc abonné de Djezzy s'est établi à 18,872 millions en novembre 2014.

En 2016, Djezzy, grâce à un partenariat avec la startup française Be-Bound, fondée par Albert Szulman et Yazid Chir, offre désormais à ses clients la possibilité d'avoir internet sur leur mobile sans connexion 3G, mais dès lors qu'ils captent un signal, même très faible en transférant les données via le réseau 2G ou le réseau SMS. [7]

### II.2.3. Structure fonctionnelle de Djezzy

La structure fonctionnelle consiste à diviser le travail dans Djezzy en fonctions (Technologie, administration, commercial, finance, chef de la stratégie d'entreprise et du développement des affaires).

#### II.2.3.1. La direction de l'administration:

Cette direction s'occupe:

- ✓ Des recrutements, et de la paie des employés ;
- ✓ De la maintenance des bureaux, de l'aménagement des boutiques et centres de services ;

- ✓ D'apporter l'aide matérielle nécessaire aux autres départements.

### **II.2.3.2. La direction des finances :**

Cette direction s'occupe :

- ✓ De la gestion des revenus de la compagnie ;
- ✓ Des prix, et des transferts bancaires ;
- ✓ Des financements, de l'établissement de bilan, et de la comptabilité.

### **II.2.3.3. La direction technologie :**

Cette direction a pour mission principale le contrôle et le suivi :

- ✓ Des projets de la technologie ;
- ✓ Des « cross-fonctionnel » projet entre les technologies et les autres ;
- ✓ Des départements de Djazzy (commercial, finances, ...etc.) ;
- ✓ De budget de la technologie ;
- ✓ Des indicateurs de performances techniques ;
- ✓ De l'audit des projets.

Il est constitué de trois départements : équipe réseau, équipe informatique et Numérisation/GIE (gestion de l'information de l'entreprise).

### **II.2.3.4. Chef de la stratégie d'entreprise et du développement des affaires:**

Il est responsable:

- ✓ Du rendement global de l'organisation des ventes;
- ✓ De la réalisation rentable des objectifs de l'organisation des ventes ;

- ✓ De l'alignement des objectifs de vente sur la stratégie commerciale de l'entreprise.

#### II.2.4. Domaine d'étude :

Notre travail se situe à « la direction technologie » et plus précisément au département « Numérisation/GIE » qui se charge d'explorer les données stockées dans les entrepôts de données et la plateforme de gestion de données.

Ce département est réparti en trois divisions :

- ✓ DWH/DMP (entrepôts de données, Data WareHouse en anglais/plateforme de gestion de données, Data Management Platform en anglais): un DWH intègre des données provenant d'un grand nombre de sources. Ces données sont nettoyées, organisées et traitées pour créer des rapports de l'Informatique Décisionnelle (Business Intelligence en anglais), suivre les tendances et soutenir la prise de décision en entreprise. Un DMP intègre également des données provenant de sources disparates, mais se concentre sur le marketing et les campagnes publicitaires.
- ✓ Informatique Décisionnelle : cette division consiste à collecter et analyser l'information portant sur les clients, les marchés, la concurrence...etc. Les analystes de l'Informatique Décisionnelle possédant le savoir-faire technique nécessaire pour écrire des requêtes et créer des rapports pour les responsables internes. Ici, des tableaux de bord sont intégrés pour rendre ces rapports plus visuels.
- ✓ Science des données (Data Science en anglais) : cette division s'appuie sur l'Intelligence Artificielle et le Machine Learning pour la création de modèle. Dans notre cas, notre tâche consiste à développer un modèle pour détecter l'attrition des clients. Ces modèles sont utilisés par le commerciale, marketing, ventes.

### II.2.5. Besoin de l'application à réaliser

L'attrition des clients, se produit lorsque ceux-ci cessent de faire affaire avec l'entreprise. L'entreprise est intéressée par l'identification de segments de ces clients car le prix pour acquérir un nouveau client est généralement supérieur à celui de conserver l'ancien.

En effet, notre travail est de développer un modèle permettant de prédire le churn d'un client en utilisant un algorithme de ML et de calculer sa probabilité de quitter. Et ceci à partir des données historiques des clients.

## II.3. Conclusion

Dans ce chapitre, nous avons présenté des informations générales sur l'entreprise et notamment le département Numérisation/GIE de la direction technologie où se situe notre travail.

Dans le prochain chapitre, nous allons présenter la conception du système proposé pour la prédiction du churn des clients, ceci en nous basant sur les clients de l'entreprise Djazzy.

# *Chapitre III*

## *Conception*

### III.1. Introduction

L'un des problèmes majeurs de la gestion des relations clients dans les sociétés de télécommunication est le départ des clients. L'objectif de l'application à présenter est de savoir quels types de clients risquent de se désabonner.

Dans ce chapitre, nous tacherons de citer l'approche proposée pour le problème de désabonnement des clients Djazzy, voir l'architecture de l'application ainsi que quelques algorithmes à utiliser.

### III.2. Approche proposée

L'approche proposée est de développer un modèle de prédiction du churn qui permet de classer le plus efficacement possible les clients désabonnés parmi les clients qui ne le sont pas, ainsi que de prédire ceux qui le feront ou pourraient éventuellement le faire prochainement et ce on se basant sur des données historiques de quelques clients de l'entreprise.

### III.3. Attrition (churn en anglais)

On va commencer par la définition de l'attrition (churn en anglais) donnée par l'entreprise. L'attrition d'un client est définie comme l'inactivité de celui-ci pendant un certain temps. La durée de cette inactivité dans la télécommunication est définie de 90 jours si le client n'effectue aucune activité et ne reçoit aucun appel et aucun message, cependant dans le cas où le client reçoit des appels ou messages la durée d'inactivité est augmentée de 30 jours soit donc 120 jours en tout.

### III.4. Cycle de vie de notre projet

Le cycle de vie d'un projet de machine learning peut se décomposer en plusieurs phases comme suit :

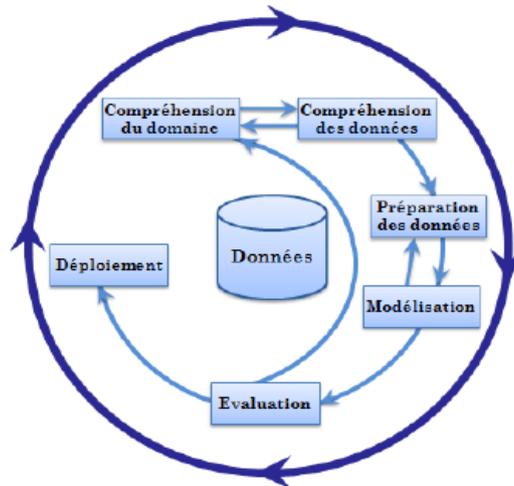


Figure III.1 : Cycle de vie d'un projet machine learning.

- La compréhension du domaine : comme tout projet, le premier objectif du responsable du projet est de bien comprendre ce que le client veut atteindre. Cette étape consiste à définir clairement les objectifs de l'entreprise et du client, tel que gagner de nouveaux clients, vendre plus de produits, etc.
- La compréhension des données : au cours de cette phase, il est important d'avoir une compréhension fine des données collectées de différentes sources de données disponible dans l'entreprise. Du coup, un contrôle de cohérence de données est effectué pour vérifier s'il convient aux objectifs du projet.
- La préparation des données : cette phase est estimée par de nombreux experts comme étant souvent la phase la plus longue et la plus critique d'un projet d'apprentissage automatique, car la qualité des résultats du modèle dépend de la qualité des données fournies. Par conséquent, les données doivent être organisées et structurées d'une manière spécifique. Cette étape rassemble toutes les activités de nettoyage et de prétraitement requises pour concevoir un tableau de données bien formé à partir duquel des modèles d'apprentissage peuvent être créés.

Généralement cette phase comprend trois étapes : transformation des données, nettoyage des données et sélection des variables.

- La modélisation : dans cette étape plusieurs algorithmes d'apprentissage automatique sont sélectionnés donnant ainsi plusieurs modèles candidats dont le meilleur sera exploité lors du déploiement.
- L'évaluation : avant l'adoption du modèle par l'organisation, il est judicieux de le soumettre à une évaluation minutieuse pour mesurer sa capacité de généralisation et s'assurer qu'il ne s'agisse guère d'un modèle souffrant de sur-apprentissage ou de sous-apprentissage.
- Le déploiement : cette dernière phase correspond à l'intégration du modèle au sein de l'organisation et sa mise à l'épreuve dans le monde réel afin d'accomplir la tâche pour laquelle il fut développé.

### III.5. Architecture du projet

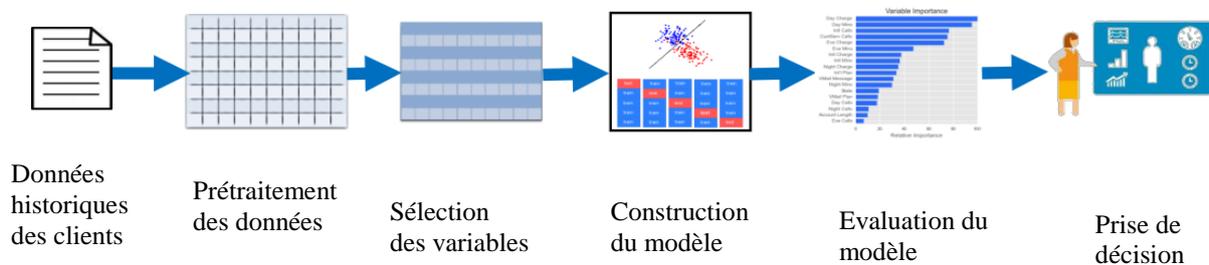


Figure III.2 : Architecture de l'application.

L'architecture proposée pour notre système de prédiction du churn est constituée de cinq phases :

1. Prétraitement des données clients
2. Sélection des fonctionnalités pertinentes pour développer les modèles de prédiction
3. Construction des modèles

4. Evaluation et calcul de la précision des modèles
5. Prise de décision pour la rétention des clients.

### III.6. Algorithmes à utiliser

Plusieurs algorithmes ont déjà été utilisés pour la prédiction du churn dans le l'industrie de la télécommunication, cependant selon l'échantillon utilisé pour l'étude le taux de prédiction été différent. Parmi ces algorithmes de machine learning qu'on utilisera, on cite :

- Les arbres de décision
- Les forêts d'arbres décisionnels (Random Forest)
- Les réseaux de neurones
- La régression logistique

### III.7. Conclusion

A travers ce chapitre, nous avons exposé la conception de notre projet et son architecture qu'on mettra en place dans le chapitre suivant. Nous avons vu le cycle de vie d'un projet en apprentissage automatique.

Dans le prochain chapitre, nous exposerons les résultats expérimentaux obtenus lors des tests. Nous présentons aussi nos perspectives pour le futur dans le domaine de la détection de la perte des clients dans le secteur de télécommunication.

*Chapitre IV*

*Implémentation*

## IV.1. Introduction

Ce chapitre est consacré à la partie réalisation et tests de notre projet. En effet nous commençons par présenter les logiciels et librairies utilisées ainsi que la base de données sur laquelle nous avons construit et testé notre système, pour ensuite exposer les résultats expérimentaux obtenus. Au final, nous discuterons ces résultats et proposerons des améliorations afin d'obtenir des résultats meilleurs.

## IV.2. Logiciels et librairies

### IV.2.1. Logiciels

- **R** : R est à la fois un logiciel statistique et un langage de programmation. R est un logiciel d'analyse statistique et graphique créé par Ross Ihaka et Robert Gentleman. Il dispose d'une bibliothèque très large de fonctions statistiques, d'autant plus larges qu'il est possibles d'en intégrer d'autres par le système des packages.

Un point fort de R réside dans le fait que ce logiciel est distribué librement. Son installation peut-être mise en œuvre à partir du site internet du *Comprehensive R Archive Network (CRAN)* qui d'une part met à disposition les exécutable et d'autres parts donne des informations relatives à la procédure d'installation.

- **RStudio** : est un environnement de développement intégré (IDE en anglais) pour R libre, gratuit et multiplateforme. Son interface est organisée de sorte que l'utilisateur puisse afficher clairement les graphiques, les tableaux de données et le code R et tout afficher en même temps.

### IV.2.2. Librairies

- **readxl** : lit un fichier Excel de type « .xls » ou « .xlsx » dans une trame de données R ;
- **sqldf** : est un package R utilisé pour l'exécution d'instruction SQL sur des trames de données R ;

- FSelector : ce package fournit des fonctions permettant de sélectionner des attributs dans un jeu de données connu.
- rpart/rpart.plot : le package rpart propose une implémentation des méthodes de construction des arbres de décision. Quant au package « rpart.plot » est utilisé pour tracer un modèle de type « rpart ».
- randomForest : ce package implémente l'algorithme de forêt aléatoire de Breiman pour la classification et la régression.
- neuralnet : un package utilisé pour entraîner les réseaux de neurones utilisant la rétropropagation, la rétropropagation résiliente avec ou sans recul du poids, ou avec la version modifiée à convergente globale d'Anastasiadis et al. Ce package permet des paramètres flexibles via le choix personnalisé de la fonction d'erreur et d'activation. En outre, le calcul des poids généralisés est mis en œuvre.
- caret : contient diverses fonctions pour la formation et le traçage de modèles de classification et de régression.

## IV.3. Mise en œuvre

### IV.3.1. Compréhension du problème métier

Dans cette phase initiale, le but est de comprendre les objectifs du projet.

L'objectif de la prédiction du churn est d'identifier les clients qui sont susceptibles de se désabonner et de quitter l'entreprise afin de d'améliorer la prise de décision et de mettre en place des actions de rétention. Pour ce faire, nous considérons les clients qui ont tourné et analysons leurs données sur une période de trois mois.

### IV.3.2. Compréhension des données

Pour notre étude, l'entreprise nous a fournis trois bases de données se présentant comme suit :

- Base d'activité contenant la date de la dernière transaction du client qui s'étend du 15/08/2018 jusqu'au 13/10/2018 et la première transaction qui a suivi cette dernière qui est du 14/10/2018 au 12/12/2018 ;
- Les données de facturation telles que les montants rechargés par le client, ces montants sont des nombres réels ;
- Les données relatives à des utilisations telles que la durée des appels effectués en national et international (nombre réel) exprimés en secondes, le nombre de messages envoyés par mois (nombre entier).

Avant de décrire les variables de ces bases, nous avons préféré joindre ces trois bases de données en une seule avec une simple requête sql pour mieux les visualiser. Et ainsi nous formons notre trame de données sur laquelle on va travailler.

Notre ensemble de données contient 12000 lignes et 24 variables. Le tableau ci-dessous explique le rôle de chaque fonctionnalité.

| <b>Le nom de la fonctionnalité</b> | <b>Description</b>  | <b>Type</b> |
|------------------------------------|---|-------------|
| Service_Number                     | Le numéro de service du client et il représente son identifiant | numeric     |
| LAST_TRANSDATE                     | La date de la dernière transaction du client.                   | Date        |
| FIRST_TRANSDATE_AFTER_LAST         | La date de la première transaction après la dernière.           | Date        |
| REV_M6                             | Le revenu durant mois 6 du client.                              | numeric     |
| REV_M5                             | Le revenu du mois 5 durant client.                              | numeric     |
| REV_M4                             | Le revenu du mois 4 durant client.                              | numeric     |

|                                     |  |         |
|-------------------------------------|--|---------|
| TRAF_OUT_VOICE_ONNET_M6             | La durée des appels émis vers le même opérateur durant mois 6.       | numeric |
| TRAF_OUT_VOICE_ONNET_M5             | La durée des appels émis vers le même opérateur durant mois 5.       | numeric |
| TRAF_OUT_VOICE_ONNET_M4             | La durée des appels émis vers le même opérateur durant mois 4.       | numeric |
| TRAF_OUT_VOICE_OFFNET_M6            | La durée des appels émis vers un opérateur concurrent durant mois 6. | numeric |
| TRAF_OUT_VOICE_OFFNET_M5            | La durée des appels émis vers un opérateur concurrent durant mois 5. | numeric |
| TRAF_OUT_VOICE_OFFNET_M4            | La durée des appels émis vers un opérateur concurrent durant mois 4. | numeric |
| TRAF_OUT_VOICE_INTERNATION<br>AL_M6 | La durée des appels émis vers l'international durant mois 6.         | numeric |
| TRAF_OUT_VOICE_INTERNATION<br>AL_M5 | La durée des appels émis vers  | numeric |

|                                 |  |         |
|---------------------------------|--|---------|
|                                 | l'international durant mois 5.                                     |         |
| TRAF_OUT_VOICE_INTERNATIONAL_M4 | La durée des appels émis vers l'international durant mois 4.       | numeric |
| TRAF_OUT_VOICE_ROAMING_M6       | La durée des appels émis en roaming <sup>1</sup> durant le mois 6. | numeric |
| TRAF_OUT_VOICE_ROAMING_M5       | La durée des appels émis en roaming durant le mois 5.              | numeric |
| TRAF_OUT_VOICE_ROAMING_M4       | La durée des appels émis en roaming durant le mois 4.              | numeric |
| CNT_OUT_SMS_TOTAL_M6            | Le nombre de messages envoyés durant le mois 6.                    | numeric |
| CNT_OUT_SMS_TOTAL_M5            | Le nombre de messages envoyés durant le mois 5.                    | numeric |
| CNT_OUT_SMS_TOTAL_M4            | Le nombre de messages envoyés durant le mois 4.                    | numeric |
| REV_REFILL_M6                   | Le total rechargé durant le mois 6.                                | numeric |
| REV_REFILL_M5                   | Le total rechargé durant le mois 5.                                | numeric |

<sup>1</sup> Roaming : c'est un service qui permet à un abonné d'un réseau téléphonique d'utiliser sa carte SIM dans une autre zone que la zone d'origine de l'opérateur, grâce à des accords entre les opérateurs.

|               |  |         |
|---------------|--|---------|
| REV_REFILL_M4 | Le total rechargé<br>durant le mois 4. | numeric |
|---------------|--|---------|

Tableau IV.1: Description des variables de la trame de données.

### IV.3.3. Préparation des données

Avant de passer au processus de prétraitement des données nous devons d'abord créer la variable cible. Dans la télécommunication une règle est définie, en théorie, si un client n'effectue aucune activité pendant 90 jours il est considéré comme étant un churner. En pratique, selon un manager de l'entreprise 90% des clients qui ne sont pas actifs pendant 60 jours finissent par quitter l'entreprise. Sur cette analyse, nous allons créer une nouvelle variable qu'on nommera Date\_diff qui calculait la différence des deux champs dates de la base d'activité. Puis, à partir de cette nouvelle variable nous avons déduit la variable cible nommée CHURN de type facteur qui valait « 1 » si le client allait quitter et « 0 » s'il restait. Ainsi nous obtenons 2000 clients étiquetés « churner » et 10000 clients étiquetés « non-churner ».

#### IV.3.3.1. Nettoyage des données

##### - Traitement des valeurs manquantes

Les valeurs manquantes figurants dans notre trame de données sont dues au caractère « ? » qui se trouvait dans les bases initiales. Ces « ? » indiquaient que le client n'ait effectué aucune action du coup on a remplacé ces « ? » par des zéros.

##### - Traitement des valeurs aberrantes

Pour commencer, nous allons d'abord détecter les valeurs extrêmes de l'échantillon. Pour ce faire, nous avons opté pour la méthode suivante : pour calculer les valeurs aberrantes de chaque variable nous avons utilisé la règle ( $moyenne + 3 * \text{écart\_type}$ ) en modifiant le paramètre 3 de façon à obtenir la courbe normale.

Puis, nous avons calculé d'autres variables nommées `outlier_flag_[variablei]` pour chaque variable<sub>i</sub>, ensuite nous avons calculé la variable globale `outlier_flag_global` pour voir quels sont les enregistrements qui ont des valeurs extrêmes. Les résultats obtenus sont résumés dans le tableau suivant :

|                     | <i>churner</i> | <i>Non-churner</i> |
|---------------------|----------------|--------------------|
| <i>Aberrant</i>     | 9717           | 283                |
| <i>Non-aberrant</i> | 1918           | 82                 |

Tableau IV.2 : Résultats de détection des valeurs extrêmes.

### IV.3.3.2. Dérivation d'autres variables

Pour que le modèle puisse apprendre mieux il a besoin d'autres variables.

Dans notre cas, nous avons généré les variables moyennes pour les variables se portant sur chaque trois mois (`AVG_REV`, `AVG_TRAF_OUT_VOICE_ONNET`, `AVG_TRAF_OUT_VOICE_OFFNET`, `AVG_TRAF_OUT_VOICE_INTERNATIONAL`, `AVG_TRAF_OUT_VOICE_ROAMING`, `AVG_CNT_OUT_SMS_TOTAL`, `AVG_REV_REFILL`) qui sont de type réel, les variations en pourcentage et réelles des variables sur trois mois aussi (`DIFF_REV5_4`, `DIFF_REV6_5`, `DIFF_REV5_4_prct`, `DIFF_REV6_5_prct...`) et des variables flags (de type facteur) pour les variables SMS et TRAFIC (`SMS_M6_FLAG`, `ONNET_M6_FLAG`, `OFFNET_M5_FLAG...`).

- Traitement des valeurs manquantes (NA's) et infinies (INF's) des variables dérivées :

Lors de la création des variables de variation en pourcentage y'a eu quelques valeurs NA's et INF's qui ont été générées du coup, pour les traiter nous avons remplacé les NA's par des 0 car elles sont dues aux valeurs des deux variables qui sont égales à 0 (c.à.d. 0/0), puis nous avons remplacé les valeurs INF's par 1 qui vaut 100%.

A la fin de ces traitements nous obtenons un échantillon de 12000 enregistrements et 98 variables.

### IV.3.3.3. Sélection des variables

La sélection des variables fait référence au processus de sélection d'un sous-ensemble d'attributs pertinent d'un ensemble de variables. Cela réduit le nombre d'attributs d'entrée de l'algorithme d'apprentissage, ce qui réduit considérablement le temps et les ressources nécessaires à l'apprentissage du modèle. Pour notre étude, au départ, nous avons choisi la méthode chi-squared.

Pour utiliser le chi-squared lors de la sélection des fonctionnalités, l'algorithme calcule le  $\chi^2$  entre chaque variable et la classe cible et sélectionne le nombre souhaité de variables avec les meilleures scores  $\chi^2$ .

Le Service\_Number et LAST\_TRANSDATE ont été supprimées car la première est identique pour chaque client et la seconde est inutile pour la prédiction.

```

547 #importation de la librairie
548 library(FSelector)
549 #suppression des variables service_number et last_transdate|
550 data3 <- data[,-c(1,2)]
551 weights <- chi.squared(CHURN~., data3)
552 print(weights)
553 subset<- cutoff.k(weights, 20)
554 f<- as.simple.formula(subset, "CHURN")
555 print(f)

```

Figure IV.1 : Sélection des variables avec chi-squared.

Le tableau suivant résume les variables obtenues par la sélection chi-squared :

| Nom de la variable   |
|----------------------|
| REV_M6               |
| DIFF_REV6_5_prct     |
| REV_REFILL_M6        |
| DIFF_REFILL_6_5_prct |
| DIFF_REV6_5          |
| DIFF_REFILL_6_5      |
| DIFF_ONNET_6_5_prct  |

|                          |
|--------------------------|
| DIFF_OFFNET_6_5_prct     |
| TRAF_OUT_VOICE_OFFNET_M6 |
| TRAF_OUT_VOICE_ONNET_M6  |
| OFFNET_M6_FLAG           |
| ONNET_M6_FLAG            |
| AVG_REV                  |
| AVG_REV_REFILL           |
| DIFF_OFFNET_6_5          |
| DIFF_ONNET_6_5           |
| DIFF_SMS_6_5_prct        |
| DIFF_ONNET_5_4_prct      |
| DIFF_SMS_6_5             |
| REV_M4                   |

Tableau IV.3 : Liste des variables pertinentes retournée par la chi-squared.

Après avoir analysé ces variables, nous avons remarqué qu'elles n'étaient pas vraiment logiques. Alors, après longue réflexion, nous avons sélectionné les variables de variation en pourcentage et celle de trafic et SMS en flag. Nous obtenons les variables pertinentes finales qui seront intégrées au modèle pour l'apprentissage suivantes :

| <b>Nom de la variable</b>   |
|-----------------------------|
| DIFF_REV5_4_prct            |
| DIFF_REV6_5_prct            |
| DIFF_ONNET_5_4_prct         |
| DIFF_ONNET_6_5_prct         |
| DIFF_OFFNET_5_4_prct        |
| DIFF_OFFNET_6_5_prct        |
| DIFF_INTERNATIONAL_5_4_prct |
| DIFF_INTERNATIONAL_6_5_prct |
| DIFF_ROAMING_5_4_prct       |

|                       |
|-----------------------|
| DIFF_ROAMING_6_5_prc  |
| DIFF_SMS_5_4_prc      |
| DIFF_SMS_6_5_prc      |
| DIFF_REFILL_5_4_prc   |
| DIFF_REFILL_6_5_prc   |
| SMS_M6_FLAG           |
| SMS_M5_FLAG           |
| SMS_M4_FLAG           |
| ONNET_M6_FLAG         |
| ONNET_M5_FLAG         |
| ONNET_M4_FLAG         |
| OFFNET_M6_FLAG        |
| OFFNET_M5_FLAG        |
| OFFNET_M4_FLAG        |
| INTERNATIONAL_M6_FLAG |
| INTERNATIONAL_M5_FLAG |
| INTERNATIONAL_M4_FLAG |
| ROAMING_M6_FLAG       |
| ROAMING_M5_FLAG       |
| ROAMING_M4_FLAG       |

Tableau IV.4 : Liste des variables pertinentes finales sélectionnée.

#### IV.3.4. Modélisation

La partition choisie pour les données pour la création des modèles est la suivante :

- 80% des données pour l'apprentissage
- 20% des données pour le test.

```

486 #partition des données
487 set.seed(1234)
488 pd <- sample(2,nrow(data2),replace = TRUE,prob = c(0.8,0.2))
489 train <- data2[pd==1,]
490 validate <- data2[pd==2,]

```

Figure IV.2 : Code pour partitionner les données.

Dans notre échantillon, la classe cible est déséquilibrée (84% non-churners et seulement 16% churners), ce qui pourrait avoir un impact négatif important sur les modèles finaux.

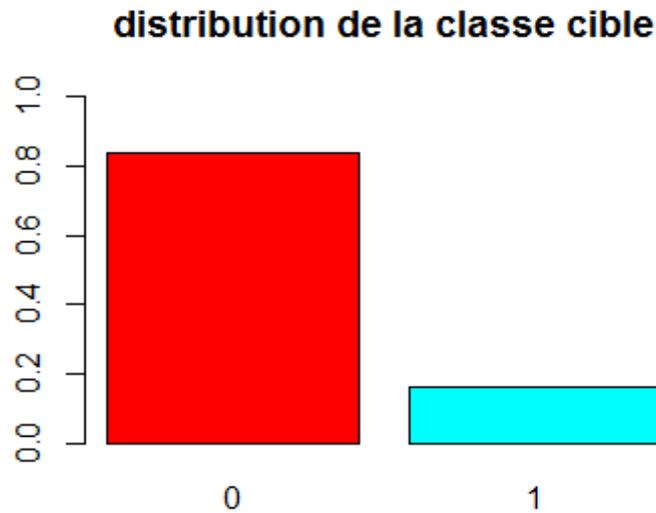


Figure IV.3 : Distribution de la classe cible.

Pour rééquilibrer l'échantillon de formation deux solutions sont possibles :

Le sous-échantillonnage qui consiste à réduire aléatoirement la taille de l'échantillon en supprimant des observations de la classe majoritaire (ici quand la variable cible vaut 0) pour qu'elle soit équilibrée avec la classe minoritaire. Cette méthode est préconisée lorsque nous avons beaucoup de données. Et comme la taille de l'échantillon de formation est de 6971 et le nombre de clients de la classe 0 est de 5828 nous avons éliminé cette solution.

Le sur-échantillonnage consiste à dupliquer les observations de la classe minoritaire pour équilibrer avec l'autre classe. Le problème avec cette solution est que en dupliquant les observations on peut facilement tomber dans le problème de sur-apprentissage. La solution répondant à cet effet est de synthétiser de nouvelles données à partir des données existantes en utilisant l'algorithme SMOTE (Synthetic Minority Oversampling Technique) qui parcourt toutes les observations de la classe minoritaire, cherche ses  $k$  plus proches voisins puis synthétise aléatoirement de nouvelles données entre ces deux points.

```

491 ▾ #####équilibrer le jeu de données####
492 library(DMwR)
493 balanced.data <- SMOTE(CHURN ~., train, perc.over = 200, k = 3, perc.under = 255)
494 as.data.frame(table(balanced.data$CHURN))

```

Figure IV.4 : Implémentation de l'algorithme SMOTE sous R.

Cette méthode SMOTE () sous R s'avère être une méthode hybride (elle utilise les deux solutions sur-échantillonnage et sous-échantillonnage) et les résultats obtenus par celle-ci touche même à la classe majoritaire, chose qu'on ne souhaitait pas avoir. Alors nous avons opté pour la solution suivante : créer un nouveau data frame qui unit les observations synthétisés de la classe minoritaires retournées par cette méthode avec les observations de la classe majoritaire existantes déjà dans l'échantillon d'apprentissage. Ainsi, nous obtenons un effectif de 7773 pour la classe majoritaire (non-churner) et 7711 pour la classe minoritaire (churner).

```

496 ▾ #####équilibrer le jeu de données####
497 library(DMwR)
498 balanced_data <- SMOTE(CHURN ~., train, perc.over = 400, k = 3, perc.under = 100)
499 as.data.frame(table(balanced_data$CHURN))
500 churners <- sqldf("select * from balanced_data where balanced_data.CHURN = 1")
501 no_churners <- sqldf("select * from train where train.CHURN = 0")
502 trainset <- sqldf("select * from churners union select * from no_churners")
503

```

Figure IV.5 : Création d'un data frame d'apprentissage équilibré.

Voici la distribution de la classe après l'avoir équilibré :

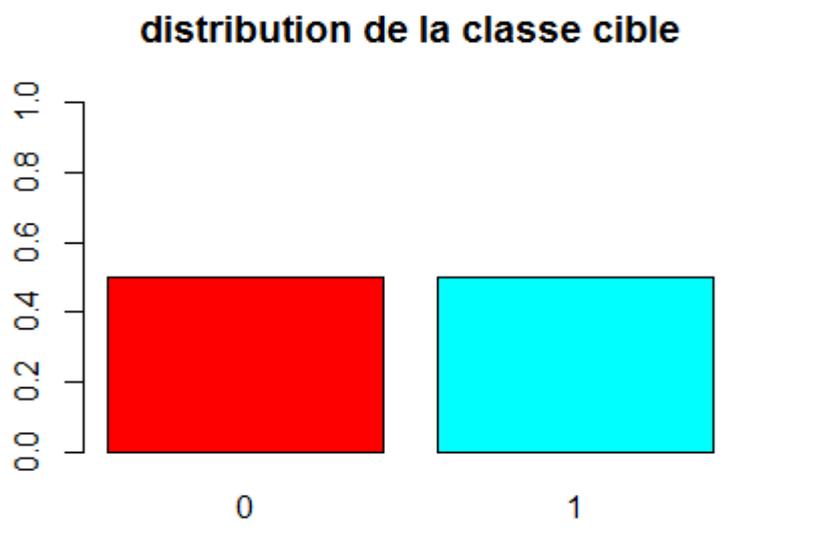


Figure IV.6 : Distribution de la classe cible après sur-échantillonnage.

➤ Implémentation du modèle arbre de décision

```

497 - #####arbre de décision#####
498 library(rpart)
499 library(rpart.plot)
500 tree1 <- rpart(CHURN ~ DIFF_REV5_4_prct + DIFF_REV6_5_prct + DIFF_ONNET_5_4_prct + DIFF_ONNET_6_5_prct +
501               DIFF_OFFNET_5_4_prct + DIFF_OFFNET_6_5_prct + DIFF_INTERNATIONAL_5_4_prct + DIFF_INTERNATIONAL_6_5_prct +
502               DIFF_ROAMING_5_4_prct + DIFF_ROAMING_6_5_prct + DIFF_SMS_5_4_prct + DIFF_SMS_6_5_prct + DIFF_REFILL_5_4_prct +
503               DIFF_REFILL_6_5_prct + SMS_M6_FLAG + SMS_M5_FLAG + SMS_M4_FLAG + ONNET_M6_FLAG + ONNET_M5_FLAG + ONNET_M4_FLAG +
504               OFFNET_M6_FLAG + OFFNET_M5_FLAG + OFFNET_M4_FLAG + INTERNATIONAL_M6_FLAG + INTERNATIONAL_M5_FLAG + INTERNATIONAL_M4_FLAG +
505               ROAMING_M6_FLAG + ROAMING_M5_FLAG + ROAMING_M4_FLAG, data = balanced.data, method="class")
506 rpart.plot(tree1)

```

Figure IV.7 : Algorithme de l'arbre de décision.

La figure suivante représente l'arbre de décision obtenu :

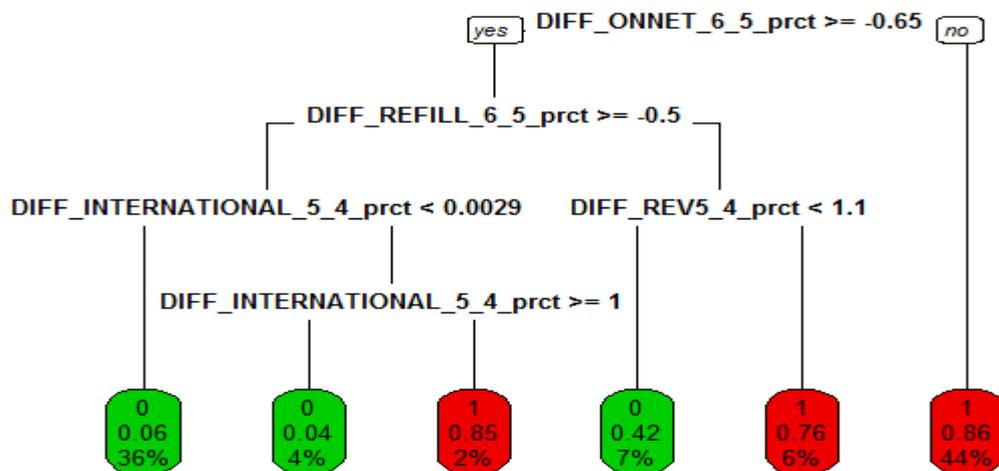


Figure IV.8 : Arbre de décision obtenu.

➤ Implémentation du modèle forêt d'arbres décisionnels

```

520 - #####random forest#####
521 library(randomForest)
522 rfModel <- randomForest(CHURN ~ CHURN + DIFF_REV5_4_prct + DIFF_REV6_5_prct + DIFF_ONNET_5_4_prct + DIFF_ONNET_6_5_prct +
523                       DIFF_OFFNET_5_4_prct + DIFF_OFFNET_6_5_prct + DIFF_INTERNATIONAL_5_4_prct + DIFF_INTERNATIONAL_6_5_prct +
524                       DIFF_ROAMING_5_4_prct + DIFF_ROAMING_6_5_prct + DIFF_SMS_5_4_prct + DIFF_SMS_6_5_prct + DIFF_REFILL_5_4_prct +
525                       DIFF_REFILL_6_5_prct + SMS_M6_FLAG + SMS_M5_FLAG + SMS_M4_FLAG + ONNET_M6_FLAG + ONNET_M5_FLAG + ONNET_M4_FLAG +
526                       OFFNET_M6_FLAG + OFFNET_M5_FLAG + OFFNET_M4_FLAG + INTERNATIONAL_M6_FLAG + INTERNATIONAL_M5_FLAG + INTERNATIONAL_M4_FLAG +
527                       ROAMING_M6_FLAG + ROAMING_M5_FLAG + ROAMING_M4_FLAG, data = balanced.data, ntree=700)

```

Figure IV.9 : Algorithme de forêt des arbres décisionnels.

➤ Implémentation du modèle de régression logistique

```

542 #####"régression logistique#####"
543 fit <- glm(CHURN ~ DIFF_REV5_4_prct + DIFF_REV6_5_prct + DIFF_ONNET_5_4_prct + DIFF_ONNET_6_5_prct +
544         DIFF_OFFNET_5_4_prct + DIFF_OFFNET_6_5_prct + DIFF_INTERNATIONAL_5_4_prct + DIFF_INTERNATIONAL_6_5_prct +
545         DIFF_ROAMING_5_4_prct +
546         DIFF_ROAMING_6_5_prct + DIFF_SMS_5_4_prct + DIFF_SMS_6_5_prct + DIFF_REFILL_5_4_prct +
547         DIFF_REFILL_6_5_prct + SMS_M6_FLAG + SMS_M5_FLAG + SMS_M4_FLAG + ONNET_M6_FLAG + ONNET_M5_FLAG + ONNET_M4_FLAG +
548         OFFNET_M6_FLAG + OFFNET_M5_FLAG + OFFNET_M4_FLAG + INTERNATIONAL_M6_FLAG + INTERNATIONAL_M5_FLAG + INTERNATIONAL_M4_FLAG
549         + ROAMING_M6_FLAG + ROAMING_M5_FLAG + ROAMING_M4_FLAG, data = balanced.data, family=binomial)

```

Figure IV.10 : Algorithme de régression logistique.

### ➤ Implémentation du modèle de réseau de neurones

```

568 #####réseau de neurones#####
569 library(neuralnet)
570 balanced.data <- model.matrix(
571   ~ CHURN + DIFF_REV5_4_prct + DIFF_REV6_5_prct + DIFF_ONNET_5_4_prct + DIFF_ONNET_6_5_prct +
572     DIFF_OFFNET_5_4_prct + DIFF_OFFNET_6_5_prct + DIFF_INTERNATIONAL_5_4_prct + DIFF_INTERNATIONAL_6_5_prct +
573     DIFF_ROAMING_5_4_prct +
574     DIFF_ROAMING_6_5_prct + DIFF_SMS_5_4_prct + DIFF_SMS_6_5_prct + DIFF_REFILL_5_4_prct +
575     DIFF_REFILL_6_5_prct + SMS_M6_FLAG + SMS_M5_FLAG + SMS_M4_FLAG + ONNET_M6_FLAG + ONNET_M5_FLAG + ONNET_M4_FLAG +
576     OFFNET_M6_FLAG + OFFNET_M5_FLAG + OFFNET_M4_FLAG + INTERNATIONAL_M6_FLAG + INTERNATIONAL_M5_FLAG + INTERNATIONAL_M4_FLAG
577     + ROAMING_M6_FLAG + ROAMING_M5_FLAG + ROAMING_M4_FLAG,
578   data = balanced.data
579 )
580 validate <- model.matrix(
581   ~ CHURN + DIFF_REV5_4_prct + DIFF_REV6_5_prct + DIFF_ONNET_5_4_prct + DIFF_ONNET_6_5_prct +
582     DIFF_OFFNET_5_4_prct + DIFF_OFFNET_6_5_prct + DIFF_INTERNATIONAL_5_4_prct + DIFF_INTERNATIONAL_6_5_prct +
583     DIFF_ROAMING_5_4_prct +
584     DIFF_ROAMING_6_5_prct + DIFF_SMS_5_4_prct + DIFF_SMS_6_5_prct + DIFF_REFILL_5_4_prct +
585     DIFF_REFILL_6_5_prct + SMS_M6_FLAG + SMS_M5_FLAG + SMS_M4_FLAG + ONNET_M6_FLAG + ONNET_M5_FLAG + ONNET_M4_FLAG +
586     OFFNET_M6_FLAG + OFFNET_M5_FLAG + OFFNET_M4_FLAG + INTERNATIONAL_M6_FLAG + INTERNATIONAL_M5_FLAG + INTERNATIONAL_M4_FLAG
587     + ROAMING_M6_FLAG + ROAMING_M5_FLAG + ROAMING_M4_FLAG,
588   data = validate
589 )
590 nn <- neuralnet(CHURN1 ~ DIFF_REV5_4_prct + DIFF_REV6_5_prct + DIFF_ONNET_5_4_prct + DIFF_ONNET_6_5_prct +
591               DIFF_OFFNET_5_4_prct + DIFF_OFFNET_6_5_prct + DIFF_INTERNATIONAL_5_4_prct + DIFF_INTERNATIONAL_6_5_prct +
592               DIFF_ROAMING_5_4_prct +
593               DIFF_ROAMING_6_5_prct + DIFF_SMS_5_4_prct + DIFF_SMS_6_5_prct + DIFF_REFILL_5_4_prct +
594               DIFF_REFILL_6_5_prct + SMS_M6_FLAG1 + SMS_M5_FLAG1 + SMS_M4_FLAG1 + ONNET_M6_FLAG1 + ONNET_M5_FLAG1 + ONNET_M4_FLAG1 +
595               OFFNET_M6_FLAG1 + OFFNET_M5_FLAG1 + OFFNET_M4_FLAG1 + INTERNATIONAL_M6_FLAG1 + INTERNATIONAL_M5_FLAG1 + INTERNATIONAL_M4_FLAG1 +
596               ROAMING_M6_FLAG1 + ROAMING_M5_FLAG1 + ROAMING_M4_FLAG1 , data = balanced.data, hidden = 1, act.fct = "logistic", err.fct = "ce",
597               linear.output = FALSE, stepmax = 1e7)

```

Figure IV.11 : Algorithme de réseau de neurones.

#### IV.3.5. Evaluation

Afin de mesurer la performance d'un modèle d'apprentissage automatique on utilise généralement la matrice de confusion qui est basée sur le nombre d'enregistrements de test correctement et incorrectement prédit par le modèle.

|                     | <i>Classe « 1 »</i> | <i>Classe « 0 »</i> |
|---------------------|---------------------|---------------------|
| <i>Classe « 1 »</i> | VP                  | FN                  |
| <i>Classe « 0 »</i> | FP                  | VN                  |

Tableau IV.5 : Exemple de matrice de confusion.

- VP (Vrai Positif) : élément de la classe « 1 » correctement prédit.
- VN (Vrai Négatif) : élément de la classe « 0 » correctement prédit.
- FP (Faux Positif) : élément de la classe « 1 » mal prédit.
- FN (Faux Négatif) : élément de la classe « 0 » mal prédit.

A partir de la matrice de confusion, on peut en tirer plusieurs indicateurs pour estimer le modèle :

- La précision (accuracy) : Indique la performance globale du modèle, calculée comme suit :

$$\text{Précision} = \frac{VP + VN}{VP + VN + FP + FN}$$

- La sensibilité (sensitivity) : le taux des observations positives correctement prédites. Elle est calculée :

$$\text{Sensibilité} = \frac{VP}{VP + FN}$$

- La spécificité (specificity) : le taux des observations négatives correctement prédites, qui est calculée comme suit :

$$\text{Spécificité} = \frac{VN}{VN + FP}$$

#### IV.3.5.1. Evaluation de l'arbre de décision :

Dans le tableau suivant la matrice de confusion obtenue pour le modèle de l'arbre de décision :

|                     | <i>Classe « 0 »</i> | <i>Classe « 1 »</i> |
|---------------------|---------------------|---------------------|
| <i>Classe « 0 »</i> | 1632                | 312                 |
| <i>Classe « 1 »</i> | 118                 | 257                 |

Tableau IV.6 : Matrice de confusion de l'arbre de décision.

La précision du modèle est calculée à l'aide de la formule suivante :

D'après la matrice de confusion retournée, nous avons obtenu les valeurs de performance suivantes :

|                          | <i>Précision</i> | <i>Sensibilité</i> | <i>Spécificité</i> |
|--------------------------|------------------|--------------------|--------------------|
| <i>Arbre de décision</i> | 81.46%           | 45.17%             | 93.26 %            |

Tableau IV.7 : Résultats de l'évaluation de l'arbre de décision.

#### IV.3.5.2. Evaluation des forêts aléatoires

Voici la matrice de confusion obtenue de forêt des arbres décisionnels :

|                     | <i>Classe « 0 »</i> | <i>Classe « 1 »</i> |
|---------------------|---------------------|---------------------|
| <i>Classe « 0 »</i> | 1826                | 136                 |
| <i>Classe « 1 »</i> | 118                 | 239                 |

Tableau IV.8 : Matrice de confusion de forêt de l'arbre décisionnel.

Les mesures de performances sont les suivantes :

|                          | <i>Précision</i> | <i>Sensibilité</i> | <i>Spécificité</i> |
|--------------------------|------------------|--------------------|--------------------|
| <i>Forêts aléatoires</i> | 89.05%           | 63.73%             | 93.93%             |

Tableau IV.9 : Résultats de l'évaluation des forêts aléatoires.

### IV.3.5.3. Evaluation de la régression logistique

La matrice de confusion obtenue pour le modèle de régression logistique est la suivante :

|                     | <i>Classe « 0 »</i> | <i>Classe « 1 »</i> |
|---------------------|---------------------|---------------------|
| <i>Classe « 0 »</i> | 1659                | 138                 |
| <i>Classe « 1 »</i> | 285                 | 237                 |

Tableau IV.10 : Matrice de confusion du modèle régression logistique.

Les résultats de l'évaluation de la régression logistique sont les suivants :

|                                  | <i>Précision</i> | <i>Sensibilité</i> | <i>Spécificité</i> |
|----------------------------------|------------------|--------------------|--------------------|
| <i>Régression<br/>logistique</i> | 81.76%           | 63.2%              | 85.34%             |

Tableau IV.11 : Résultats de l'évaluation la régression logistique.

### IV.3.5.4. Evaluation des réseaux de neurones

Le tableau ci-dessous représente la matrice de confusion obtenue pour les réseaux de neurones :

|                     | <i>Classe « 0 »</i> | <i>Classe « 1 »</i> |
|---------------------|---------------------|---------------------|
| <i>Classe « 0 »</i> | 1716                | 134                 |
| <i>Classe « 1 »</i> | 228                 | 241                 |

Tableau IV.12 : Matrice de confusion des réseaux de neurones.

Le tableau suivant résume les résultats de l'évaluation des réseaux de neurones :

|                          | <i>Précision</i> | <i>Sensibilité</i> | <i>Spécificité</i> |
|--------------------------|------------------|--------------------|--------------------|
| <i>Réseau de neurone</i> | 81.76%           | 63.2%              | 85.34%             |

Tableau IV.13 : Résultats de l'évaluation du réseau de neurones.

#### IV.4. Résultats et discussion

Le tableau ci-dessous résume les résultats obtenus pour l'évaluation de chaque modèle :

|                    | <i>Arbre de<br/>décision</i> | <i>Forêts<br/>aléatoires</i> | <i>Régression<br/>logistique</i> | <i>Réseau de<br/>neurone</i> |
|--------------------|------------------------------|------------------------------|----------------------------------|------------------------------|
| <i>Précision</i>   | 81.46%                       | 89.05%                       | 81.76%                           | 84.39%                       |
| <i>Sensibilité</i> | 45.17%                       | 63.73%                       | 63.2%                            | 88.27%                       |
| <i>Spécificité</i> | 93.26%                       | 93.93%                       | 85.34%                           | 64.27%                       |

Tableau IV.14 : comparaison des résultats des mesures d'évaluation des pour chaque algorithme d'apprentissage supervisé.

Comme indiqué dans le tableau ci-dessus, nous observons que les forêts aléatoires ont surperformé les autres algorithmes testés avec une valeur de précision de 89.05%. Le réseau de neurone occupe la deuxième place avec une valeur de précision de 84.39%. Tandis que, la

régression logistique et l'arbre de décision arrivent en dernière position avec des valeurs de précision de 81.76% et 81.46% respectivement.

En plus de cette évaluation de la précision, il est important d'évaluer celle-ci sur le classement des données positives et négatives tels que la sensibilité (taux de vrais positifs) et la spécificité (taux de vrais négatifs). Or, ce qui nous intéresse le plus dans ce projet est de mesurer la précision du classement des clients qui sont susceptibles de churner (donc la sensibilité) et les taux de sensibilité des algorithmes dans notre cas sont assez-similaires à l'exception de celui de l'arbre de décision qui est très faible. Du coup, ce qui va trancher pour le choix du bon modèle est le taux de sa précision. Donc, le meilleur modèle pour la prédiction du churn dans l'entreprise est les forêts aléatoires.

Les principales caractéristiques qui contribuent à prédire le taux du churn ont été classées à l'aide de la mesure du gain d'information. Plus la valeur du gain est élevée plus la variable est importante dans la prédiction du churn. Les caractéristiques importantes selon l'algorithme des forêts aléatoires sont représentées dans la figure suivante :

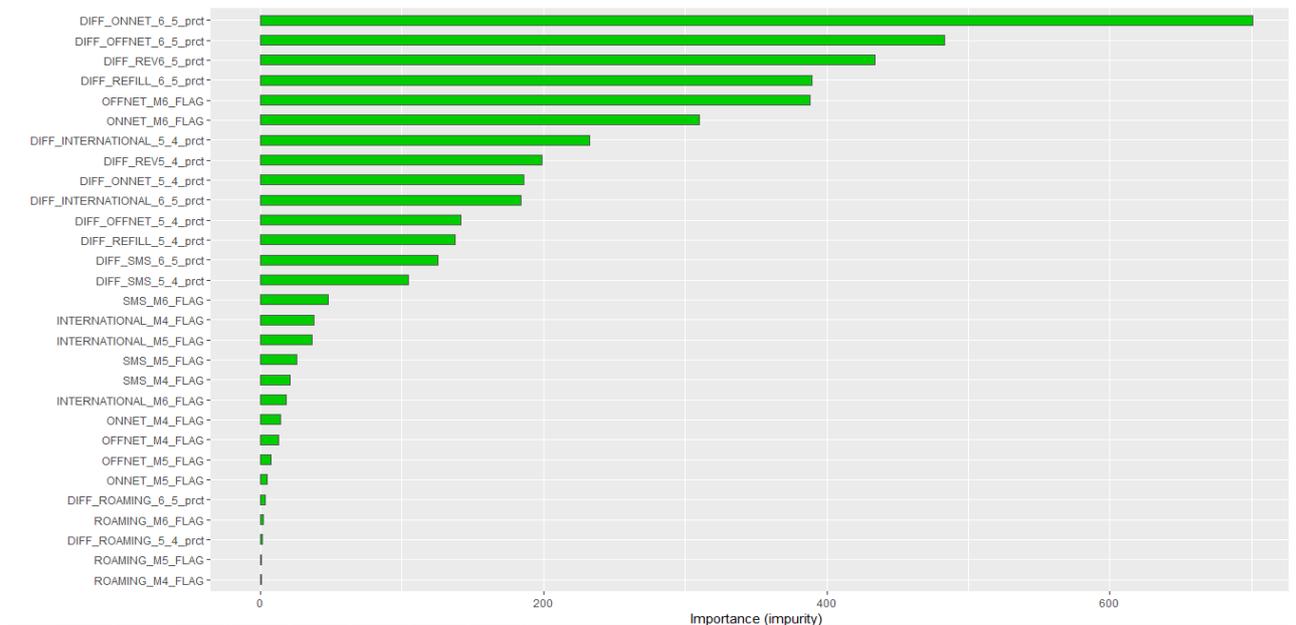


Figure IV.12 : Graphique des variables importantes par ordre croissant.

En termes de temps d'entraînement, les performances des algorithmes d'apprentissage automatique sont similaires à l'exception des réseaux de neurones qui sont plusieurs fois plus lents que les autres en raison de leur nature itérative.

## IV.5. Implémentation du meilleur modèle dans notre étude

Le modèle développé est un module qui sera intégré à un système global pour prédire la probabilité de churn pour chaque client.

Alors dans notre cas, nous avons créé une nouvelle colonne qu'on a nommé « score » qui va calculer pour chaque client sa probabilité de quitter l'entreprise.

Dans la figure suivante, on a effectué une requête sql pour voir le numéro de service du client et sa probabilité de cherner exprimée dans la colonne « score » et voici un aperçu des résultats obtenus :

| Service_Number | score       |
|----------------|-------------|
| 416512140      | 0.020000000 |
| 366245005      | 0.171428571 |
| 175106212      | 0.104285714 |
| 118145780      | 0.000000000 |
| 374789615      | 0.044285714 |
| 175389288      | 0.011428571 |
| 105826686      | 0.008571429 |
| 161915250      | 0.008571429 |
| 171126518      | 0.010000000 |
| 103646903      | 0.007142857 |
| 133104504      | 0.030000000 |
| 106568459      | 0.038571429 |
| 169148697      | 0.130000000 |
| 172760533      | 0.001428571 |
| 149833659      | 0.004285714 |
| 445067550      | 0.000000000 |
| 177442548      | 0.408571429 |
| 166761096      | 0.074285714 |
| 625052021      | 0.374285714 |
| 559664612      | 0.000000000 |
| 467096487      | 0.055714286 |
| 1298546497     | 0.328571429 |

Figure IV.13 : Aperçu des données de la colonne « score ».

## IV.6. Perspectives

La prédiction du churn est en train de devenir une des sources qui génèrent des revenus aux entreprises de télécommunications et à renforcer les relations avec les clients en prévenant quand ces derniers vont cesser leur activité avec l'entreprise ce qui ouvre la possibilité de renégocier avec le client et de le fidéliser. Afin de les retenir, l'entreprise devraient mettre quelques stratégies comme :

- Récompenser ses clients pour leur fidélité et atténuer leur insatisfaction grâce à l'utilisation d'un cadeau automatisé ou de stratégies de tarifications.
- Le traitement du langage naturel pourrait être utilisé pour effectuer une analyse thématique des thèmes les plus communs identifiés dans les appels de service à la clientèle.

## IV.7. Conclusion

Dans ce chapitre, nous avons présenté notre base de données sur laquelle nous avons testé notre système de prédiction du churn. Nous avons exposé nos résultats expérimentaux ainsi que quelques solutions pour améliorer ce travail.

## *Conclusion générale*

Notre étude qui s'est porté sur « l'utilisation d'une méthode d'apprentissage supervisé pour la détection de l'attrition d'un client » au sein de Djezzy nous a permis de toucher à un domaine de l'intelligence artificielle qui est l'apprentissage automatique, de connaître quelques techniques et algorithmes de celui-ci, aussi nous avons implémenté quelques algorithmes de l'apprentissage supervisé pour notre étude ce qui nous a permis de développer nos connaissances en ce domaine.

L'importance de ce type d'étude sur le marché des télécommunications est d'aider les entreprises à accroître leurs bénéfices. On sait maintenant que la prédiction de l'attrition des clients est l'une des sources de revenus les plus importantes pour ce type d'entreprises. Par conséquent, cette recherche visait à construire un système qui prédit le départ des clients dans la société Djezzy.

Nous souhaitons à l'avenir utiliser le traitement du langage naturel afin de prédire le churn des clients dans le secteur des télécommunications en analysant les mots et les phrases répétés dans les appels au service client pour classifier leur réclamations en thèmes (problème de facturation, d'offre, ou de couverture réseau,...) ou encore extraire certains sentiments qui indiqueraient la probabilité que le client reste encore un an ou plus ou bien quitterait bientôt l'entreprise. Dans ce cas, là encore, en utilisant le traitement du langage naturel, l'entreprise pourrait former un système à détecter les clients à risque et à répondre à leurs besoins profonds afin de se protéger contre la rétention des clients ou prévoir un système de messagerie automatique qui pourrait envoyé au client un e-mail personnalisé et convivial reconnaissant les problème que le client a eu tout en lui notifiant une récompense.

# *Bibliographie*

## Bibliographie

[1] : <https://emerj.com/ai-glossary-terms/what-is-machine-learning/>.

[2] : <https://www.itpedia.nl/fr/2018/04/05/wat-is-machine-learning/>.

[3] : [http://elearning.univjijel.dz/elearning/pluginfile.php/4333/mod\\_resource/content/1/SupportCours\\_Mokhtar\\_Taffar\\_ApprAuto.pdf](http://elearning.univjijel.dz/elearning/pluginfile.php/4333/mod_resource/content/1/SupportCours_Mokhtar_Taffar_ApprAuto.pdf).

[4] : <https://tel.archives-ouvertes.fr/tel-00335313v2/document>.

[5] : F.Souam Ait El Hadj, cour fouille de données, Master I, CPI, UMMTO 2018.

[6] : Document interne à l'entreprise.

[7] : <https://www.mpttn.gov.dz/fr/content/djezzy>.

<http://chercheurs.lille.inria.fr/pgermain/publis/memoire.pdf>

[http://eric.univ-lyon2.fr/~jahpine/cours/m2\\_dm-ml/cm.pdf](http://eric.univ-lyon2.fr/~jahpine/cours/m2_dm-ml/cm.pdf).

[http://www.dmi.usherb.ca/~larocheh/cours/ift725\\_A2014/diapositives/ift615-apprentissage-automatique.pdf](http://www.dmi.usherb.ca/~larocheh/cours/ift725_A2014/diapositives/ift615-apprentissage-automatique.pdf).

<http://www.math.univ-angers.fr/~labatte/enseignement%20UFR/master%20MIM/classificationsupervisee.pdf>.

<http://www.math.univ-angers.fr/~labatte/enseignement%20UFR/master%20MIM/classificationsupervisee.pdf>.

<https://www.math.u-psud.fr/~arlot/enseign/2009Centrale/cours-classif.pdf>.

<https://www.r-project.org/other-docs.html>.

[https://www.researchgate.net/figure/Random-forest-model-Example-of-training-and-classification-processes-using-random\\_fig5\\_280533599](https://www.researchgate.net/figure/Random-forest-model-Example-of-training-and-classification-processes-using-random_fig5_280533599).

KALACHE Soumia, KOULOUGHLI Asma, 02-juillet-2013, "Les forêts aléatoires floues", mémoire de fin d'études.

Laurent Rouvière : "Introduction aux méthodes d'agrégation, boosting, bagging et forêts aléatoires".

G.M. ApurvaSree, S. Ashika, S. Karthi, V. Sathesh, M. Shankar, J. Pamina, Mar-Apr 2019 :  
"Churn Prediction in Telecom using".

T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, K. Ch. Chatzisavvas, February 2011 : A  
"Comparison of machine learning techniques for customer churn prediction".

Abdelrahim Kasem Ahmad, Assef Jafar, Kadan Aljoumaa, 20 March 2019: "Customer churn  
"Prediction in telecom using machine learning in big data platform".