REPUBLIQUE DU CAMEROUN
Paix-Travail-Patrie

UNIVERSITE DE YAOUNDE I

CENTRE DE RECHERCHE ET FORMATION DOCTORALE EN SCIENCE, TECHNIQUE ET GEOSCIENCE

UNITE DE RECHERCHE ET FORMATION DOCTORALE EN SCIENCE DE L'INGENIEURE ET APPLICATIONS

REPUBLIC OF CAMEROUN
Peace-Work-Fatherland

UNIVERSITY OF YAOUNDE I

POSTGRADUATE SCHOOL OF SCIENCE, TECHNOLOGY AND GEOSCIENCE

RESEARCH AND POSTGRADUATE TRAINING UNIT IN ENGINEERING AND ITS APPLICATION

LABORATOIRE : ENERGIE-EAU-ENVIRONNEMENT
LABORATORY : ENERGY- WATER - ENVIRONMENT

ETUDE DE LA FIABILITE A BASE DES TECHNIQUES DE L'INTELLIGENCE ARTIFICIELLE

Mémoire de Master Recherche / Research Master

Présenté et soutenu par :

NZEGANG Frantz Dunant

En vue de l'obtention du

Diplôme de Master recherche en Science de l'Ingénieur

Option : Energétique

Sous la direction de :

Paul Salomon NGOHE EKAM, Maitre de Conférences

Devant le Jury composé de :

Président : Joseph KENFACK, Maitre de Conférences

Rapporteur : Paul Salomon NGOHE EKAM, Maitre de Conférences

Examinateur : Joseph VOUFO, Chargé de cours

DÉDICACE

A Arlette, et Elouan pour les prières, l’amour, la patience

REMERCIEMENTS

Je remercie grandement L'Eternel Dieu qui rend toute chose possible et le Seigneur JESUS-CHRIST pour son soutien tout au long de ce travail.

Je remercie le Professeur Joseph KENFACK, Maître de Conférences, pour l'insigne honneur qu'il nous fait en présidant ce Jury.

Je remercie également le Docteur Joseph VOUFO, Chargé de cours*, qui a accepté de nous faire l'honneur d'examiner ce travail.*

Je tiens à exprimer ma profonde gratitude au Professeur Paul Salomon NGOHE EKAM, Maitre de conférences, chef du département Génie Electrique et Télécommunications et membre du Laboratoire Eau Energie et Environnement où se sont déroulés nos travaux de recherche, pour avoir accepté de suivre et d'encadrer le présent travail, pour sa rigueur, et orientations qui ont contribué à l'aboutissement de ce mémoire.

Je remercie aussi les Docteurs Eric DEUSSOM et Aurelle TCHANIA pour les discussions et échanges que nous avons menées ensemble, pour la motivation et pour l'aide dans la phase pratique de ce mémoire

Je tiens aussi à exprimer ma gratitude à ma bien aimée Arlette Syntiche BILOA NDZOMO, pour ses prières, son soutien, ses constants encouragements qui m'ont permis de relever la tête dans les moments de découragement. Je remercie aussi mon fils Elouan pour l'amour qu'il m'a donné durant ce travail.

J'envoie aussi de chaleureux remerciements à mes chers parents ; M. et Mme NZEGANG, à mes beaux-parents M. et Mme NDZOMO, mes frères Yvan, Léonel, Clément, Thierry, Nathanael et mes sœurs Edith, Alvine, Perpétue, Bouquet, pour le soutien, l'amour, les encouragements.

Merci pour votre amour, votre patience et votre aide durant ce travail.

LISTE DES ABREVIATIONS

AI : Artificial Intelligence

AMDEC : Analyse des Modes de Défaillances, de leurs Effets et de leur Criticité

CAMTEL : Cameroon Telecommunications

CNN : Convolutional Neural Network

CV : Computer Vision

DL : Deep Learning

FMEA : Failure modes and effects analysis

IA : Intelligence Artificielle

IDE : Integrated Development Environment

KNN : K-Nearest Neighbors

ML : Machine Learning

OHE : One Hot Encoding

OR : Object Recognition

RL : Reinforcement Learning

RNN : Recurent neural network

SVM : Support Vector Machine

UL : Unsupervised Learning

SL : Supervised Learning

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

L'intelligence artificielle est aujourd'hui considérée comme la nouvelle électricité des sciences. Les secteurs dans lesquels elle intervient sont de plus en plus nombreux et variés. De l'informatique à la médecine en passant par l'industrie, ses applications sont sans cesse croissantes.

Les premières applications de l'IA se sont beaucoup plus orientées dans le domaine de la santé, et nous voyons son impact positif dans ce domaine. Ces prouesses de l'IA dans ce domaine nous ont conduit donc à nous interroger sur l'impact que l'IA pourrait avoir sur la maintenance industrielle en Afrique en général et au Cameroun en particulier, quand nous connaissons tous les problèmes auxquels notre industrie fait face en maintenance industrielle (pannes à répétition, temps de maintenance trop long, long temps d'arrêts, faible fiabilité, coût élevé de la maintenance).

Pour ce faire nous avons tout d'abord élaboré une méthodologie qui permettrait à toute personne d'utiliser l'IA pour résoudre son problème. Puis, nous avons utilisé cette méthodologie pour résoudre un cas réel, celui de la prédiction d'une défaillance sur le réseau de transport optique du Cameroun.

Ainsi, nous avons procédé à la collecte des données devant être utilisées pour résoudre notre problème. Ensuite, nous avons procédé au nettoyage de ces données c'est-à-dire à l'élimination des données non pertinentes pour la résolution de notre problème pour ne laisser que celle qui sont utiles, et incontournables pour effectuer notre prédiction.

Et c'est sur la base de tout ceci que nous avons écrit l'algorithme qui nous permettrait de prédire un incident sur le réseau de transport optique du Cameroun.

Les différentes étapes permettant de faire cette prédiction ont été détaillées et expliquées. Les codes du programme d'apprentissage automatique pour la résolution du problème écrits avec le logiciel Anaconda et son IDE Jupyter notebook ont aussi été présentés.

Au bout de ce travail, nous avons fait la prédiction des pannes avec une précision de 65%. Ces pannes correspondant à l'indisponibilité du signal réseau de télécommunications à fibre optique chez ses abonnés.

Mots clés : Intelligence Artificielle, Apprentissage automatique, Apprentissage profond, fibre optique, Anaconda, Jupyter notebook

ABSTRACT

Artificial intelligence is now considered the new electricity of science. The sectors in which it operates are more and more numerous and varied. From IT to medicine and industry, its applications are constantly growing.

The first applications of AI were much more oriented in the field of health, and we see its positive impact in this field. These prowess of AI in this field has therefore led us to wonder about the impact that AI could have on industrial maintenance in Africa in general and in Cameroon in particular, when we know all the problems that our industry faces. face in industrial maintenance (repeated breakdowns, maintenance time too long, long downtime, low reliability, high cost of maintenance).

To do this, we first developed a methodology that would allow anyone to use AI to solve their problem. Then, we used this methodology to solve a real case, that of the prediction of a failure on the optical transport network of Cameroon.

Thus, we proceeded to collect the data to be used to solve our problem. Then, we proceeded to the cleaning of these data, that is to say to the elimination of the irrelevant data for the resolution of our problem to leave only that which are useful, and essential to carry out our prediction.

And it is on the basis of all this that we wrote the algorithm that would allow us to predict an incident on the optical transport network of Cameroon.

The different steps to make this prediction have been detailed and explained. The machine learning program codes for solving the problem written with Anaconda software and its Jupyter notebook IDE were also presented.

At the end of this work, we made the prediction of failures with an accuracy of 65%. These failures correspond to the unavailability of the fiber optic telecommunications network signal to its subscribers.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, fiber optics, Anaconda, Jupyter notebook

INTRODUCTION GENERALE

Grâce aux progrès scientifiques et techniques, plusieurs domaines ont pu naître et connaitre un essor croissant. Ainsi, les domaines tels que la santé, le transport, l'informatique ont vu le jour et ont grandement contribué à l'épanouissement de l'homme.

Ces progrès scientifiques ont permis de créer des machines, que l'homme a utilisées pour subvenir à ses besoins de production et de consommation. Mais très vite, l'homme s'est trouvé face à un sérieux problème : celui d'assurer le fonctionnement permanent de ses équipements ou alors à défaut, réduire au maximum, les temps d'arrêt, les délais de maintenance, et augmenter la fiabilité de ses équipements.

Plusieurs méthodes ont alors vue le jour : l'AMDEC, la FMEA, la méthode LEAN SIX SIGMA, etc…chacune apportant un plus par rapport aux autres, mais présentant aussi des limites. L'évolution de la technologie a fait naître un domaine nouveau : celui de l'intelligence artificielle.

Ces dernières années, grâce aux avancées de la recherche, l'intelligence artificielle a connu un regain, et sa popularité s'est accrue grâce aux prouesses réalisées par des intelligences artificielles. A partir des années 2011, l'on a vu l'utilisation de l'IA, d'abord dans les jeux, ensuite et de manière croissante dans la médecine ; et depuis lors, les travaux ne cessent d'aller dans ce sens, que ce soit la détection des cancers, des maladies chroniques, rien n'échappe à l'IA.

Face à tout ceci, l'Afrique traine le pas. Que ce soit dans son industrie ou dans le domaine de la santé, l'Afrique en général et la Cameroun en particulier accuse encore un énorme retard. Son industrie fait face à de nombreux problèmes tels que la recrudescence des défaillances, de trop longs délais de maintenance, ou encore des coûts de maintenance élevés.

La venue de l'intelligence artificielle sonne une nouvelle ère pour l'industrie camerounaise et africaine, car, elle a permis d'améliorer le quotidien, permettant même de réaliser des tâches qui dans le passé semblaient encore impossibles à réaliser. Ces dernières années il est indéniable que l'intelligence artificielle a eu un très grand impact en médecine en sauvant des vies, par la détection précoce des maladies, et même par le choix du meilleur traitement à adopter pour soigner un malade. Mais cette intelligence artificielle tant utilisée pour le diagnostic et le traitement d'un patient, ne peut-elle pas être utilisée en industrie pour le diagnostic des systèmes de production afin d'accroitre leur efficacité ?

Ainsi, notre travail intitulé « Etude de la fiabilité à base des techniques d'intelligence artificielle » a pour objectif de montrer comment utiliser les outils de l'intelligence artificielle en industrie pour la détection rapide ou la prédiction des incidents afin d'entreprendre le plus rapidement possible les actions requises permettant d'augmenter sa fiabilité.

La question est donc de savoir : Comment utiliser l'IA pour diagnostiquer les pannes, et prédire un dysfonctionnement sur un système, afin non seulement de réduire le temps d'arrêt, mais aussi d'augmenter le temps de bon fonctionnement ?

Notre travail est organisé en trois chapitres. Le premier porte sur l'IA ; dans cette partie, nous présentons l'IA, un historique bref de l'IA, ses différentes composantes et les différents types algorithmes qui sont utilisés pour réaliser l'apprentissage en IA. Le second chapitre porte sur la méthodologie de la résolution d'un problème avec l'intelligence artificielle. Ici, nous proposons une démarche à suivre pour résoudre un problème avec l'IA, en utilisant particulièrement l'apprentissage automatique. Le troisième enfin, porte sur l'expérimentation de la méthodologie décrite au second chapitre, en utilisant un algorithme spécifique et pour un cas pratique bien défini.

CHAPITRE I : L'INTELLIGENCE ARTIFICIELLE

Introduction

Au cours des dernières années, l'intelligence artificielle a eu un impact considérable sur notre quotidien. Plusieurs tâches auparavant considérées comme impossibles à réaliser par les ordinateurs le sont aujourd'hui grâce à elle. Auparavant une machine ne pouvait exécuter une tâche qu'après avoir été explicitement programmée. L'avènement de l'intelligence artificielle a changée toute la donne : Grâce à elle, une machine peut exécuter une tâche, prédire l'avenir sans avoir été explicitement programmée. Que ce soit la médecine, le sport, la littérature, la musique, l'industrie, son implication s'y est fait ressentir et aucun domaine ne lui échappe. Il faut reconnaitre que son avènement a considérablement simplifié nos tâches, fait évoluer nos mentalités, bref amélioré notre quotidien.

L'intelligence artificielle se compose de nombreux domaines différents (figure 1). Généralement, il y a trois domaines majeurs en IA :

- Le Symbolic Learning ou Apprentissage symbolique ;
- Le Machine Learning ou Apprentissage automatique ;
- Le Deep Learning(DL) ou Apprentissage Profond.

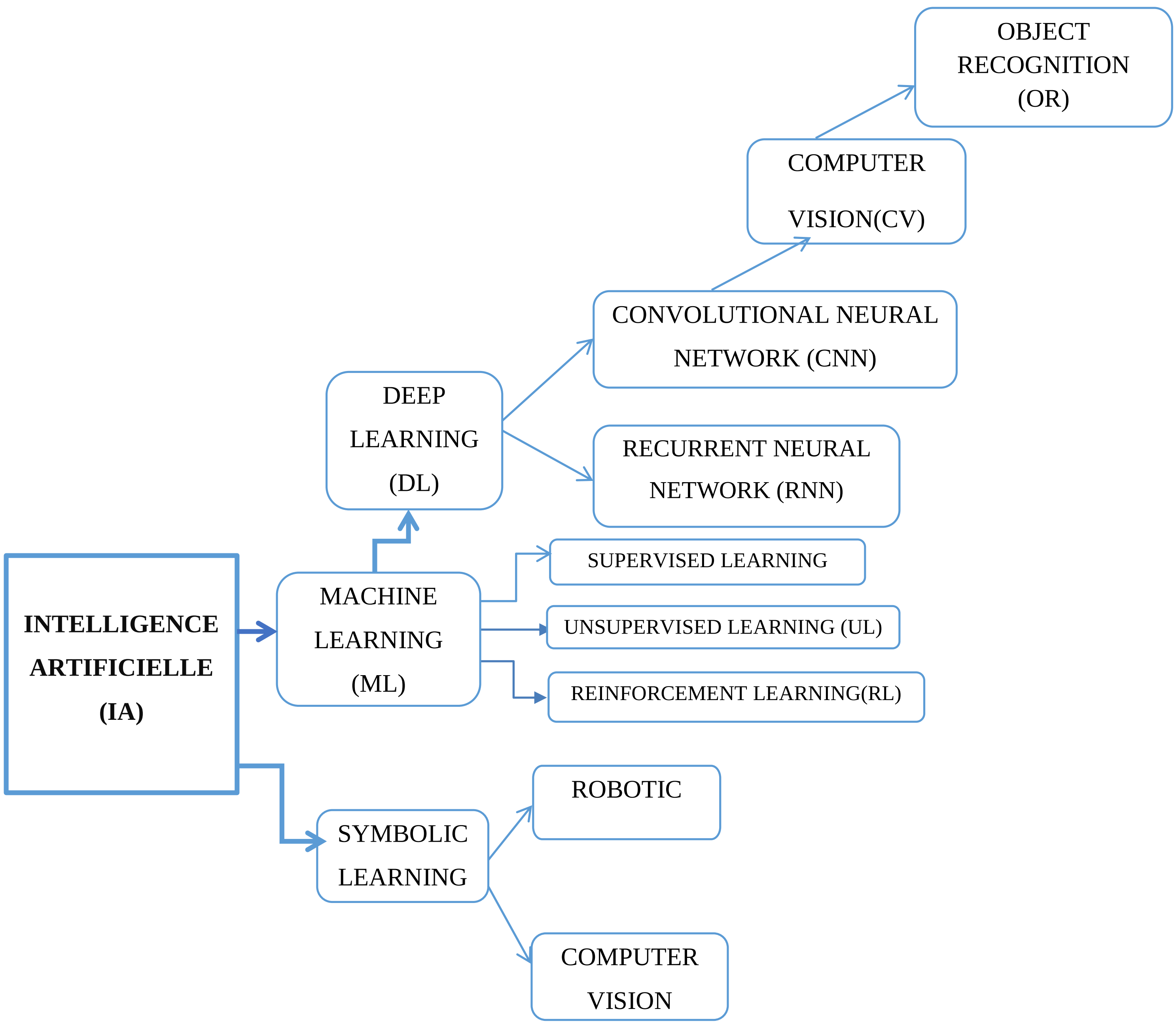


Figure 1: Domaines de l'Intelligence Artificielle

I-LE SYMBOLIC LEARNING OU APPRENTISSAGE SYMBOLIQUE

Aussi appelée Intelligence Artificielle Symbolique, c'est une théorie proposée pour rendre compte de l'efficacité de l'imagerie, qui suggère que l'imagerie aide à développer un schéma mental en créant un programme moteur dans le système nerveux central.

Ce que nous devons savoir, c'est que l'IA symbolique, est l'ancêtre de ce que nous appelons aujourd'hui le Machine Learning. Ce type d'IA ne nécessite pas de formation, de conjectures, de quantité massive de données. Il est basé sur l'utilisation de symboles, puis sur la logique pour rechercher des solutions. Tout ce que nous avons à faire est donc de représenter l'univers entier qui nous intéresse sous forme de symboles dans un ordinateur. Il repose sur trois concepts principaux :

- Les humains pensent en utilisant des symboles.
- Les ordinateurs fonctionnent à l'aide de symboles.
- Les ordinateurs peuvent être entraînés à penser

En définitive, l'apprentissage symbolique consiste à utiliser des symboles, plutôt que des statistiques, pour résoudre un problème particulier et proposer une solution. Il est également devenu la base de la création de l'IA et a été le précurseur dans la création de la technologie, c'est pourquoi il est également appelé par beaucoup de gens "IA classique" ou "bonne IA à l'ancienne". L'IA symbolique, comme son nom l'indique, fonctionne avec le principe de la représentation. Une représentation peut être définie comme un symbole interne dans l'esprit qui représente une réalité externe par association, convention ou ressemblance.

Les approches symboliques en intelligence artificielle représentent des choses dans un domaine de connaissance à travers des symboles physiques, qui sont ensuite transformés en expressions et structures symboliques, afin de modéliser l'esprit avec des représentations.

Même si l'IA symbolique est « ancienne », elle est toujours utilisée par les scientifiques et a de nombreuses applications dans la société d'aujourd'hui. On peut citer le NLP (Traitement du Langage Naturel) est une branche de l'IA qui permet aux machines d'analyser le langage humain, permettant aux gens de communiquer avec eux. C'est l'une des nombreuses utilisations de l'IA symbolique, pour les chatbots conversationnels.

II- LE MACHINE LEARNING OU APPRENTISSAGE AUTOMATIQUE

L'un des domaines de l'IA qui est en plein essor est sans aucun doute le Machine Learning ou ML. Presque tout peut être associé au Machine Learning: Marketing, e-business, ingénierie, etc…

L'apprentissage automatique selon Arthur Samuel, est une science qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés [5] ; ce qui signifie que nous offrons aux machines la possibilité d'apprendre à faire quelque chose pour lequel elles n'ont pas été programmées.

Ainsi, les objectifs du Machine Learning sont de permettre aux ordinateurs d'apprendre à faire quelque chose par eux-mêmes. Par exemple, un ordinateur peut être entrainé à classer les mails et à dire s'il s'agit d'un spam ou non.

- Cette science peut être divisée en trois grands domaines.[6]Le Supervised Learning ou Apprentissage supervisé
- Unsupervised Learning ou Apprentissage non supervisé
- Le Reinforcement Learning ou Apprentissage par renforcement.

II-1. Le supervised learning

C'est le domaine du ML où un modèle est capable de prédire à l'aide d'un ensemble de données "étiquetées". Pour faire plus simple, nous apprenons à la machine en utilisant des données qui sont "étiquetées". Comment est-ce possible? Pour comprendre, prenons un exemple. Nous avons quelques végétaux : pastèque, melon, pommes, petits pois.

Nous apprenons à la machine à connaître chaque végétal et chacun d'eux est "étiqueté". En fait, nous entraînons la machine à apprendre à quoi ressemble chaque végétal (en termes de couleurs et de forme). Une fois entraînée, la machine est alimentée avec de nouvelles données, et elle pourra prédire de quel végétal il s'agit. L'illustration est donnée à la figure 2:

Nous pouvons voir que dans l'apprentissage supervisé, nous utilisons des exemples étiquetés dans l'ensemble d'apprentissage pour que l'ordinateur apprenne à prédire les étiquettes pour de nouveaux exemples inédits. L'aspect supervisé fait référence à la nécessité pour chaque exemple d'apprentissage d'avoir une étiquette afin que l'algorithme apprenne à faire des prédictions précises sur les futurs exemples.

Dans l'apprentissage supervisé, nous utilisons des exemples étiquetés dans l'ensemble d'apprentissage pour que l'ordinateur apprenne à prédire les étiquettes pour de nouveaux exemples inédits. Nous parlons d'apprentissage supervisé car l'ordinateur doit être entraîné pour chaque exemple avec des ensembles de données étiquetés, afin d'apprendre à faire des prédictions précises sur les futurs exemples.

L'apprentissage supervisé peut être divisé en deux sous-catégories :

- La Classification
- La Régression

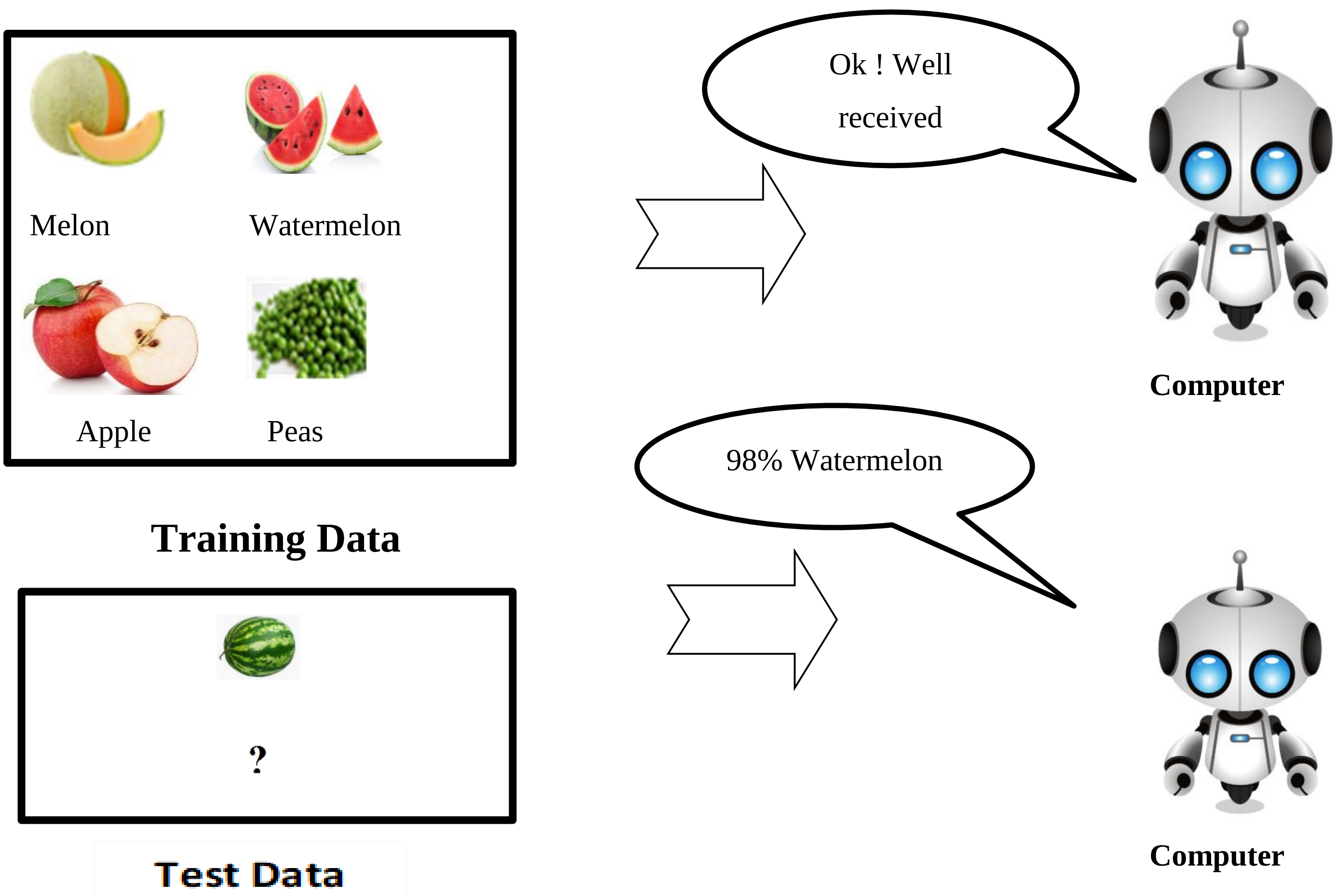


Figure 2: Processus simplifié du Supervised Learning : entrainnement des données [7]

II-1-1. La Classification

La classification est un type d'apprentissage dirigé qui prédit une classe pour une variable d'entrée et peut être effectuée sur des données structurées ou non structurées. Il permet de définir la classe à laquelle appartiennent les éléments de données et mieux utilisés lorsque la sortie a des valeurs finies et discrètes. Dans l'algorithme de classification, nous catégorisons un ensemble donné de données en classes qui sont souvent appelées cibles, étiquette ou catégories. Le processus commence par la prévision de la classe des points de données donnés. Un exemple simple de classement est le classement des mails pour nous dire s'il s'agit d'un spam ou non.

Il existe de nombreux algorithmes de classification [8]:

- Le Logistic Regression
- Le Naïve Bayes
- Le DecisionTree
- Le K-NearestNeighbours
- Le Support Vector Machine
- Etc…

II-1-2. La Regression

Un problème de régression survient lorsque la variable de sortie est une valeur réelle, telle que « livres » ou « poids ». La régression est la tâche de prédire une quantité continue. Il peut également prédire une valeur discrète, mais la valeur discrète sous la forme d'une quantité entière. Fondamentalement, la régression dans l'apprentissage automatique est une approche statistique utilisée pour trouver la relation entre les variables. L'objectif est de prédire l'issue d'un événement sur la base de la relation entre les variables obtenues à partir de l'ensemble de données.

Il existe plusieurs types de regression [9] :

- Régression linéaire simple
- La régression linéaire multiple
- Régression linéaire polynomiale

Avantages de l'apprentissage supervise [10]

- Il vous aide à résoudre divers types de problèmes de calcul du monde réel.
- Les données d'entrée sont connues et étiquetées.

- La connaissance des données d'entrée et leur étiquetage rendent les résultats plus précis et fiables que ceux de l'apprentissage non supervise
- En apprentissage supervisé, les classes utilisées sont connues, donc les réponses dans l'analyse et la sortie de votre algorithme sont susceptibles d'être connues.
- Dans l'apprentissage supervisé, vous pouvez collecter des données ou produire une sortie de données à partir de l'expérience précédente
- Aide à optimiser les critères de performance grâce à l'expérience
- Permet d'entraîner l'algorithme à distinguer différentes classes où vous pouvez définir une limite de décision idéale.

❖ Inconvénients de l'apprentissage supervisé [11]

- Nous savons que, dans l'apprentissage supervisé, les données doivent être étiquetées. La classification des mégadonnées peut donc être très difficile.
- La nécessité de bien comprendre et étiqueter les entrées dans l'apprentissage supervisé peut le rendre plus complexe par rapport à la méthode non supervisée
- La formation pour l'apprentissage supervisé nécessite beaucoup de temps de calcul.
- Il ne se déroule pas en temps réel alors que l'apprentissage non supervisé concerne le temps réel. Pendant ce temps, l'apprentissage supervisé utilise l'analyse hors ligne.
- La limite de décision peut être surentraînée si votre ensemble d'entraînement n'a pas d'exemples que vous souhaitez avoir dans une classe
- Nécessite la sélection de beaucoup de bons exemples dans chaque classe pendant que l'entrainement du classificateur.
- Si vous avez des données dynamiques volumineuses et croissantes, vous n'êtes pas sûr des étiquettes pour prédéfinir les règles. Cela peut être un vrai défi.

❖ Applications d'apprentissage supervise [12]

L'apprentissage supervisé a des applications dans beaucoup de domaines parmi lesquels :

- La détection de spam par e-mail (spam, pas spam).
- L'analyse des sentiments (heureux, pas heureux), en marketing, en utilisant un algorithme de textmining.
- Détection de fraude par carte de crédit (fraude, pas fraude) dans le secteur bancaire.

II-2. L'apprentissage non supervisé

Dans le cas d'un algorithme d'apprentissage non supervisé, les données ne sont pas explicitement étiquetées dans différentes classes. Le modèle est capable d'apprendre des données en trouvant des modèles implicates [13]. Les algorithmes d'apprentissage non supervisé identifient les données en fonction de caractéristiques similaires telles que des structures, des segments similaires, etc. À titre d'exemple, le système de recommandation d'un site Web de commerce électronique où l'algorithme d'apprentissage découvre des articles similaires souvent achetés ensemble.

L'un des algorithmes importants qui relève de l'apprentissage non supervisé est le Clustering.

C'est un concept important dans l'apprentissage non supervisé. Il s'agit principalement de trouver une structure ou un modèle dans une collection de données non catégorisées. Le principe est de traiter les données et de trouver des groupes (d'où le nom de cluster) s'ils existent dans ces données. Il est également possible de modifier le nombre de clusters que vos algorithmes doivent identifier, ou d'ajuster la granularité de ces groupes.

Le clustering peut être appliqué dans le domaine de la segmentation du marché : une entreprise peut utiliser l'analyse de cluster pour créer différents groupes de clients en fonction de leur comportement d'achat.

Le clustering peut être divisé en sous-champs [14] :

- K-Means Clustering
- Hierarchical Clustering
- Probabilistic Clustering
- Association

Les règles d'association permettent d'établir des associations entre des objets de données à l'intérieur d'un grand ensemble de données. Cette technique non supervisée consiste à découvrir des relations intéressantes entre les variables dans de grandes bases de données. Par exemple, les personnes qui achètent une nouvelle maison sont plus susceptibles d'acheter de nouveaux meubles.

En association, l'objectif est de découvrir des relations intéressantes entre les variables dans un grand ensemble de données, en établissant des associations entre les objets de données. À titre d'exemples dans le commerce électronique, nous pouvons regrouper les chopper en fonction de leurs historiques de navigation et d'achat.

- **Avantages de l'apprentissage non supervisé [15]**

Voici quelques avantages de l'apprentissage automatique non supervisé :

- Il trouve toutes sortes de modèles inconnus dans les données
- Cela aide à trouver des fonctionnalités qui peuvent être utiles pour la catégorisation.
- Dans l'apprentissage non supervisé, il est plus facile d'obtenir des données non étiquetées d'un ordinateur que des données étiquetées, qui nécessitent une intervention manuelle.
- IL se déroule en temps réel, donc toutes les données d'entrée doivent être analysées et étiquetées en présence des apprenants.

- **Inconvénients de l'apprentissage non supervisé [15]**

- Dans l'apprentissage non supervisé, les données d'entrée ne sont pas connues et étiquetées à l'avance par l'agent d'IA ; c'est la machine qui le fait elle-même. Ce qui a pour effet, d'avoir moins de précision des résultats
- L'utilisateur doit passer du temps à interpréter et étiqueter les classes qui suivent cette classification.
- Vous ne pouvez pas obtenir d'informations précises concernant le tri des données, et la sortie en tant que données utilisées dans l'apprentissage non supervisé est étiquetée et inconnue
- Les classes spectrales ne correspondent pas toujours à des classes informationnelles.
- Les propriétés spectrales des classes peuvent également changer au fil du temps, vous ne pouvez donc pas avoir les mêmes informations de classe lorsque vous passez d'une image à une autre.

II-3. Apprentissage par renforcement

L'apprentissage par renforcement est le domaine de l'apprentissage automatique qui traite de la prise de décision séquentielle, dans le but d'apprendre à l'ordinateur comment prendre des mesures afin de maximiser la récompense [16] dans l'apprentissage par renforcement, un système de récompenses et de pénalités est utilisé pour obliger l'ordinateur à résoudre les problèmes par lui-même. L'implication humaine se limite uniquement à changer l'environnement et à peaufiner le système de récompenses et de pénalités. Comme l'ordinateur maximise la récompense, il est enclin à chercher des moyens inattendus de le faire. L'objectif de l'implication humaine est de se concentrer sur l'empêchement d'exploiter le système et de motiver la machine à effectuer la tâche de la manière attendue. L'apprentissage par renforcement est utile lorsqu'il n'existe pas de « moyen approprié » d'effectuer une tâche, mais qu'il existe des règles que le modèle doit suivre pour exécuter correctement ses taches [16] .

- **Avantages de l'apprentissage par renforcement [17]**

- L'apprentissage par renforcement peut être utilisé pour résoudre des problèmes très complexes qui ne peuvent pas être résolus par des techniques conventionnelles.
- Si vous voulez obtenir des résultats à long terme, qui sont très difficiles à atteindre, l'apprentissage par renforcement est la solution.
- Cet algorithme est proche de la perfection, car il est très similaire à l'apprentissage des êtres humains
- Le modèle peut corriger les erreurs qui se sont produites pendant le processus de formation.
- Une fois qu'une erreur est corrigée par le modèle, les chances de se produire la même erreur sont très moindres.
- Si la seule façon de collecter des informations sur l'environnement est d'interagir avec lui, l'apprentissage par renforcement sera utile.
- Il peut créer le modèle parfait pour résoudre un problème particulier.
- Il peut être mis en œuvre par des robots pour leur apprendre à marcher.
- Les modèles d'apprentissage par renforcement peuvent surpasser les humains dans de nombreuses tâches.
- L'apprentissage par renforcement vise à obtenir le comportement idéal d'un modèle dans un contexte spécifique, afin de maximiser ses performances.

- **Inconvénients de l'apprentissage par renforcement [17]**

- Trop d'apprentissage par renforcement peut conduire à une surcharge d'états, ce qui peut diminuer les résultats.
- Non recommandé pour résoudre des problèmes simples.
- L'apprentissage par renforcement est gourmand en données. En fait, cet algorithme a besoin de beaucoup de données et de calculs, c'est pourquoi il est préférable de l'utiliser dans le jeu vidéo, où il y a beaucoup de données.
- De nombreux problèmes d'apprentissage par renforcement peuvent être résolus en utilisant une combinaison d'apprentissage par renforcement avec d'autres techniques telles que l'apprentissage en profondeur.

- L'apprentissage par renforcement suppose que le monde est markovien, ce qu'il n'est pas. Le modèle markovien décrit une séquence d'événements possibles dans laquelle la probabilité de chaque événement ne dépend que de l'état atteint lors de l'événement précédent.

III. LE DEEP LEARNING

Encore appelé apprentissage profond, le Deep Learning est un type d'intelligence artificielle dérivé du machine learning où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

Le deep learning s'appuie sur un réseau de neurones artificiels (s'inspirant du cerveau humain), et composé de dizaines, voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Sa force réside dans le fait qu'il s'améliore dans l'analyse des données, et peut se corriger quand une haute précision de prédiction est en jeu.

Un réseau de neurones Deep Learning a une couche d'entrée et une couche de sortie. Entre ces deux, il y a plusieurs "couches cachées" où les calculs sont effectués. Plus il y a de couches cachées, plus le réseau est « profond » et performant.

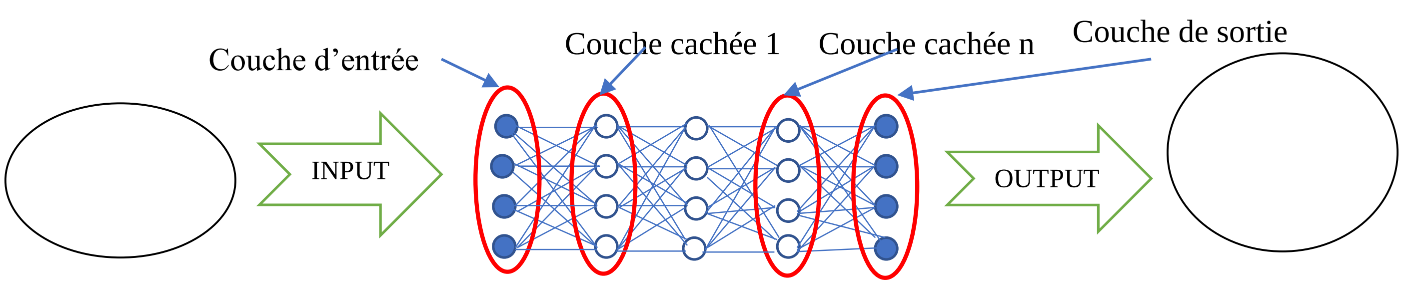


Figure 3 : matérialisation d'un réseau de neurones

Les applications d'apprentissage en profondeur sont utilisées dans des secteurs allant de la conduite automatisée aux dispositifs médicaux. En conduite automatisée, il permet de détecter automatiquement des objets tels que les panneaux de signalisation, les feux de circulation, la détection des piétons, tout ceci afin de réduire les accidents.

Il existe plusieurs types de réseaux de neurones profonds :

III-1. Les Réseau de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs (Convolutional Neural Network) sont à ce jour les modèles les plus performants pour classer des images, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).
Ce type de réseau de neurones est le mieux adapté à l'analyse d'images.

III-2. Réseau de neurones récurrents (RNN)

Les réseaux récurrents (ou RNN pour Recurrent Neural Networks) sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches.
Ils possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire et peuvent ainsi prendre en compte à un instant t un certain nombre d'états passés. Ces avantages font à ce qu'ils soient particulièrement adaptés aux applications faisant intervenir le contexte, et plus particulièrement au traitement des séquences temporelles comme l'apprentissage et la génération de signaux, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres.

III-3. Réseau contradictoire génératif (GAN)

Un GAN utilise deux réseaux de neurones différents pour en créer un nouveau. Un exemple de GAN dans la pratique est lorsque de nouvelles photographies sont générées à partir de données qui semblent réelles à l'œil humain.

III-4. Réseau de croyances profondes (DBN)

Les DBN ne sont pas aussi populaires que les autres types de réseaux de neurones. Ils ont été inventés comme une solution alternative pour aider à former les réseaux de neurones qui devenaient bloqués.

III-5. Comparaison Machine Learning , Deep Learning

Bien que le Machine Learning et le Deep Learning soient tous des domaines de l'Intelligence Artificielle, ils présentent des différences notables :

- Les algorithmes d'apprentissage automatique fonctionnent souvent bien même si l'ensemble de données est petit. Mais l'apprentissage en profondeur est gourmand en données, ce qui fait que plus vous avez de données, mieux il est susceptible de fonctionner. En effet, pendant que l'apprentissage en profondeur nécessite un grand ensemble de données pour éliminer les fluctuations et faire des interprétations de haute qualité, l'algorithme d'apprentissage automatique traditionnel pour fonctionner correctement, ne nécessite qu'un petit ensemble de données. Bien plus, avec un grand ensemble de données, en apprentissage automatique, on pourrait arriver au phénomène d'overfitting alors qu'avec le deep learning on aboutirait à une meilleure prédiction..
- Alors que les algorithmes d'apprentissage automatique traditionnels ont une structure assez simple, telle que la régression linéaire, l'arbre de décision, l'apprentissage en profondeur quant à lui est basé sur un réseau de neurones artificiels multicouche, comme un cerveau humain, lui permettant ainsi de résoudre les problèmes qui ne peuvent être résolus par le ML.
- Les algorithmes d'apprentissage en profondeur nécessitent beaucoup moins d'intervention Humaine, par ce que l'algorithme apprend de ses propres erreurs.
- Le deep learning est beaucoup plus gourmand en utilisation du CPU que le machine learning, ceci s'explique par le fait que le deep learning demande un grand ensemble de données pour être implémenté ; d'où la nécessité des processeurs puissants pour une meilleure implémentation.
- Contrairement au machine learning où l'ingénieur choisit lui-même les features, en deep learning, l'algorithme choisit lui-même les features.

IV. QUELQUES ALGORITHMES D'APPRENTISSAGE

En IA et précisément en ML il existe plusieurs algorithmes qui sont utilisés pour réaliser l'apprentissage.

Il est à noter que l'implémentation des algorithmes d'apprentissage automatique se fait grâce à un logiciel. Ceux couramment utilisés sont : MATLAB et ANACONDA. MATLAB est un puissant logiciel, très utilisé en ingénierie électrique, mécanique, informatique et même en intelligence artificielle. Le problème est que c'est un logiciel payant et il n'y a pas encore une communauté importante qui travaille en intelligence artificielle en l'utilisant.

ANACONDA par contre, est un logiciel libre, constitué de plusieurs bibliothèques qui sont sans cesse améliorées, en occurrence Spyder, Jupyter Notebook, etc… Au vu de ces différences notables, notre choix se portera sur ANACONDA et sa bibliothèque Jupyter Notebook. Dans la suite de ce travail, les codes et l'environnement de travail sont exécuté dans Jupyter Notebook.
Comme algorithmes d'apprentissage, On peut citer entre autres :

IV-1. L'arbre de Décision

L'arbre de Décision est un algorithme d'apprentissage supervisé, utilisé pour les tâches de classification et de régression, avec pour objectif de créer un modèle qui prédit la valeur d'une variable cible en apprenant des règles de décision simples (généralement sous la forme d'instructions if-then-else), déduites des caractéristiques des données.

En réalité, l'arbre de décision est un graphe en forme d'arbre avec des nœuds représentant l'endroit où nous choisissons un attribut et posons une question ; les bords représentent les réponses à la question, et les feuilles représentent la sortie réelle ou l'étiquette de classe. Plus l'arbre a des ramifications (branches), plus les règles sont complexes et plus le modèle est ajusté.

Les arbres de décision classent les exemples en les triant dans l'arborescence de la racine à un nœud feuille, le nœud feuille fournissant la classification à l'exemple. Chaque nœud de l'arborescence agit comme un cas de test pour un attribut, et chaque arête descendant de ce nœud correspond à l'une des réponses possibles au cas de test. Ce processus est de nature récursive et est répété pour chaque sous-arbre enraciné aux nouveaux nœuds.

Il présente un certain nombre d'avantages non négligeables à savoir :

- ✓ Il est facile à lire et à interpréter, et ceci, sans même nécessairement avoir des connaissances en statistiques.
- ✓ Il est Facile à élaborer
- ✓ Requière moins de nettoyage de données.

Illustrons cela à l'aide d'un exemple , celui de la maintenance d'un moteur.

Les moteurs tombent souvent en panne sans avertissement et lorsqu'ils le font, la principale préoccupation est de mettre en service les systèmes concernés le plus rapidement possible. Si un remplacement est facilement disponible, le moteur défaillant est généralement remplacé et envoyé pour réparation. Si aucun remplacement n'est disponible, la réparation est souvent la seule option.

La décision de réparer ou de remplacer un moteur défectueux a un effet direct sur les coûts d'exploitation et la fiabilité. Évaluer les moteurs et prendre des décisions de réparation ou de remplacement à l'avance évite les décisions réactives qui peuvent être coûteuses à long terme.

Inventaire des moteurs de rechange et politiques de disponibilité des moteurs de remplacement : garantir que les bons moteurs sont disponibles en temps opportun afin que la résolution la plus rapide d'une panne de moteur soit également la plus économique et la plus fiable.

Avec un programme de gestion des moteurs, les directeurs d'usine et de maintenance peuvent travailler avec les fournisseurs pour s'assurer que les moteurs sont disponibles en cas de besoin. L'arbre de décision peut permettre de résumer le processus de prise de décision lorsqu'un moteur tombe en panne.

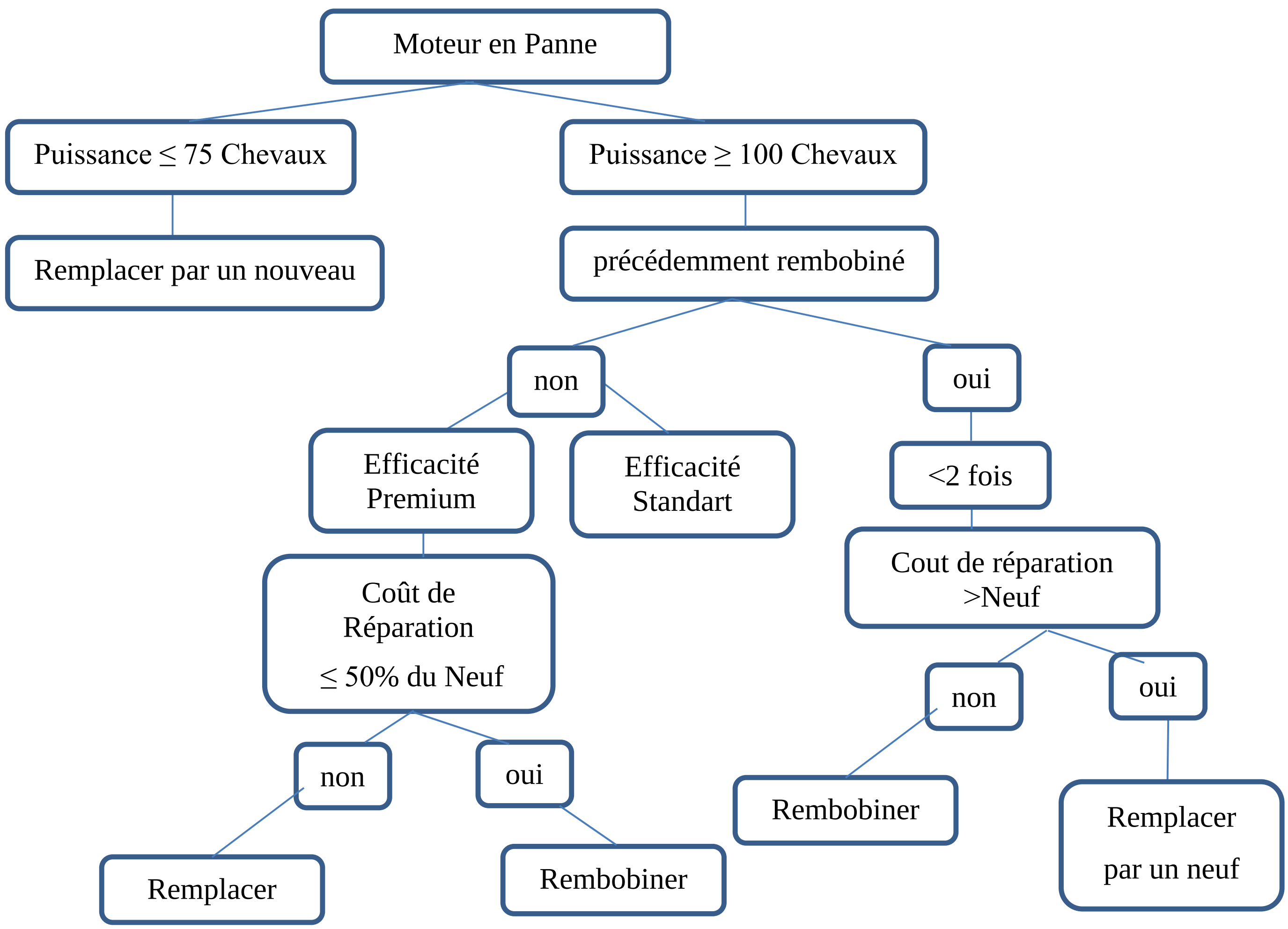


Figure 4 : Arbre de décision

La figure ci-dessus (Figure 4) illustre un arbre de décision issus des évènements connus ; Chaque nœud représentant une caractéristique et la branche de chaque nœud représentant le résultat de ce nœud. Ces branches sont terminées par des feuilles, qui représentent la décision finale prise.

Il peut arriver que les caractéristiques que nous possédons soient des valeurs continues. Dans ce cas, les nœuds internes peuvent tester la valeur d'une caractéristique par rapport à un seuil tel que le montre la figure ci-dessus.

Il est à noter que dans cet algorithme, la meilleure division est celle qui sépare deux étiquettes différentes en deux ensembles.

De manière générale, l'algorithme de l'arbre de décision peut être décrit comme suit :

- ✓ Choisir la meilleure fonctionnalité : c'est celle qui divise ou sépare au mieux les données.
- ✓ Se Poser la question pertinente (celle qui facilite la séparation des données).
- ✓ Suivre le chemin de réponse.
- ✓ Répéter la première étape Passer à l'étape jusqu'à obtention de la réponse.

Toutefois, malgré ses avantages, l'arbre de décision présente aussi des inconvénients et qui ne sont pas les moindres :

- ✓ Il est Instable : En effet l'ajout d'un nouveau point de données peut entraîner la régénération de l'arborescence globale et tous les nœuds doivent être recalculés et recréés.
- ✓ Il ne convient pas aux grands ensembles de données : en effet, si la taille des données est importante, un seul arbre peut devenir complexe et entraîner un sur apprentissage. Donc, dans ce cas, nous devrions utiliser une forêt aléatoire au lieu d'un seul arbre de décision.
- ✓ Les calculs numériques impliqués dans un arbre de décision consomment généralement beaucoup de mémoire.

IV-2. Le Support Vector Machine (SVM)

Les machines à vecteurs de support (SVM), sont un ensemble de méthodes d'apprentissage supervisé. Bien qu'adaptés pour la classification, la régression et la détection des valeurs aberrantes, ils sont principalement utilisés dans les problèmes de classification.

L'objectif de cet algorithme, est simple : Trouver un hyperplan dans un espace à N dimensions (N représentant le nombre d'entités) qui classe distinctement les points de données. En bref, bien placer la frontière entre deux catégories. Or séparer nos entités, par une droite ou un plan, est certes bien mais vague, puisqu'il existe plusieurs lignes droites, voir même une infinité (figure ci-dessous) qui peuvent séparer nos catégories. D'où la bonne question : laquelle choisir ?

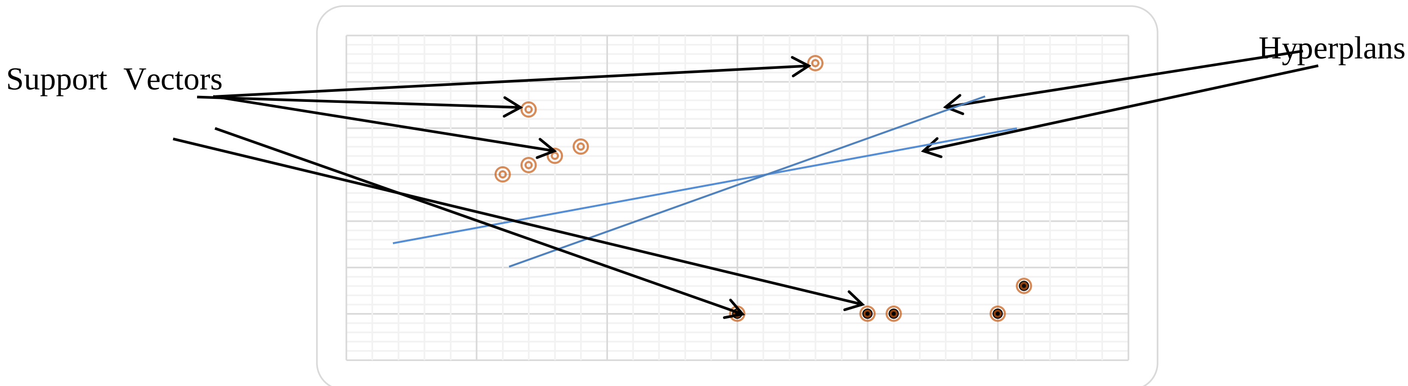


Figure 5: Modélisation des données dans le SVM

L'objectif du SVM est de créer la meilleure ligne ou limite de décision, ou encore Hyperplan, qui puisse séparer l'espace à n dimensions en classes distinctes, afin qu'il soit facile de placer le nouveau point de données dans la bonne catégorie à l'avenir.

Pour parvenir à cela, le SVM choisit les points/vecteurs extrêmes (appelés vecteurs de support) qui aident à créer l'hyperplan, 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, comme la figure ci-dessous nous le montre.

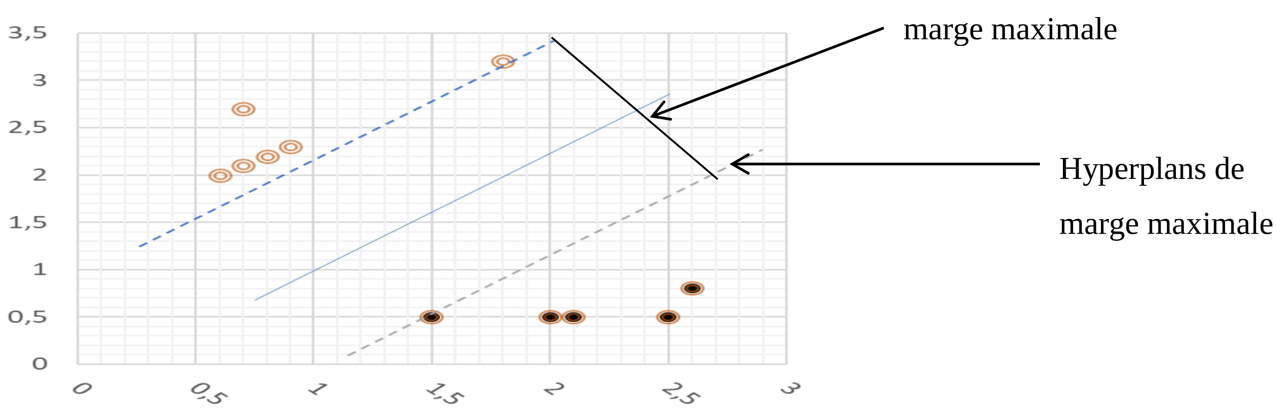


Figure 6: Matérialisation de l'hyperplan et de la marge dans le SVM

Toutefois, cet algorithme présente comme tout algorithme des avantages et des inconvénients à tenir en compte lors du choix d'un algorithme d'apprentissage supervisé.

❖ **Les Avantages du Support Vector Machine (SVM)**

Comme avantages nous pouvons citer : S'utilise très bien pour les dimensions élevées : en effet, Lorsque les données ont une dimension élevée (pensez à plus de 1000 fonctionnalités), une machine à vecteurs de support avec les bons paramètres (bon choix de noyau, etc.) peut permettre d'aboutir à des résultats vraiment précis.

✓ Sa flexibilité : Les SVM ont l'avantage de permettre de choisir en fonction du problème à résoudre le noyau adéquat, avec les bons paramètres.
✓ Fonctionne relativement bien lorsqu'il existe une marge de séparation claire entre les classes.
✓ Plus efficace dans les espaces de grande dimension.
✓ Efficace dans les cas où le nombre de dimensions est supérieur au nombre d'échantillons.
✓ Relativement peut gourmand en utilisation de la mémoire
✓ Prédiction rapide
✓ Ils peuvent être utilisés aussi bien en classification qu'en régression

❖ **Les Inconvénients du Support Vector Machine (SVM)**

Comme inconvénients nous pouvons citer :

✓ Convient aux petits ensembles de données : En effet, Les SVM n'ont pas une nature évolutive et ne fonctionnent pas très bien avec des ensembles de données de taille moyenne ou volumineuse.

✓ Dans certains cas, ils sont gourmands en mémoire : Les SVM sont coûteux en termes de calcul lorsqu'ils sont appliqués avec des noyaux non linéaires

✓ Choisir une « bonne » fonction du noyau n'est pas facile.

✓ Les hypers paramètres SVM sont difficiles à affiner, rendant la visualisation de leur impact complexe.

IV-3. L'algorithme du k-Nearest Neighbors (k-NN)

Aussi appelé l'algorithme des K-plus proches voisins, le KNN est un algorithme très simple, utilisé pour les problèmes de classification et de régression. Il est facile à comprendre, polyvalent et l'un des meilleurs algorithmes d'apprentissage automatique. Ces multiples avantages lui permettent d'être utilisé dans diverses applications telles que la finance, la santé, les sciences politiques, la détection de l'écriture manuscrite, la reconnaissance d'images et la reconnaissance vidéo.

- Par exemple, dans les cotes de crédit, les agences d'évaluation du crédit peuvent utiliser cet algorithme pour prédire la cote de crédit des clients.
- Dans les banques, lors du décaissement du prêt, les instituts bancaires peuvent utiliser cet algorithme pour prédire si le prêt est sûr ou risqué.
- En science politique, le KNN peut être utilisé pour classer les électeurs potentiels en deux classes : votera ou ne votera pas.

Cet Algorithme KNN basé sur une approche simple, celle de la similarité des caractéristiques.

KNN fonctionne en trouvant les distances entre une requête et tous les exemples dans les données, en sélectionnant le nombre d'exemples spécifié (K) le plus proche de la requête, puis en votant pour l'étiquette la plus fréquente dans le cas d'une classification ou en faisant la moyenne des étiquettes dans le cas de la régression.

Son implémentation peut être résumée en étapes (E_n) comme suit :

✓ E_1 : choix de l'entité à classifier parmi les différentes entités groupées

Il ne peut y avoir classification que s'il y'a au moins deux entités différentes. Dans la figure ci-dessous (figure 12), nous matérialisons la première étape de l'algorithme du KNN. Nous avons deux entités différenciées chacune par une couleur (rouge clair pour l'entité du groupe 1 et rouge noire pour celle du groupe 2). Nous avons une entité en rouge vif, qui est l'entité à classer.

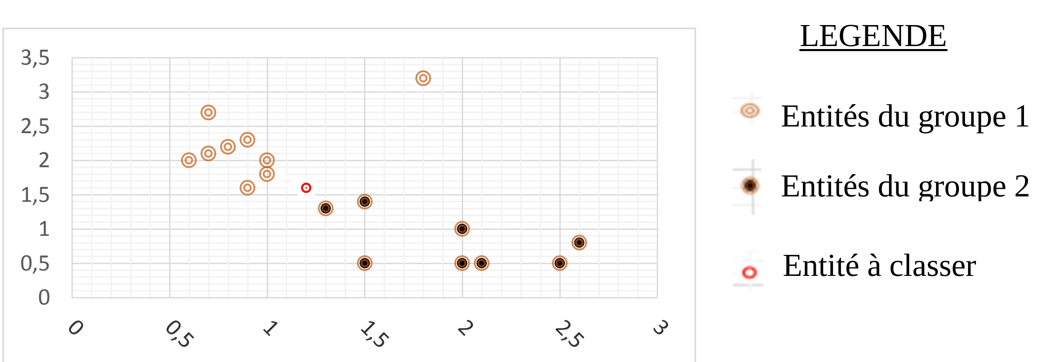


Figure 7: Matérialisation des différents groupes dans le SVM

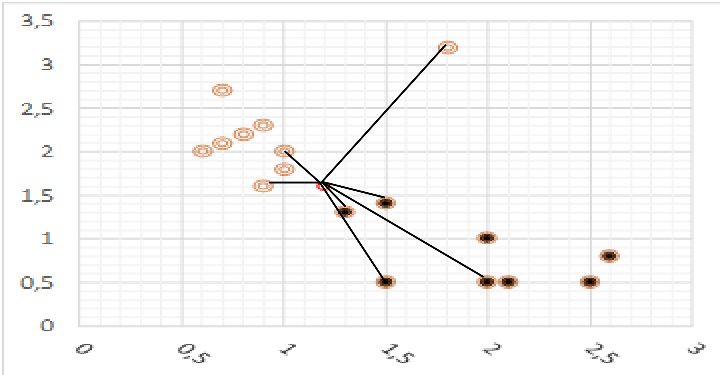


Figure 8: Représentation du calcul des distances entre divers points dans le SVM

- ✓ E_3 : Prendre les K voisins les plus proches selon la distance calculée.

Après le calcul des différentes distances, l'algorithme sélectionne les éléments/points les plus proches encore appelés voisins les plus proches de l'entité à classifier, qui vont porter le nom de « K-voisins ».

Nous avons l'illustration à la figure ci-dessous (figure 8)

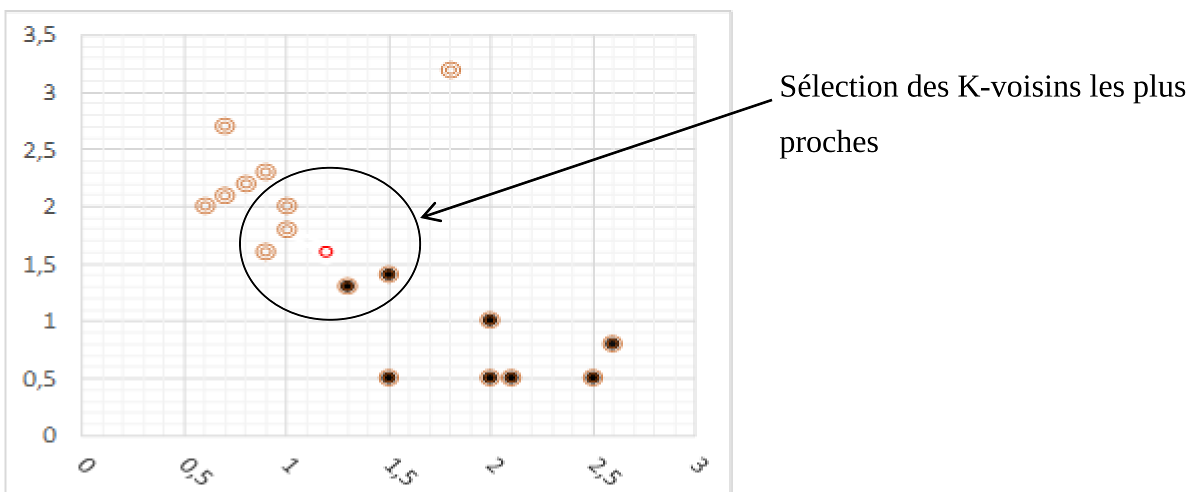


Figure 9: Représentation du choix de la sélection des k-voisins dans le SVM

- ✓ E4 : Parmi ces K voisins, compter le nombre de points appartenant à chaque catégorie.

Sur la base du précédent schéma, nous voyons que nous avons trois voisins de l'entité du groupe 1 et deux voisins de l'entité du groupe 2 : Nous avons une plus grande présence du groupe 1 que du groupe 2.

- ✓ E5 : Attribuer le nouveau point à la catégorie la plus présente parmi ces K voisins.

Sur la base de tout ce qui précède, nous pouvons conclure ou prédire que l'entité à classer fera partie du groupe 1.

❖ Avantages de l'algorithme du k-Nearest Neighbors (k-NN)

Parmi les avantages du K-NN on peut citer entre autres :

✓ Son intuitivité et sa simplicité : l'algorithme K-NN est très simple à comprendre et tout aussi facile à mettre en œuvre. Pour classer le nouveau point de données, l'algorithme K-NN lit tout l'ensemble de données pour trouver les K voisins les plus proches.

✓ Le K-NN n'a pas d'hypothèses : en effet, c'est un algorithme non paramétrique, ce qui signifie qu'il y a des hypothèses à respecter pour son implémentation. Les modèles paramétriques tels que la régression linéaire comportent de nombreuses hypothèses auxquelles les données doivent répondre avant de pouvoir être mises en œuvre, ce qui n'est pas le cas avec le K-NN.

✓ Ne nécessite aucune étape de formation : Le K-NN ne construit de manière explicite aucun modèle, mais, il balise simplement la nouvelle entrée de données basée sur l'apprentissage obtenu des données historiques. La nouvelle entrée de données est donc étiquetée avec la classe majoritaire, constituée des voisins les plus proches.

✓ Il est évolutif et son évolution est perpétuelle : étant donné qu'il s'agit d'un apprentissage basé sur des instances, le K-NN est une approche basée sur la mémoire. Ceci veut dire qu'ici, le classificateur s'adapte immédintelligence artificielle tement, au fur et à mesure que nous collectons de nouvelles données d'entraînement, permettant ainsi à l'algorithme de répondre rapidement aux changements de l'entrée pendant l'utilisation en temps réel.

✓ Très facile à implémenter pour les problèmes multi-classes : La plupart des algorithmes de classificateur sont faciles à implémenter pour les problèmes binaires et nécessitent des efforts pour être implémentés pour les multi-classes, ce qui n'est pas le cas avec le K-NN. Il s'adapte aux multi-classes sans effort supplémentaire.

❖ Inconvénients de l'algorithme du k-Nearest Neighbors (k-NN)

Bien que présentant de nombreux avantages, le K-NN à aussi de nombreux inconvénients :

- ✓ La vitesse d'exécution du K-NN fonction du jeu de données : Il est vrai que le K-NN peut être très facile à mettre en œuvre, mais plus l'ensemble de données augmente, plus l'efficacité ou la vitesse de l'algorithme diminue très rapidement.
- ✓ Forte dépendance de la dimensionnalité : Le KNN fonctionne bien avec un petit nombre de variables d'entrée, mais au fur et à mesure que le nombre de variables augmente, l'algorithme K-NN a du mal à prédire la sortie d'un nouveau point de données.
- ✓ Le K-NN a besoin de caractéristiques homogènes : si vous décidez de construire le K-NN en utilisant une distance commune, comme les distances euclidiennes ou de Manhattan, il est absolument nécessaire que les caractéristiques aient la même échelle, car les différences absolues dans les caractéristiques ont le même poids, c'est-à-dire la distance dans la caractéristique 1 doit signifier la même chose pour la caractéristique 2.
- ✓ Le Nombre optimal de voisins : l'un des plus gros problèmes avec le K-NN est de choisir le nombre optimal de voisins à prendre en compte lors de la classification de la nouvelle entrée de données.
- ✓ La sensibilité aux valeurs aberrantes : l'algorithme K-NN est très sensible aux valeurs aberrantes car il choisit simplement les voisins en fonction des critères de distance.

Conclusion:

Dans ce chapitre, nous avons tour à tour présenté l'intelligence artificielle, son implication de plus en plus croissante dans notre quotidien, les différentes parties de l'intelligence artificielle et leurs différents avantages et inconvénients. Toutefois, l'épineux problème reste en suspens : celui de l'incorporation de l'IA en industrie comme c'est le cas en médecine.

Nous avons aussi présenté les différents types algorithmes qui sont utilisés pour réaliser l'apprentissage en IA. Le récapitulatif de ces différents algorithmes est donné au tableau 3 .

Dans la suite de notre travail, il sera question de présenter la méthodologie qu'il faut utiliser pour étudier la fiabilité

Tableau 1: Modèles d'apprentissage automatique [26]

Modèle	**Algorithme lié au Modèle**	**Domaines d'application**
Apprentissage Supervisé	Régression Linéaire	• Prévision des ventes • Evaluation des risques
	KNN(K-Nearest Neighbors)	• Classification(images, pannes,etc..) • comparaison des performances financières
	Arbre de Décision	• Analyses prédictives
Apprentissage Non Supervisé	A priori	• fonction de vente • classification des mots • outil de recherche
	K-Means Clustering	• Suivi/Analyse de performance • Searcher Intent
Apprentissage par renforcement	Q learning	➢ création de politique optimale ➢ Optimisation des processus ➢ Minimiser les coûts énergétiques ➢ Maximisation des parts de revenus
	Model based value estimation	➢ tâches linéaire ➢ estimtion des paramètres

CHAPITRE II: METHODOLOGIE DE RESOLUTION D'UN PROBLEME AVEC L'INTELLIGENCE ARTIFICIELLE

Introduction

Que ce soit la reconnaissance vocale, la reconnaissance d'images, l'analyse des données, résoudre un problème par l'intelligence artificielle et précisément l'apprentissage automatique demande de suivre un ensemble d'étapes précises.

Nous allons, dans cette partie de notre travail, décrire la démarche, ou un ensemble d'étapes à suivre, pour utiliser l'intelligence artificielle et plus particulièrement l'apprentissage automatique à un cas précis. Cette méthodologie est issue du recoupage de nombreux faits sur l'apprentissage automatique. Il est certes vrai que l'apprentissage automatique a des branches différentes, mais la résolution d'un problème relatif à l'une ou l'autre branche exige de suivre un ensemble d'étapes qui seront décrites dans cette partie.

I-DÉMARCHE

Une démarche peut être définie comme étant une méthode utilisée pour guider toute production scientifique ; c'est aussi un ensemble d'étapes nécessaires pour parvenir à un résultat.
En Apprentissage automatique, la résolution d'un problème suit les étapes illustrées dans la figure 9 [23]:

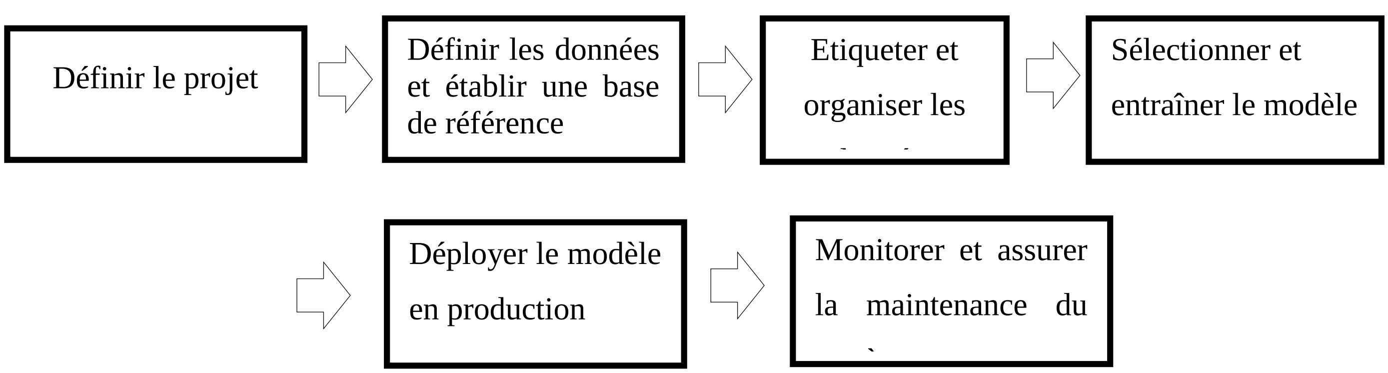


Figure 10: Cycle de vie d'un projet en Intelligence artificielle [23]

I-1. La définition du problème ou du projet

C'est la première partie et le point de départ de tout problème qui doit être résolu par l'apprentissage automatique. Il est question ici de définir clairement le problème à résoudre ; ce problème peut être la reconnaissance des formes, la reconnaissance vocale, la classification, la régression, etc…)

I-2. Définition des données et l'établissement d'une base de référence

Cette étape permet de faire un choix quant au type de données à choisir pour la résolution du problème. Nous savons qu'une donnée est une information ; il est donc nécessaire de recueillir les informations utiles, qui serviront à la meilleure compréhension et à la résolution du problème posé. Cette étape porte aussi le nom de collecte de données ou data collection.

La collecte des données, est une étape cruciale en intelligence artificielle le, dans la mesure où elle permet de capturer, d'enregistrer des événements passés, qui serviront de base dans l'analyse des données pour trouver des modèles récurrents. C'est donc à partir de ces modèles, qu'on pourra créer des modèles prédictifs à l'aide d'algorithmes d'apprentissage automatique qui rechercheront les tendances et se serviront de ces tendances pour prédire les changements futurs.

En général, en intelligence artificielle, la fiabilité et la précision des modèles prédictifs sont fonction de la qualité des données à partir desquelles ils sont construits, de sorte que de bonnes pratiques de collecte de données permettent de développer des modèles performants. Il s'agit en fait de s'assurer que les données soient exemptes d'erreurs (garbage in, garbage out) et contenir des informations pertinentes pour la tâche à accomplir.

Afin de développer des solutions pratiques d'intelligence artificielle, et d'apporter une solution concrète au problème posé, la collecte des données ne se fait pas au hasard. En effet, les données collectées doivent être des informations en rapport étroit avec le problème à résoudre.

- ✓ Dans le cas d'une application d'Intelligence artificielle, qui se chargera sur la base des images, de donner un diagnostic précis (détection d'une maladie dans le cadre médical, ou d'une panne dans le cadre industriel), la collecte des données consistera à rassembler et stocker les images (anciennes) mettant en exergue ces maladies ou ces anomalies, ou les pannes survenues le jour précédent la collecte.
- ✓ Dans le cas de la détection d'une maladie mentale, la collecte des données consistera à rassembler les anciens EEG des patients observés.

✓ Dans le cas de la prédiction de l'équipe qui pourra remporter un championnat/compétition, la collecte des données consistera à collecter l'ensemble des performances des différentes équipes durant les précédents championnats.

✓ Dans le cas de la prédiction ou du diagnostic en industrie, la collecte des données consistera à collecter l'ensemble des historiques des défaillances du système de production.

Notons qu'en général, les données collectées peuvent être soit sous forme de fichier excel (xls, csv) , de fichiers JPG, etc… et portent le nom de données brutes.

Le tableau 1 nous donne un exemple de fichier de données

Tableau 2: Données collectées ou « données brutes »

150,4,setosa,versicolor,virginica					
5.1,3.5,1.4,0.2,0					
4.9,3.0,1.4,0.2,0					
4.7,3.2,1.3,0.2,0					
4.6,3.1,1.5,0.2,0					
5.0,3.6,1.4,0.2,0					
5.4,3.9,1.7,0.4,0					
4.6,3.4,1.4,0.3,0					
5.0,3.4,1.5,0.2,0					
4.4,2.9,1.4,0.2,0					

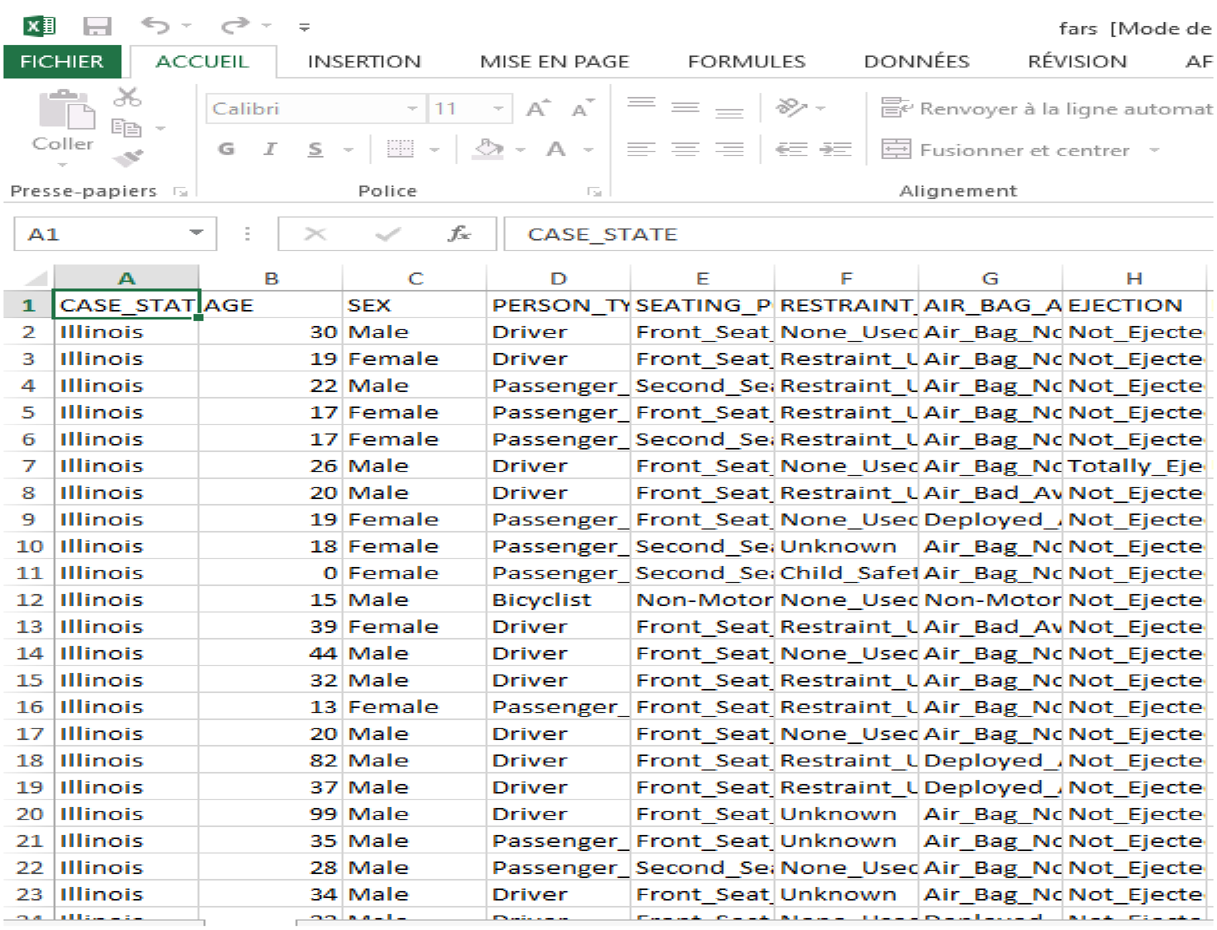
I-3. Étiquetage et organisation des données

L'étiquetage consiste à prendre les données dites « brutes » issues de la collecte et les classer en fonction de leurs catégories (lignes, colonnes). En général, lorsque les données sont par exemple dans un fichier excel, l'organisation et l'étiquetage des données consiste à s'assurer que chaque type de données se retrouve dans une colonne qui lui est propre. Si nous prenons le cas d'un modèle de prédiction d'une maladie, il est clair que dans notre fichier excel figurera dans chaque colonne : l'âge, le sexe, etc…

Cette étape présente de nombreux avantages à savoir :

➢ La Correction rapide des erreurs : La préparation des données permet de détecter les erreurs avant le traitement. Une fois les données supprimées de leur source d'origine, ces erreurs deviennent plus difficiles à comprendre et à corriger.

- La production des données de qualité supérieure : Le nettoyage et le reformatage des jeux de données garantit que toutes les données utilisées dans l'analyse seront de haute/bonne qualité.
- Une meilleure prise de décisions : Le fait d'avoir des données de meilleure qualité entraine un traitement et une analyse plus rapide et plus efficace, conduisant à de prises de décisions plus rapides, efficaces et de haute qualité.
- Habituellement, dans un environnement professionnel, votre patron vous enverra simplement un ensemble de données, et c'est à vous de les transformer, et les rendre exploitables.
- Pour ce faire, nous devrons explorer les données. Tout d'abord, il faut inspecter les données et leurs propriétés. Différents types de données comme les données numériques, les données catégorielles, les données ordinales et nominales, etc. nécessitent des traitements différents.
- La figure 10 présente un exemple de données étiquetées sous Excel



	A	B	C	D	E	F	G	H
1	CASE_STAT	AGE	SEX	PERSON_TY	SEATING_P	RESTRAINT	AIR_BAG_A	EJECTION
2	Illinois	30	Male	Driver	Front_Seat	None_Usec	Air_Bag_Nc	Not_Ejecte
3	Illinois	19	Female	Driver	Front_Seat	Restraint_U	Air_Bag_Nc	Not_Ejecte
4	Illinois	22	Male	Passenger_	Second_Se	Restraint_U	Air_Bag_Nc	Not_Ejecte
5	Illinois	17	Female	Passenger_	Front_Seat	Restraint_U	Air_Bag_Nc	Not_Ejecte
6	Illinois	17	Female	Passenger_	Second_Se	Restraint_U	Air_Bag_Nc	Not_Ejecte
7	Illinois	26	Male	Driver	Front_Seat	None_Usec	Air_Bag_Nc	Totally_Eje
8	Illinois	20	Male	Driver	Front_Seat	Restraint_U	Air_Bad_Av	Not_Ejecte
9	Illinois	19	Female	Passenger_	Front_Seat	None_Usec	Deployed_	Not_Ejecte
10	Illinois	18	Female	Passenger_	Second_Se	Unknown	Air_Bag_Nc	Not_Ejecte
11	Illinois	0	Female	Passenger_	Second_Se	Child_Safet	Air_Bag_Nc	Not_Ejecte
12	Illinois	15	Male	Bicyclist	Non-Motor	None_Usec	Non-Motor	Not_Ejecte
13	Illinois	39	Female	Driver	Front_Seat	Restraint_U	Air_Bad_Av	Not_Ejecte
14	Illinois	44	Male	Driver	Front_Seat	None_Usec	Air_Bag_Nc	Not_Ejecte
15	Illinois	32	Male	Driver	Front_Seat	Restraint_U	Air_Bag_Nc	Not_Ejecte
16	Illinois	13	Female	Passenger_	Front_Seat	Restraint_U	Air_Bag_Nc	Not_Ejecte
17	Illinois	20	Male	Driver	Front_Seat	None_Usec	Air_Bag_Nc	Not_Ejecte
18	Illinois	82	Male	Driver	Front_Seat	Restraint_U	Deployed_	Not_Ejecte
19	Illinois	37	Male	Driver	Front_Seat	Restraint_U	Deployed_	Not_Ejecte
20	Illinois	99	Male	Driver	Front_Seat	Unknown	Air_Bag_Nc	Not_Ejecte
21	Illinois	35	Male	Passenger_	Front_Seat	Unknown	Air_Bag_Nc	Not_Ejecte
22	Illinois	28	Male	Passenger_	Second_Se	None_Usec	Air_Bag_Nc	Not_Ejecte
23	Illinois	34	Male	Driver	Front_Seat	Unknown	Air_Bag_Nc	Not_Ejecte

Figure 11: Données étiquetées et organisées dans un fichier xls ou csv

I-4. Sélectionner et entraîner le modèle

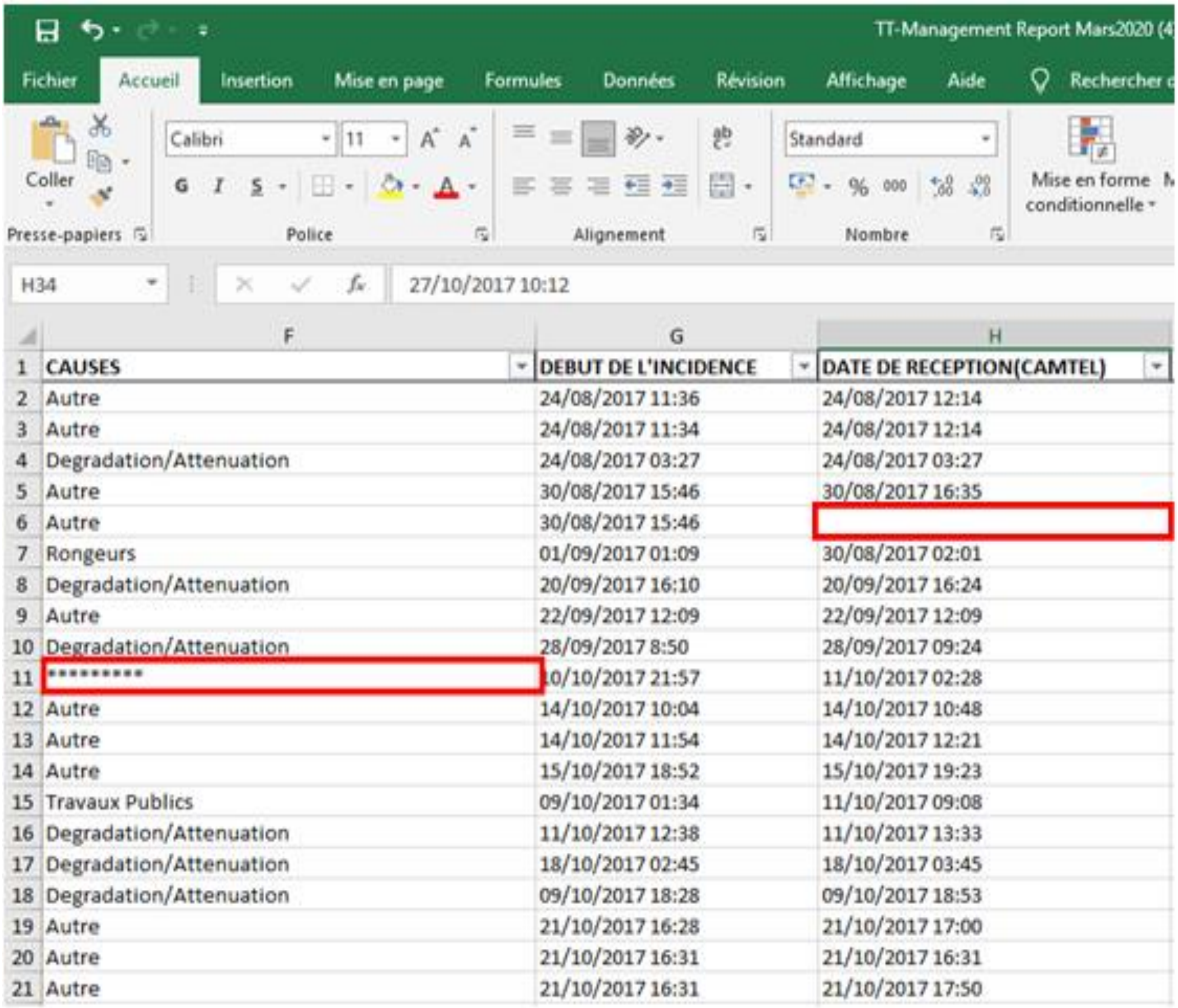
Cette étape est subdivisée en trois sous étapes :

- ✓ Le Data cleaning
- ✓ Le Feature engineering
- ✓ Le Model Selection ou Sélection du modèle

I-5. Le Data Cleaning ou Nettoyage des données

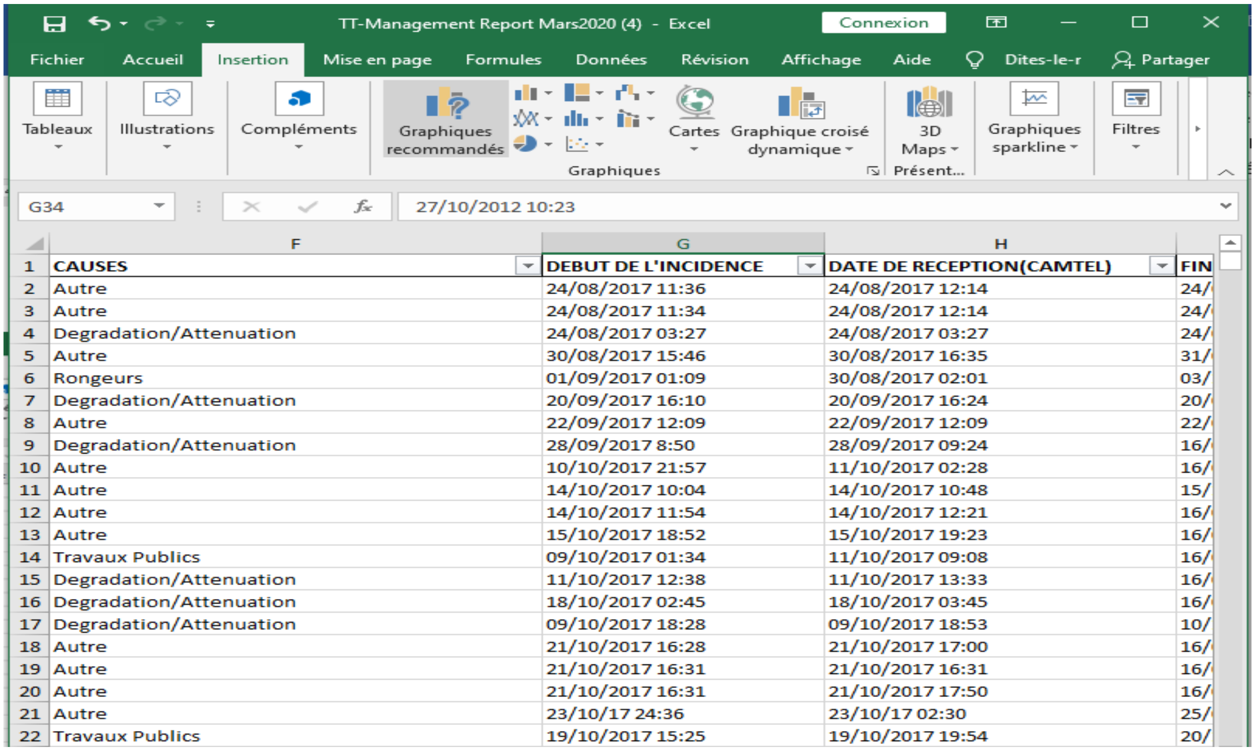
Le nettoyage des données joue un rôle important dans le processus analytique, l'objectif étant de créer des ensembles de données standardisés et uniformes pour faciliter l'analyse de données. Ainsi, cette étape, consiste à effectuer les opérations suivantes :

- ➢ Supprimer les doublons ou valeurs non pertinentes, en se débarrassant des données qui apparaissent plus d'une fois ou alors les données qui n'ont pas une implication/corrélation avec le problème à résoudre.
- ➢ Supprimer les données mal étiquetées, ou de même catégorie se produisant plusieurs fois : il peut arriver que certaines données portent des noms ou valeurs erronées, ou des catégories identiques qui se reproduisent plusieurs fois
- ➢ Supprimer les points de données manquants ou nuls.
- ➢ Supprimer les valeurs aberrantes ou inattendues (figure 11).



	CAUSES	DEBUT DE L'INCIDENCE	DATE DE RECEPTION(CAMTEL)
2	Autre	24/08/2017 11:36	24/08/2017 12:14
3	Autre	24/08/2017 11:34	24/08/2017 12:14
4	Degradation/Attenuation	24/08/2017 03:27	24/08/2017 03:27
5	Autre	30/08/2017 15:46	30/08/2017 16:35
6	Autre	30/08/2017 15:46	
7	Rongeurs	01/09/2017 01:09	30/08/2017 02:01
8	Degradation/Attenuation	20/09/2017 16:10	20/09/2017 16:24
9	Autre	22/09/2017 12:09	22/09/2017 12:09
10	Degradation/Attenuation	28/09/2017 8:50	28/09/2017 09:24
11	*********	10/10/2017 21:57	11/10/2017 02:28
12	Autre	14/10/2017 10:04	14/10/2017 10:48
13	Autre	14/10/2017 11:54	14/10/2017 12:21
14	Autre	15/10/2017 18:52	15/10/2017 19:23
15	Travaux Publics	09/10/2017 01:34	11/10/2017 09:08
16	Degradation/Attenuation	11/10/2017 12:38	11/10/2017 13:33
17	Degradation/Attenuation	18/10/2017 02:45	18/10/2017 03:45
18	Degradation/Attenuation	09/10/2017 18:28	09/10/2017 18:53
19	Autre	21/10/2017 16:28	21/10/2017 17:00
20	Autre	21/10/2017 16:31	21/10/2017 16:31
21	Autre	21/10/2017 16:31	21/10/2017 17:50

Figure 12: Données contenant les valeurs aberrantes ou manquantes



	CAUSES	DEBUT DE L'INCIDENCE	DATE DE RECEPTION(CAMTEL)	FIN
2	Autre	24/08/2017 11:36	24/08/2017 12:14	24/
3	Autre	24/08/2017 11:34	24/08/2017 12:14	24/
4	Degradation/Attenuation	24/08/2017 03:27	24/08/2017 03:27	24/
5	Autre	30/08/2017 15:46	30/08/2017 16:35	31/
6	Rongeurs	01/09/2017 01:09	30/08/2017 02:01	03/
7	Degradation/Attenuation	20/09/2017 16:10	20/09/2017 16:24	20/
8	Autre	22/09/2017 12:09	22/09/2017 12:09	22/
9	Degradation/Attenuation	28/09/2017 8:50	28/09/2017 09:24	16/
10	Autre	10/10/2017 21:57	11/10/2017 02:28	16/
11	Autre	14/10/2017 10:04	14/10/2017 10:48	15/
12	Autre	14/10/2017 11:54	14/10/2017 12:21	16/
13	Autre	15/10/2017 18:52	15/10/2017 19:23	16/
14	Travaux Publics	09/10/2017 01:34	11/10/2017 09:08	16/
15	Degradation/Attenuation	11/10/2017 12:38	11/10/2017 13:33	16/
16	Degradation/Attenuation	18/10/2017 02:45	18/10/2017 03:45	16/
17	Degradation/Attenuation	09/10/2017 18:28	09/10/2017 18:53	10/
18	Autre	21/10/2017 16:28	21/10/2017 17:00	16/
19	Autre	21/10/2017 16:31	21/10/2017 16:31	16/
20	Autre	21/10/2017 16:31	21/10/2017 17:50	16/
21	Autre	23/10/17 24:36	23/10/17 02:30	25/
22	Travaux Publics	19/10/2017 15:25	19/10/2017 19:54	20/

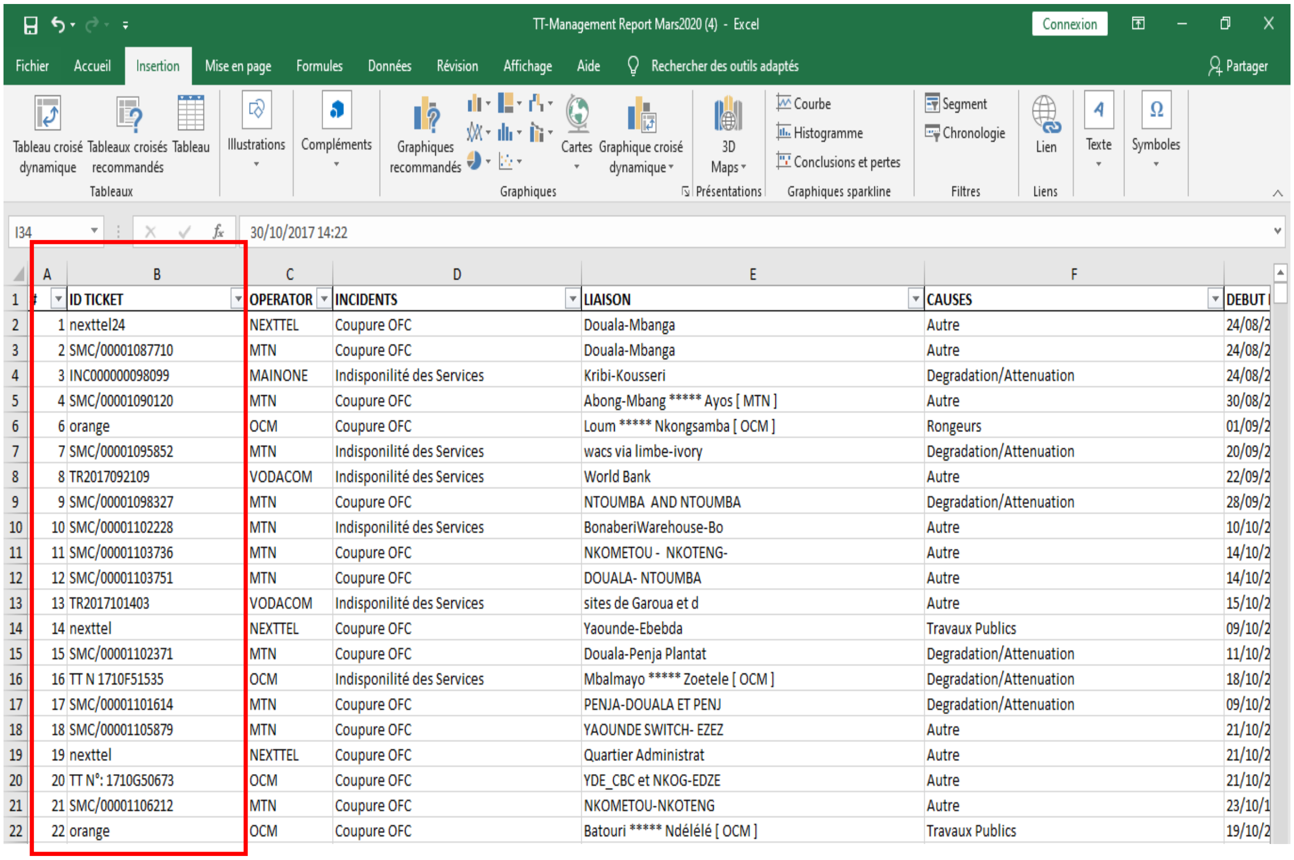
Figure 13: Données nettoyées(après suppression des données manquantes et/ou aberrantes)

I-5-1. Le Feature Engineering

Dans cette étape on procède à la transformation des données collectées en fonctionnalités qui représentent mieux le problème que nous essayons de résoudre pour le modèle, afin d'améliorer ses performances et sa précision. Par exemple, si nous prenons un algorithme qui veut prédire l'apparition d'une maladie sur un individu, la taille à priori n'est pas un indicateur de la maladie ! par conséquent cette entrée dans notre ensemble de données peut être ignorée.

Il est donc question dans cette étape, à partir de la connaissance de nos données, de construire des variables explicatives, des fonctionnalités, qui peuvent être utilisées pour former un modèle prédictif. La puissance du modèle de prédiction sera fonction de la bonne mise en œuvre de cette étape (Choix judicieux des variables explicatives et des fonctionnalités).

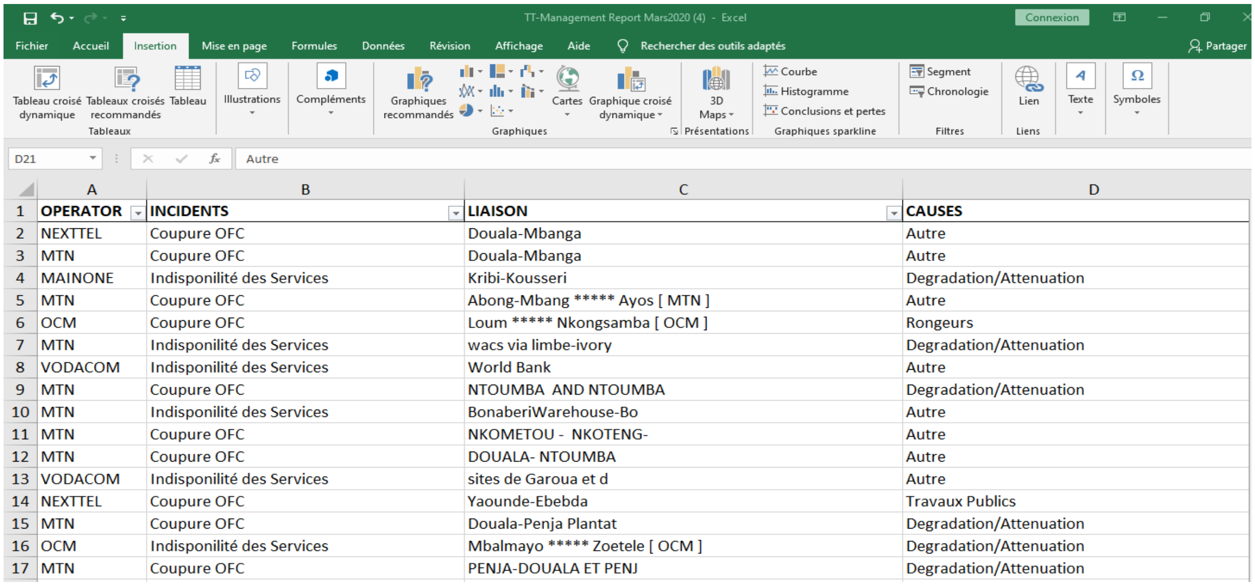
Pour résoudre le problème dont les données sont dans à la figure 13, nous devons nous appuyer sur l'objectif recherché par la résolution de notre algorithme. Notre algorithme doit permettre de prédire une défaillance. Pour ce faire, nous devons, parmi les entrées de notre jeu de données supprimer celles qui ne sont pas pertinentes dans la résolution de notre problème. Par exemple, l'information ID TICKET dans notre jeu de données n'est pas pertinente.



#	ID TICKET	OPERATOR	INCIDENTS	LIAISON	CAUSES	DEBUT
1	nexttel24	NEXTTEL	Coupure OFC	Douala-Mbanga	Autre	24/08/2
2	SMC/00001087710	MTN	Coupure OFC	Douala-Mbanga	Autre	24/08/2
3	INC000000098099	MAINONE	Indisponilité des Services	Kribi-Kousseri	Degradation/Attenuation	24/08/2
4	SMC/00001090120	MTN	Coupure OFC	Abong-Mbang ***** Ayos [MTN]	Autre	30/08/2
5	orange	OCM	Coupure OFC	Loum ***** Nkongsamba [OCM]	Rongeurs	01/09/2
7	SMC/00001095852	MTN	Indisponilité des Services	wacs via limbe-ivory	Degradation/Attenuation	20/09/2
8	TR2017092109	VODACOM	Indisponilité des Services	World Bank	Autre	22/09/2
9	SMC/00001098327	MTN	Coupure OFC	NTOUMBA AND NTOUMBA	Degradation/Attenuation	28/09/2
10	SMC/00001102228	MTN	Indisponilité des Services	BonaberiWarehouse-Bo	Autre	10/10/2
11	SMC/00001103736	MTN	Coupure OFC	NKOMETOU - NKOTENG-	Autre	14/10/2
12	SMC/00001103751	MTN	Coupure OFC	DOUALA- NTOUMBA	Autre	14/10/2
13	TR2017101403	VODACOM	Indisponilité des Services	sites de Garoua et d	Autre	15/10/2
14	nexttel	NEXTTEL	Coupure OFC	Yaounde-Ebebda	Travaux Publics	09/10/2
15	SMC/00001102371	MTN	Coupure OFC	Douala-Penja Plantat	Degradation/Attenuation	11/10/2
16	TT N 1710F51535	OCM	Indisponilité des Services	Mbalmayo ***** Zoetele [OCM]	Degradation/Attenuation	18/10/2
17	SMC/00001101614	MTN	Coupure OFC	PENJA-DOUALA ET PENJ	Degradation/Attenuation	09/10/2
18	SMC/00001105879	MTN	Coupure OFC	YAOUNDE SWITCH- EZEZ	Autre	21/10/2
19	nexttel	NEXTTEL	Coupure OFC	Quartier Administrat	Autre	21/10/2
20	TT N°: 1710G50673	OCM	Coupure OFC	YDE_CBC et NKOG-EDZE	Autre	21/10/2
21	SMC/00001106212	MTN	Coupure OFC	NKOMETOU-NKOTENG	Autre	23/10/1
22	orange	OCM	Coupure OFC	Batouri ***** Ndélélé [OCM]	Travaux Publics	19/10/2

Figure 14: Données contenant les paramètres non pertinents pour le modèle de prédiction

Après le feature ingeineering, on a la figure 14 ci-dessous .



	A	B	C	D
1	OPERATOR	INCIDENTS	LIAISON	CAUSES
2	NEXTTEL	Coupure OFC	Douala-Mbanga	Autre
3	MTN	Coupure OFC	Douala-Mbanga	Autre
4	MAINONE	Indisponilité des Services	Kribi-Kousseri	Degradation/Attenuation
5	MTN	Coupure OFC	Abong-Mbang ***** Ayos [MTN]	Autre
6	OCM	Coupure OFC	Loum ***** Nkongsamba [OCM]	Rongeurs
7	MTN	Indisponilité des Services	wacs via limbe-ivory	Degradation/Attenuation
8	VODACOM	Indisponilité des Services	World Bank	Autre
9	MTN	Coupure OFC	NTOUMBA AND NTOUMBA	Degradation/Attenuation
10	MTN	Indisponilité des Services	BonaberiWarehouse-Bo	Autre
11	MTN	Coupure OFC	NKOMETOU - NKOTENG-	Autre
12	MTN	Coupure OFC	DOUALA- NTOUMBA	Autre
13	VODACOM	Indisponilité des Services	sites de Garoua et d	Autre
14	NEXTTEL	Coupure OFC	Yaounde-Ebebda	Travaux Publics
15	MTN	Coupure OFC	Douala-Penja Plantat	Degradation/Attenuation
16	OCM	Indisponilité des Services	Mbalmayo ***** Zoetele [OCM]	Degradation/Attenuation
17	MTN	Coupure OFC	PENJA-DOUALA ET PENJ	Degradation/Attenuation

Figure 15: Données après suppression des paramètres non pertinents

Tout ceci nous fait comprendre que la suppression d'une ligne ne se fait pas au hasard. Car plus on supprime de ligne plus on peut courir le risque de compliquer la prédiction. En général, pour mieux statuer sur la suppression ou non des lignes contenant les valeurs aberrantes, on peut chercher à connaitre le pourcentage de valeurs aberrantes par ligne et en fonction de ce pourcentage, supprimer ou non. En effet, si le pourcentage de valeurs aberrantes est très faible, il est inutile de supprimer les lignes, puisqu'elles n'auront aucun impact sur la prédiction

I-5-2. Le Model Selection ou Sélection du Modèle

Ici, nous sélectionnons le modèle ou l'algorithme qui fonctionne mieux pour l'ensemble des données en notre possession. Dans cette phase, nous avons plusieurs approches d'apprentissage automatique qui s'offrent à nous. Mais le choix final se focalise sur celui qui a la meilleure précision.

Il existe de nombreux algorithmes d'apprentissage. Nous pouvons citer entre autre :

- ✓ L'arbre de décision (DecisionTree)
- ✓ Les Séparateurs à Vastes Marges(SVM ou Support Vector Machine)
- ✓ Les *K* plus proches voisins(K-Nearest Neighbors)

❖ Notion de Valeur Catégorielle

Dans de nombreuses activités d'apprentissage automatique ou de science des données, il peut arriver que les données qui sont en notre possession contiennent du texte ou des valeurs non numériques : on les appelle valeurs catégorielles.

Malheureusement, les algorithmes de l'intelligence artificielle fonctionnent avec des entrées numériques. Par conséquent, le principal défi est de convertir des données textuelles/ catégorielles en données numériques, tout en gardant le fond et le sens du problème à résoudre.

On distingue donc différentes méthodes de transformation des données catégorielles :

– **One Hot Encoding (OHE)**

Il existe de nombreuses façons de convertir des valeurs catégorielles en valeurs numériques. Chaque approche ayant ses propres compromis et son impact sur l'ensemble de fonctionnalités.

Le One Hot Encoding est l'une de ces méthodes:

Cette méthode consiste à convertir chaque valeur de catégorie en une nouvelle, mais cette fois numérique, en attribuant la valeur 1 ou 0 en fonction de l'état de la donnée. Cela présente l'avantage de ne pas pondérer une valeur de manière incorrecte.

Son implémentation dans notre jeu de données en choisissant comme entrées celles qui ont des textes à savoir : OPERATOR, LIAISON, INCIDENTS, CAUSES et SUIVI ACTION 01 :

INCIDENCE	OPERATOR_CAMTEL	OPERATOR_GILAT	OPERATOR_MAINONE	OPERATOR_MATRIX	OPERATOR_MTN	OPERATOR_NEXTTEL
2017-08-24	0	0	0	0	0	1
2017-08-24	0	0	0	0	1	0
2017-08-24	0	0	1	0	0	0
2017-08-30	0	0	0	0	1	0
2017-08-30	0	0	0	0	0	0

Figure 16:Implémentation du OHE à l'entrée OPERATOR

INCIDENCE	CAUSES_AH	CAUSES_ATR	CAUSES_BCI	CAUSES_CEE	CAUSES_CFO	CAUSES_CN	CAUSES_DGT	CAUSES_FB	CAUSES_PE
2017-08-24	0	1	0	0	0	0	0	0	0
2017-08-24	0	1	0	0	0	0	$\boxed{0}$	0	0
2017-08-24	0	0	0	0	0	0	1	0	0
2017-08-30	0	1	0	0	0	0	0	0	0
2017-08-30	0	1	0	0	0	0	0	0	0

Figure 17: Implémentation du OHE à l'entrée CAUSES

Les figures 15&16 nous montrent les résultats de l'implémentation du OHE. Nous constatons qu'effectivement les tableaux sont remplis de 1&0. Si nous prenons la figure 16, nous constatons que le 24-08-2017 CAUSES_DGT est à 1 et 0 les autres jours. Ce qui veut dire que cette cause est apparue le 24-08-2017 et n'est plus apparue les autres jours. Le raisonnement est le même dans la figure 15. Sur la base de ce qui précède nous voyons que le OHE permet effectivement de passer des données catégorielles aux données numériques.

– **Le Frequency Encoding**

C'est une façon d'utiliser la fréquence des catégories comme étiquettes. Dans les cas où la fréquence est quelque peu liée à la variable cible, elle aide le modèle à comprendre et à attribuer le poids en proportion directe et inverse, selon la nature des données. Mais avant de l'implémenter, nous devons d'abord sélectionner les entrées sur lesquelles l'appliquer.

Nous avons son illustration à la figure ci-dessous :

	#	OPERATOR	INCIDENTS	LIAISON	CAUSES	STATU	CAUSES_encode	INCIDENTS_encode	LIAISON_encode	OPERATOR_encode
0	1	NEXTTEL	I4	Douala-Mbanga	ATR	OK	533	18	2	625
1	2	MTN	I4	Douala-Mbanga	ATR	OK	533	18	2	1104
2	3	MAINONE	I6	Kribi-Kousseri	DGT	OK	190	1763	20	30
3	4	MTN	I4	Abong-Mbang ***** Ayos [MTN]	ATR	OK	533	18	4	1104
4	5	OCM	I4	Ayos ***** Abong-MbangII [OCM]	ATR	OK	533	18	13	1101

Figure 18: Visualisation des données après application du Frequency Encoding

– Le Label Encoding

Encore appelé Encodage d'étiquettes, est une approche qui consiste à convertir chaque valeur d'une colonne en un nombre choisi au hasard. Donc pour chaque valeur texte on a une valeur numérique Elle peut se faire de deux manières:

- Soit manuellement: dans ce cas, on attribut des valeurs numériques aux valeurs catégorielles qu'on veut étudier
- Soit automatiquement : dans ce cas, on laisse le soin à la machine d'attribuer les valeurs numériques aux valeurs catégorielles

L'utilisation de la première méthode a l'avantage de décider nous-mêmes des valeurs que nous voulons donner, mais dans le cas où il y'a un jeu de données considérables, cette opération devient fastidieuse.

La deuxième méthode est utilisée pour des jeux de données volumineux. Son avantage est que tout est fait par la machine. Son illustration est faite à la figure 18 (ci-dessous)

	OPERATOR	INCIDENTS	LIAISON	CAUSES	OPERATOR_L	INCIDENTS_L	LIAISON_L	CAUSES_L
0	NEXTTEL	I4	Douala-Mbanga	ATR	7	3	211	1
1	MTN	I4	Douala-Mbanga	ATR	6	3	211	1
2	MAINONE	I6	Kribi-Kousseri	DGT	4	5	320	6
3	MTN	I4	Abong-Mbang ***** Ayos [MTN]	ATR	6	3	9	1
4	OCM	I4	Ayos ***** Abong-MbangII [OCM]	ATR	8	3	20	1

Figure 19: label Encoding: en vert les données non encodées et en rouge les données encodé

INCIDENT	ABREVIATION
Coupure OFC	I1
Coupure d'une BTS MSAN Noeuds	I2
Coupure Energie	I3
Indisponibilite du Signal	I5
Indisponilite des Services	I6
Travaux programmes	I7

(a)

CAUSE	ABREVIATION
Pannes Electriques	PE
Probleme Distant lie à l'Operateur	PLA
Rongeurs	RG
Travaux programmes	TPP
Travaux Publics	TP
Vandalisme	VD
ACTION HUMAINE	AH
Autre	ATR
Baisse de la capacité Internet	BCI
Catastrophe Naturelle	CN
Coupure de la Fibre Optique	CFO
Coupure Energie Electrique	CEE
Degradation/Attenuation	DGT
Feu de brousse	FB

(b)

Figure 20: Légendes utilisées : (a) les INCIDENTS ; (b) les CAUSES

❖ Choix du Type d'encodage

Nous avons comparé les différentes méthodes d'encodage des donnés à savoir :

1- Le OHE, qui avait l'avantage de convertir nos données catégorielles en données numériques, mais qui présentait l'inconvénient d'alourdir le jeu de données

2- Le Frequency encoding, qui présentait l'avantage de représenter numériquement les valeurs catégorielles en fonction de leur occurrence dans le jeu de données. Mais son principal inconvénient est qu'il peut arriver que deux données différentes aient la même fréquence d'occurrence, ce qui va fausser la prédiction.

3- Le Label encoding, qui lui peut se faire automatiquement ou manuellement ; son avantage est qu'il transforme les données catégorielles en données numériques tout en conservant le volume du jeu de données.

I-6. Déploiement et monitoring du modèle en production

Ici, c'est la dernière partie, de notre projet. Elle consiste en fait à déployer l'algorithme et à le monitorer. En fait l'objectif ici est d'élaborer l'algorithme de fonctionnement de l'apprentissage, de choisir le langage de programmation à utiliser et à écrire le code permettant d'exécuter cette application dans une machine pour qu'il fonctionne dans un environnement industriel

Pour les besoins d'apprentissage, il sera question, sur la base de l'historique des pannes collectées, de diviser notre jeu de données en trois lots : une partie de données pour l'entrainement, une autre partie de données pour le test, et une dernière quantité de données qui serviront à d'avantage affiner notre algorithme. Pourquoi procéder de la sorte ?

- Par ce que le jeu de données qui est en notre possession est identique au jeu de données que l'algorithme étudiera dans le futur pour faire des prédictions.
- Par soucis de performance et en vue de s'assurer de l'effectivité de la prédiction.

En effet, Il serait bon d'avoir au moins 60% des données pour l'entrainement, 10% de données pour le test et le reste pour s'assurer du bon fonctionnement de notre algorithme. Ensuite, il faut définir un état aléatoire, qui garantit que les divisions générées sont reproductibles. Et pour finir, il faut passer à l'implémentation en utilisant l'algorithme choisi parmi les algorithmes d'apprentissage.

Conclusion

Dans cette partie il a été question pour nous, de décrire de manière succincte la démarche permettant d'utiliser l'apprentissage automatique pour faire des prédictions. Les étapes décrites ici sont indépendantes du fait que l'on veule faire de la reconnaissance vocale, la reconnaissance des formes ou la prédiction d'un évènement quelconque.

Dans la partie suivante, il sera question d'appliquer les méthodes décrites ici à un problème concret, celui de la prédiction des pannes dans un réseau de télécommunications

CHAPITRE III : CAS PRATIQUE : PREDICTION D'UNE PANNE DANS LE RESEAU DE TELECOMMUNICATION A FIBRE OPTIQUE

Introduction

Après avoir décrit et présenté la méthodologie à employer pour résoudre un problème en intelligence artificielle et plus précisément en apprentissage automatique, nous allons dans cette partie nous atteler à utiliser cette méthodologie pour faire la prédiction des pannes qui pourraient surgir dans un réseau de télécommunications, particulièrement le réseau en fibre optique

I-DÉMARCHE

Rappelons qu'au chapitre précédent, nous avons schématisé la démarche à suivre pour la résolution d'un problème en Intelligence artificielle, démarche qui est matérialisée à la figure ci-dessous :

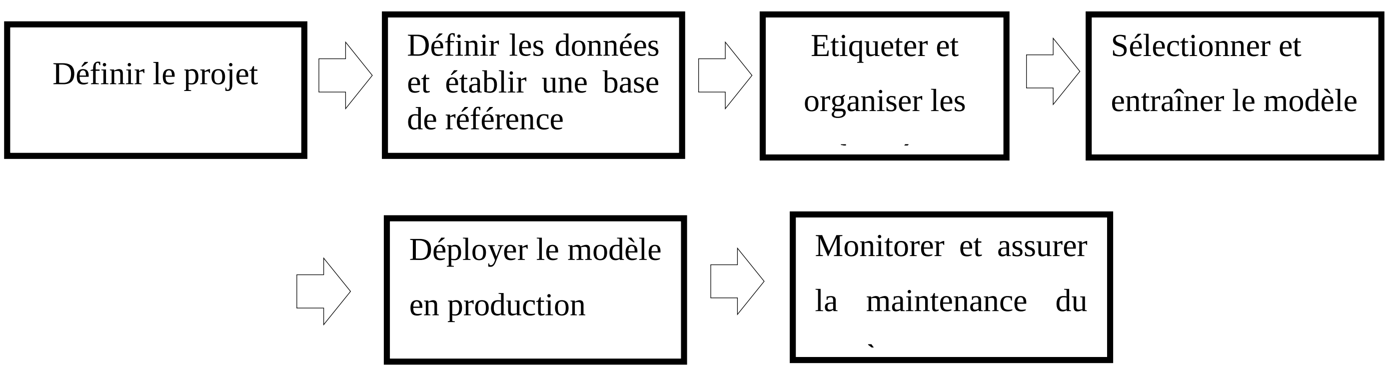


Figure 21: Cycle de vie d'une Projet en Intelligence Artificielle [24]

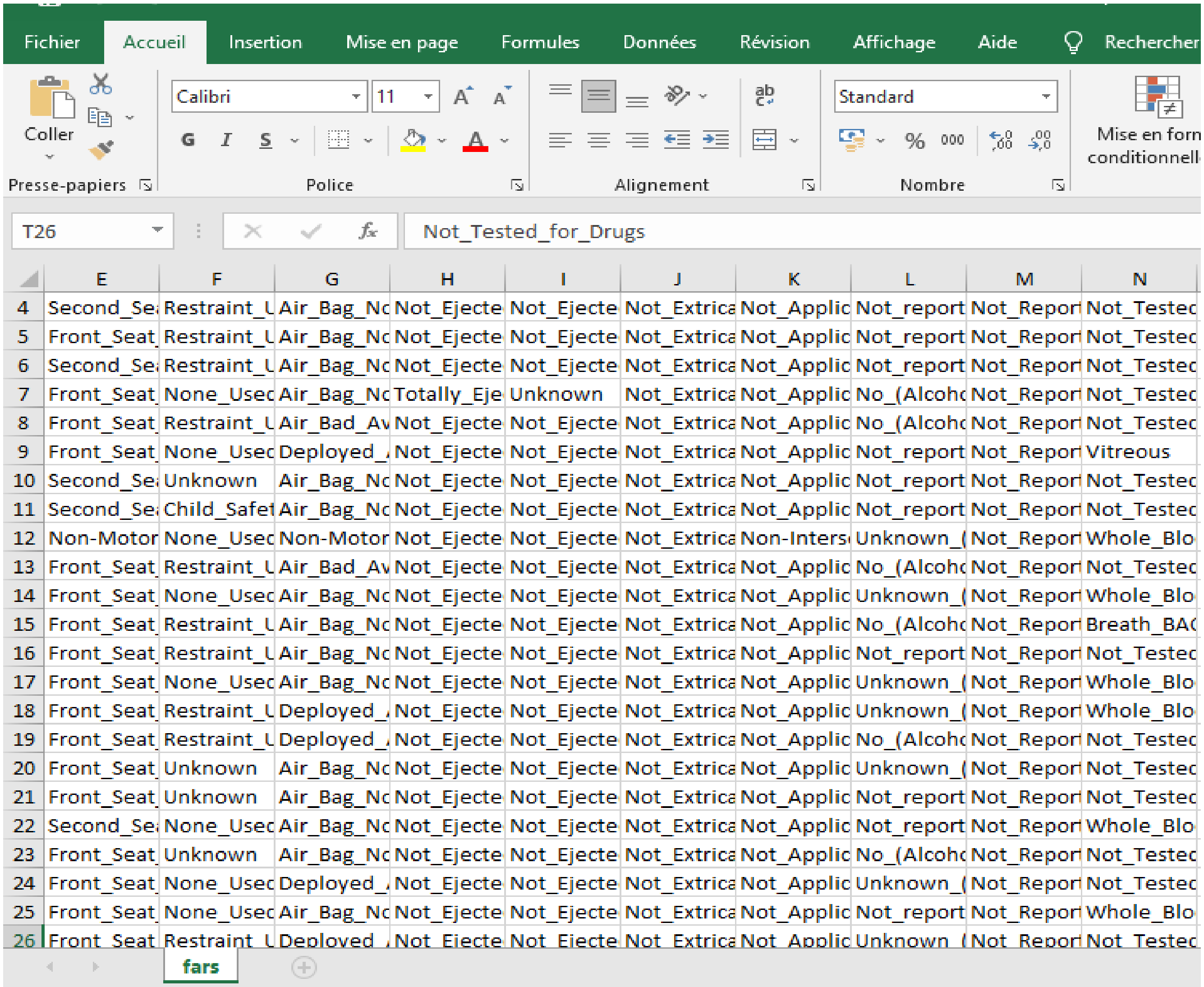
I-1. La définition du problème ou du projet

C'est le point de départ de tout problème qui doit être résolu par l'apprentissage automatique.

Le problème qui nous est donné de résoudre est celui de la prédiction d'une défaillance pouvant surgir sur un réseau de télécommunication en fibre optique. En effet, le réseau de télécommunication Camerounais étant sujet à de multiples pannes et à répétition, il sera question, d'utiliser l'apprentissage automatique et l'historique des pannes, afin de prédire l'occurrence d'une panne dans ce réseau. Pour ce faire, nous allons suivre les étapes décrites au chapitre précédent. Les données collectées sont celles de l'historique des pannes enregistrées sur le serveur de CAMTEL pendant quatre ans (de 2017 à 2020). Ces données contiennent entre autre, l'incident répertorié, le jour d'occurrence de l'incident, l'heure, la cause de l'incident, l'opérateur impacté par l'incident, la liaison impactée par l'incident, et biens d'autres informations qui seront utiles pour les opérateurs afin d'assurer la maintenance de leur réseau.

I-2. La Définition des données et l'établissement d'une base de référence

Comme expliqué précédemment, cette étape consiste à choisir les données devant être collectées pour la résolution de notre problème. Les données que nous utiliserons sont des données collectées par l'outil de monitoring (M2000/U2000) de l'opérateur de télécommunications du Cameroun CAMTEL. Ces données représentent l'ensemble des pannes qui sont enregistrées sur le réseau pendant les quatre dernières années précédant la collecte. Toutefois, dans un souci de clarté il serait préférable de conserver ces données dans un fichier excel ou xls, car ce sont ces extensions qui sont prises en charge par le logiciel ANACONDA. Un échantillon de ces données est à la figure 22



	E	F	G	H	I	J	K	L	M	N
4	Second_Se	Restraint_U	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
5	Front_Seat_	Restraint_U	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
6	Second_Se	Restraint_U	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
7	Front_Seat_	None_Usec	Air_Bag_Nc	Totally_Eje	Unknown	Not_Extrica	Not_Applic	No_(Alcohc	Not_Repor	Not_Testec
8	Front_Seat_	Restraint_U	Air_Bad_Av	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	No_(Alcohc	Not_Repor	Not_Testec
9	Front_Seat_	None_Usec	Deployed_	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Vitreous
10	Second_Se	Unknown	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
11	Second_Se	Child_Safet	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
12	Non-Motor	None_Usec	Non-Motor	Not_Ejecte	Not_Ejecte	Not_Extrica	Non-Inters	Unknown_(	Not_Repor	Whole_Blo
13	Front_Seat_	Restraint_U	Air_Bad_Av	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	No_(Alcohc	Not_Repor	Not_Testec
14	Front_Seat_	None_Usec	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Unknown_(	Not_Repor	Whole_Blo
15	Front_Seat_	Restraint_U	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	No_(Alcohc	Not_Repor	Breath_BA(
16	Front_Seat_	Restraint_U	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
17	Front_Seat_	None_Usec	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Unknown_(	Not_Repor	Whole_Blo
18	Front_Seat_	Restraint_U	Deployed_	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Unknown_(	Not_Repor	Whole_Blo
19	Front_Seat_	Restraint_U	Deployed_	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	No_(Alcohc	Not_Repor	Not_Testec
20	Front_Seat_	Unknown	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Unknown_(	Not_Repor	Not_Testec
21	Front_Seat_	Unknown	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Not_Testec
22	Second_Se	None_Usec	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Whole_Blo
23	Front_Seat_	Unknown	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	No_(Alcohc	Not_Repor	Not_Testec
24	Front_Seat_	None_Usec	Deployed_	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Unknown_(	Not_Repor	Not_Testec
25	Front_Seat_	None_Usec	Air_Bag_Nc	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Not_report	Not_Repor	Whole_Blo
26	Front_Seat_	Restraint_U	Deployed_	Not_Ejecte	Not_Ejecte	Not_Extrica	Not_Applic	Unknown_(	Not_Repor	Not_Testec

Figure 22: Données collectées ou brutes

I-3. Étiquetage et organisation des données

Les données collectées telles que vues précédemment sont des données « brutes », donc non directement utilisables et à priori difficilement déchiffrables. Grâce à cette opération, qui peut être manuelle ou automatique, on parvient à rendre les données collectées ou données « brutes », plus lisibles, et facilement exploitables, ce qui nous donne une vue globale de notre jeu de données. L'objectif étant de classer le flot de données collectées de telle sorte qu'elles puissent être regroupées en lignes et colonnes.

Vu que la préparation manuellement est fastidieuse, nous utilisons la préparation automatique avec le Logiciel Anaconda, particulièrement l'environnement de développement Jupyter Notebook :

La syntaxe permettant de ranger ces données et de les afficher dans Jupyter Notebook est à la figure 23.

```
Import pandas as pd
df=pd.read_excel("dtas.xls")
df.head(5)
```

Figure 23: Syntaxe permettant de ranger et afficher les donner dans Jupyter Notebook

Cette syntaxe nous permet d'importer tous les packages nécessaires à la prise en charge et l'importation des données collectées situées sur notre fichier (excel, csv, ect..).

De manière générale, le jeu de données dans python porte le nom df qui veut dire dataframe, et d'afficher les cinq premières lignes du fichier excel. Ainsi, nous pouvons donc visualiser les données de manière plus ordonnée et plus lisible. Cette syntaxe nous permet de passer aux données brutes à la figure 22 aux données préparées et organisées en figure 24.

Comme nous pouvons le constater, les différentes entrées de notre jeu de données apparaissent clairement, et nous pouvons constater que notre jeu de données a 15 colonnes et 3229 lignes.

	#	ID TICKET	OPERATOR	INCIDENTS	LIAISON	CAUSES	INCIDENCE	HEURE_INCIDENCE	DATE _RECEPTION(CAMTEL)	HEURE _RECEPTION(CAMTEL)	DATE_FIN
0	1	nexttel24	NEXTTEL	I4	Douala-Mbanga	ATR	24/08/2017	11:36:00	2017-08-24 00:00:00	12:14:00	2017-08
1	2	SMC/00001087710	MTN	I4	Douala-Mbanga	ATR	24/08/2017	11:34:00	2017-08-24 00:00:00	12:14:00	2017-08
2	3	INC000000098099	MAINONE	I6	Kribi-Kousseri	DGT	24/08/2017	03:27:00	2017-08-24 00:00:00	03:27:00	2017-08
3	4	SMC/00001090120	MTN	I4	Abong-Mbang ***** Ayos [MTN]	ATR	30/08/2017	15:46:00	2017-08-30 00:00:00	16:35:00	2017-08
4	5	1708T51802	OCM	I4	Ayos ***** Abong-MbangII [OCM]	ATR	30/08/2017	15:46:00	NaN	NaN	2017-08

Figure 24: Visualisation des données collectées dans Jupyter Notebook

I-4. Sélection et entraînement du modèle

Elle est subdivisée en deux sous étapes :

- ✓ Le Data cleaning
- ✓ Le Feature engineering

I-4-1. Le Data Cleaning ou Nettoyage des données

Nous procédons ici à la suppression des valeurs aberrantes. Pour ce faire, Nous allons utiliser une syntaxe qui nous permettra de voir les cases contenant les valeurs aberrantes. Ceci se fait colonne après colonne. Nous pouvons par exemple par souci de simplicité prendre les colonnes 'DUREE' et 'SUIVI ACTION 01'. Nous avons donc les syntaxes et les résultats suivants :

Pour la Colonne 'DUREE'

print (df ['DUREE']) : avec print qui veut dire afficher ; df qui représente le jeu de données ou dataframe ; et entre crochets 'DUREE', qui représente la colonne dans laquelle on veut évaluer le nombre de valeurs aberrantes. Cette syntaxe nous permet d'avoir le résultat suivant :

```
print (df ['DUREE'])
```

```
0        420.0
1        422.0
2        264.0
3        614.0
4        576.0
         ...
3224     517.0
3225    -120.0
3226     540.0
3227   -2259.0
3228   -2358.0
Name: DUREE, Length: 3229, dtype: float64
```

Figure 25: visualisation des lignes ayant les valeurs aberrantes dans la colonne 'DUREE'

Nous pouvons constater ici que les lignes pointées en rouges (3225,3227 et 3228) contiennent des valeurs aberrantes et donc seront supprimées.

Pour la Colonne 'SUIVI ACTION 01', en procédant comme précédemment, nous avons :

print (df ['SUIVI ACTION 01']) qui est la même syntaxe que précédemment avec le nom de colonne qui a changé. Elle nous donne le résultat suivant :

```
1 print (df ['SUIVI ACTION 01'])
```

```
0                      NaN
1                      NaN
2                      NaN
3                      NaN
4                      NaN
             ...
3224          APPEL + MAIL
3225                   NaN
3226     Travaux CAMWATER
3227            MAIL+APPEL
3228                  MAIL
Name: SUIVI ACTION 01, Length: 3229, dtype: object
```

Figure 26: Visualisation des lignes à valeur aberrantes dans la colonne 'SUIVI ACTION 01'

Nous constatons qu'ici, les lignes 0 à 4 et la ligne 3225 contiennent des valeurs 'NaN', qui sont des valeurs aberrantes et donc ces lignes seront supprimées.

Toutefois, il arrive souvent que des valeurs aberrantes dans deux colonnes ne se trouvent pas toujours dans la même ligne. Ceci peut poser un problème, dans la mesure où la suppression des lignes sur la simple base selon laquelle elles contiennent des valeurs aberrantes peut considérablement réduire le jeu de données et rendre encore plus difficile la prédiction. La solution dans ce cas est de connaitre le nombre de valeurs aberrantes par colonne. Et en fonction de ce nombre on peut :

1. Soit laisser les lignes contenant les valeurs aberrantes sans les supprimer si leur nombre représente un faible pourcentage par rapport au nombre de valeurs total.
2. Soit voir dans quelle mesure supprimer la/les colonne(s) contenant les valeurs aberrantes si leur pourcentage est élevé. (approximativement 40%)

Les figures ci-dessous nous montre par exemple des données avec valeurs aberrantes et celles après suppression de ces valeurs aberrantes.

)N	CAUSES	INCIDENCE	HEURE_INCIDENCE	DATE _RECEPTION(CAMTEL)	HEURE _RECEPTION(CAMTEL)	DATE_FIN_INCIDENT	HEURE_FIN INCIDENT	DUREE	SUIVI ACTION 01	STATU
ng ze in.	DGT	2020-12-03 00:00:00	01:29:00	2020-03-13 00:00:00	03:17:00	2020-03-13 00:00:00	10:18:00	1969.0	la liaison FH de Biwong Bane est tr?s instable...	OK
ll [N]	TP	13/03/2020	09:23:00	2020-03-13 00:00:00	09:55:00	2020-03-14 00:00:00	15:23:00	1800.0	APPEL + MAIL	OK
C [M]	RG	13/03/2020	10:46:00	2020-03-13 00:00:00	10:55:00	2020-03-16 00:00:00	16:40:00	4674.0	APPEL + MAIL	OK
CM]	TP	13/03/2020	09:23:00	2020-03-13 00:00:00	10:46:00	2020-03-16 00:00:00	12:25:00	4502.0	APPEL + MAIL	OK
n [M]	TP	13/03/2020	09:23:00	2020-03-13 00:00:00	13:29:00	2020-03-13 00:00:00	17:45:00	502.0	appel + mail	OK
...	...	...	...	...	...	...	...	...	...	...
ga M]	CFO	29/03/2020	09:00:00	2020-03-29 00:00:00	09:43:00	2020-03-29 00:00:00	17:37:00	517.0	APPEL + MAIL	OK
ar[N]	TP	30/03/2020	06:23:00	2020-03-30 00:00:00	08:19:00	2020-04-01 00:00:00	04:23:00	-120.0	NaN	OK
a [M]	AH	28/03/2020	08:15:00	2020-03-28 00:00:00	06:46:00	2020-03-28 00:00:00	17:15:00	540.0	Travaux CAMWATER	OK
ja- N]	ATR	2020-01-04 00:00:00	06:39:00	2020-04-01 00:00:00	07:06:00	2020-03-30 00:00:00	17:00:00	-2259.0	MAIL+APPEL	Encours et Assign?es
*** M]	ATR	2020-01-04 00:00:00	08:18:00	2020-04-01 00:00:00	08:18:00	2020-03-30 00:00:00	17:00:00	-2358.0	MAIL	Encours et Assign?es

Figure 27: Données contenant les valeurs aberrantes et manquantes en rouge

```
df.tail(72)
```

)N	CAUSES	INCIDENCE	HEURE_INCIDENCE	DATE _RECEPTION(CAMTEL)	HEURE _RECEPTION(CAMTEL)	DATE_FIN_INCIDENT	HEURE_FIN INCIDENT	DUREE	SUIVI ACTION 01	STATU
*** N]	TP	2020-12-03 00:00:00	23:29:00	2020-03-12 00:00:00	23:43:00	2020-03-14 00:00:00	18:00:00	2551.0	mail,appel	OK
C [M]	RG	2020-12-03 00:00:00	23:30:00	2020-03-13 00:00:00	00:21:00	2020-03-14 00:00:00	19:00:00	2610.0	57km d?Ed? a cŸble optique BB2 coup? en terre ...	OK
ng ze in.	DGT	2020-12-03 00:00:00	01:29:00	2020-03-13 00:00:00	03:17:00	2020-03-13 00:00:00	10:18:00	1969.0	la liaison FH de Biwong Bane est tr?s instable...	OK
ll [N]	TP	13/03/2020	09:23:00	2020-03-13 00:00:00	09:55:00	2020-03-14 00:00:00	15:23:00	1800.0	APPEL + MAIL	OK
C [M]	RG	13/03/2020	10:46:00	2020-03-13 00:00:00	10:55:00	2020-03-16 00:00:00	16:40:00	4674.0	APPEL + MAIL	OK
...	...	...	...	...	...	...	...	...	...	...
ga M]	CFO	29/03/2020	09:00:00	2020-03-29 00:00:00	09:43:00	2020-03-29 00:00:00	17:37:00	517.0	APPEL + MAIL	OK

Figure 28: Données nettoyées (après suppression des données aberrantes)

La syntaxe permettant de faire cette vérification est la suivante :

df.isnull().sum()

Cette syntaxe veut dire, pour le jeu de données 'df', faire apparaitre le total de toutes les valeurs aberrantes de chaque colonne. On aboutit au résultat suivant :

```
df.isnull().sum()

#                          0
ID TICKET                  0
OPERATOR                   0
INCIDENTS                  0
LIAISON                    0
CAUSES                     0
INCIDENCE                  1
HEURE_INCIDENCE            6
DATE _RECEPTION(CAMTEL)  782
HEURE _RECEPTION(CAMTEL) 788
DATE_FIN_INCIDENT         14
HEURE_FIN INCIDENT        56
DUREE                     11
SUIVI ACTION 01         1727
STATU                      0
dtype: int64
```

Figure 29: visualisation des lignes aberrantes dans toutes les colonnes du dataframe

Nous constatons que 'HEURE_INCIDENCE' a 6 valeurs aberrantes. Nous nous souvenons que notre jeu de données a 3229 lignes. Nous voyons bien que ces valeurs sont insignifiantes pour que nous puissions décider de les supprimer.

Par contre, la colonne 'SUIVI ACTION 01' contient 1727 entrées aberrantes. Ce qui est assez élevé. Dans ce cas la solution serait de voir s'il est possible de ne pas utiliser cette colonne dans notre algorithme.

I-4-2. Le Feature Engineering

Le Feature Engineering est l'étape qui consiste à ignorer une ou plusieurs entrées qui n'ont pas une incidence pertinente dans la résolution de notre problème et de réduire notre jeu de données à l'essentiel.

Dans cotre cas précis, nous voulons prédire un incident sur le réseau de transport optique Camerounais. Il va sans dire que les entrées telles que :

'HEURE_INCIDENCE','DATE_RECEPTION(CAMTEL)','HEURE_RECEPTION(CAMTEL' 'DATE_FIN_INCIDENT', 'HEURE_FIN INCIDENT','DUREE SUIVI ACTION 01','STATU'

Sont des entrées moins pertinentes pour la prédiction des incidents que les entrées :

'OPERATOR', 'INCIDENTS','LIAISON','CAUSES','DUREE'.

Ceci se fera avec la syntaxe de figure 29

```
df.drop(['HEURE_INCIDENCE',  'DATE _RECEPTION(CAMTEL)','HEURE _RECEPTION(C
AMTEL)', 'DATE_FIN_INCIDENT', 'HEURE_FIN INCIDENT','DUREE SUIVI ACTION 01'
,'STATU'], axis=1,inplace=True)
df.head(100)
```

Figure 30: Syntaxe de suppression des entrées inutiles pour notre algortihme

Cette syntaxe donc permet de sélectionner les entrées qui sont inutiles et les supprimer. Nous passons d'un tableau à 15 entrées (figure 23), à un tableau à 5 entrées tel que nous le voyons à la figure 30. Les entrées à laisser et celle supprimées se font en fonction de l'objectif recherché et de 'l'état des données en notre possession'. En général c'est le Data scientist en collaboration avec les responsables du département qui définissent quelles données supprimer et quelles données laisser.

Nous avons donc désigné neuf(9) entrées qui n'ont pas d'incidence dans notre prédiction à savoir : 'HEURE_INCIDENCE','DATE_RECEPTION(CAMTEL)','HEURE_RECEPTION(CAMTEL) ', 'DATE_FIN_INCIDENT', 'HEURE_FIN INCIDENT','DUREE SUIVI ACTION 01', 'STATU'.

Nous voyons (figure 30) donc que les 5 entrées pertinentes qui nous restent sont : 'OPERATOR', 'INCIDENTS','LIAISON','CAUSES','DUREE'

	OPERATOR	INCIDENTS	LIAISON	CAUSES	DUREE
0	NEXTTEL	I4	Douala-Mbanga	ATR	420.0
1	MTN	I4	Douala-Mbanga	ATR	422.0
2	MAINONE	I6	Kribi-Kousseri	DGT	264.0
3	MTN	I4	Abong-Mbang ***** Ayos [MTN]	ATR	614.0
4	OCM	I4	Ayos ***** Abong-MbangII [OCM]	ATR	576.0
...	...	...	...	...	...
95	MAINONE	I6	Airtel Tchad	CFO	274.0
96	MTN	I6	DOUALA WACS BATOKE	TPP	264.0
97	NEXTTEL	I6	Bidzar ***** Figuil [Nexttel]	VD	268.0
98	OCM	I6	Ribao ***** Moutourwa [OCM]	VD	269.0
99	OCM	I6	Yaounde-PDC-New ***** Yaounde-CNDH [OCM]	TP	5579.0

Figure 31: Données après suppression des paramètres non pertinents

I-4-3. Le Model Selection ou Sélection du Modèle

Ici, il est question de choisir le modèle à utiliser pour faire la prédiction. Nous connaissons les différents algorithmes de prédiction qui existent tels que vus au chapitre 1. Dans notre étude, nous avons des données et il nous faut prédire les pannes. Prédire les pannes revient à dire quel type de panne nous avons. Et dire quel type de panne nous avons revient à classifier les différentes pannes. Ce qui nous conduit donc à un problème de classification.

Toujours Au chapitre 1 nous avons répertorié un tableau (Tableau 3) regroupant les différents algorithmes et leurs domaines d'utilisation. Sur la base de ce tableau, et du type de problème que nous devons résoudre (problème de classification), nous allons utiliser l'algorithme du KNN.

Mais avant de passer, notons que nous avons à faire ici à des données catégorielles (des textes). Or notre algorithme travaille avec des valeurs numériques. Pour pallier à ce problème, nous allons utiliser une méthode de transformation des valeurs catégorielles en valeurs numériques : Le label Encoder.

L'implémentation du Label Encoder sur Jupyter notebook nous donne ceci :

```
fromsklearn.preprocessingimportLabelEncoder
le =LabelEncoder()
df2['OPERATOR_L'] =le.fit_transform(df.OPERATOR.values)
df2['LIAISON_L'] =le.fit_transform(df.LIAISON.values)
df2['CAUSES_L'] =le.fit_transform(df.CAUSES.values)
df2['INCIDENTS_L'] =le.fit_transform(df.INCIDENTS.values)
```

Figure 32: transformation par le LabelEncoder des catégorielles en valeurs numériques

Cette syntaxe permet de transformer les entrées OPERATOR, LIAISON, CAUSES, INCIDENTS qui sont des entrées catégorielles en entrées OPERATOR_L, LIAISON_L, CAUSES_L, INCIDENTS_L qui sont des entrées numériques , qui peuvent être exploitées par notre algorithme de prédiction.

Après la transformation des données catégorielles, nous devons procéder à l'entrainement des données.

La syntaxe ci-dessous (figure 32) nous permet de faire l'entrainement des données.

Pour ce faire, nous devons diviser notre jeu de données en deux lots : les données d'entrainement (X_train, y_train), et les données de test (X_test, y_test). Ceci est d'autant plus important que L'utilisation du même ensemble de données pour l'entraînement et les tests laisse place à des erreurs de calcul, augmentant ainsi les risques de prédictions inexactes.

Etant donné que cette division ne peut se faire manuellement, nous faisons appel à la fonction train_test_split, qui est est une fonction de la sélection de modèle Sklearn permettant de diviser les tableaux de données en ces deux sous-ensembles qui sont les données d'entraînement (X_train, y_train), et les données de test (X_test, y_test). Cette fonction train_test_split vous permet de casser facilement un ensemble de données tout en poursuivant un modèle idéal. De plus, le modèle ne doit être ni surajusté ou sous-ajusté. Or, l'utilisation du même ensemble de données pour l'entraînement et les tests laisse place à des erreurs de calcul, augmentant ainsi les risques de prédictions inexactes. Pour éviter cela, nous devons définir un état aléatoire, qui garantit que les divisions générées sont reproductibles. Scikit-learn utilise des permutations aléatoires pour générer les divisions. Cet état aléatoire est utilisé comme une graine pour le générateur de nombres aléatoires, garantissant que les nombres aléatoires sont générés dans le même ordre.

```
X_train, X_test, y_train, y_test=train_test_split( X, Y, test_size=0.10,
random_state=4)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

Figure 33: syntaxe de l'entrainement et du test des données

Ce qui nous donne comme résultat :

```
Train set: (2906, 4) (2906,)
Test set: (323, 4) (323,)
```

Figure 34: Résultat

La syntaxe de la figure 32 nous donne le résultat à la figure 33. Nous voyons bien que notre tableau de données à bel et bien été divisé en données d’entrainement (Train set) et données de test (Test set). Et nous voyons que les données d’entrainement sont de 2906 et les données de test de 323, soit 10% des données d’entrainement, qui est bel et bien ce que nous avions précisés dans la syntaxe de la figure 38 (test_size = 0.10).

Nous pouvons, maintenant que nous avons les données d’entrainement et de test, passer à l’implémentation du KNN.

L’implémentation du KNN nous donne donc :

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
classifier =KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train,y_train)
```

Figure 35: implémentation de l’algorithme du KNN dans Jupyter notebook

La première étape dans l’implémentation du KNN consiste à importer la classe KNeighborsClassifier à partir de la bibliothèque sklearn.neighbors. ensuite l’initialiser avec un paramètre, le n_neigbours. Il s'agit essentiellement de la valeur pour le K, ici 5.
À la figure 35, nous prédisons la sortie de notre algorithme.

```
y_pred=classifier.predict(X_test)
y_pred  [0:5]
array([5., 5., 5., 5., 5.])
```

Figure 36: Visualisation de la prédiction

Puis nous prédisons la sortie de notre algorithme en fonction des données de test. Et nous demandons à l'algorithme de donner les cinq valeurs possibles de prédiction qu'il a.

D'après notre implémentation du label encoding, l'algorithme nous prédit une *« indisponibilité du signal »*.

Par ailleurs, pour tout algorithme dans l'apprentissage automatique, il serait bon de connaitre la précision de la prédiction, car elle permet de connaitre à quel point notre algorithme est précis, et donc à quel point nous pouvons-nous fier à lui. Plus la précision de l'algorithme est grande, plus il est fiable, et plus nous pouvons l'utiliser pour les systèmes de production.

Nous obtenons l'implémentation suivante :

```
cnf_matrix=confusion_matrix(y_test, y_pred)
np.set_printoptions(precision=2)

print (classification_report(y_test, y_pred))
```

Figure 37: Syntaxe pour la précision de l'algorithme du KNN sur le jeu de données

```
precisionrecall  f1-score   support

         4.0       0.57      0.56      0.56       135
         5.0       0.67      0.71      0.69       182
         6.0       0.00      0.00      0.00         6

    accuracy                           0.63       323
   macro avg       0.41      0.42      0.42       323
weightedavg       0.62      0.63      0.62       323
```

Figure 38: Précision de l'algorithme du KNN sur le jeu de données

Nous demandons par cette syntaxe à l'algorithme de nous donner son degré de précision, avec deux chiffres après la virgule.

Nous affichons les éléments sus-cités, avec comme demandé notre précision de 0.63 qui est bel et bien de deux chiffres après la virgule.

Toutefois, il est à noter qu'en améliorant les performances de notre algorithme, nous obtenons une précision de 65%., comme le montre la capture d'écran ci-dessous (figure 38). Cette amélioration se fait par le choix du nombre de voisins qui permettrait d'avoir la meilleure précision. Nous obtenons l'implémentation à la figure 38.

```
print( "The best accuracywaswith", mean_acc.max(), "with k=", mean_acc.argmax()+1)
```

```
The best accuracy was with 0.653250773993808 with k= 12
```

Figure 39: Visualisation de la précision améliorée

Conclusion

Tout au long de ce chapitre, nous avons utilisé l'Intelligence Artificielle pour prédire les pannes qui peuvent surgir sur un réseau télécommunications en fibre optique. Nous avons démontré comment à partir d'un jeu de données collectées dans un fichier CSV ou XLS, on implémente un algorithme d'apprentissage automatique et on étudie ainsi la fiabilité d'un système, en occurrence un réseau de télécommunications dans notre cas d'espèce. La fiabilité de notre algorithme était de 63% et a été améliorée à 65%. Ceci montre que grâce à l'IA, le problème à résoudre, celui de la prédiction d'une panne pouvant surgir sur un réseau de télécommunication en fibre optique a été fait au moyen de l'apprentissage automatique en se servant de l'historique des pannes qui surgissaient dans ce réseau.

Par ailleurs, il est à noter qu'en apprentissage automatique la précision des algorithmes dépend d'autres paramètres comme:

- ✓ Le choix de l'algorithme utilisé (KNN,SVM,etc…), En effet, il peut arriver que pour un même jeu de données, on obtien un meilleur pourcentage avec un algorithme X qu'avec un algorithme Y;
- ✓ La manière d'entrainner les données (choix judicieux entre la quantité des données d'entrainnement et de données de test). Il peut arriver que le choix d'un pourcentage de données d'entrainnement qui n'est pas suffisant donne une mauvaise fiabilité.

- ✓ De la procédure de collecte des données
- ✓ De la corrélation qui existe entre ces données ; ce qui veut dire que lors de l'élaboration des paramètres à prendre en compte lors de la récolte des données doivent être choisis avec soin afin de mieux étudier la fiabilité de ce système et d'aboutir à de meilleurs résultats.

CONCLUSION GENERALE ET PERSPECTIVES

CONCLUSION

L'objectif de notre travail était d'étudier la fiabilité à base de l'intelligence artificielle. Dans un contexte marqué par l'absence de bonnes politiques solides de maintenance, de problèmes ralentissant le bon fonctionnement de notre industrie et engendrant de nombreuses pertes causées par des pannes récurrentes, par des temps d'arrêt élevés et par des détections tardives des incidents, il est devenu urgent de trouver des solutions pour faire face à cela ! Le choix de l'intelligence artificielle parmi tant d'autres possibilités s'est offert à nous : si l'intelligence artificielle est déjà utilisée en médecine pour le diagnostic, et la prévention de certaines maladies, pourquoi ne pas l'utiliser aussi en industrie pour le diagnostic et la prédiction des défaillances, bref pour étudier la fiabilité, maintenabilité et la prévention des défaillances des équipements ? Vu la vaste étendue de l'intelligence artificielle et ses nombreux démembrements, nous avons utilisé l'apprentissage supervisé. Nous avons procédé à la collecte des données, à leur nettoyage, à la sélection des paramètres pertinents pour notre étude avant de choisir parmi les algorithmes de l'apprentissage supervisé, l'algorithme qui nous permettait d'avoir les meilleurs résultats à savoir l'algorithme du KNN. Cette étude à permi d'arriver à une prédiction de défaillances avec une précision de 65%. C'est la preuve qu'on peut bien utiliser l'intelligence artificielle dans notre industrie pour réduire le temps d'arrêt des équipements, prédire les défaillances et même détecter à temps les défaillances. Nous pouvons aussi retenir que, l'intelligence artificielle est un excellent outil qui, utilisé convenablement participerait grandement et à moindre coût à l'essor de notre industrie. Il faut aussi noter que ce travail est une méthodologie qui permet à toute personne désireuse d'utiliser le Machine Learning et particulièrement l'apprentissage supervisé de le faire en suivant juste les grandes étapes mentionnées dans ce travail. Ainsi, que ce soit la prédiction des défaillances dans un réseau électrique, dans un système industriel, la méthodologie restera la même.

PERSPECTIVES

La précision de notre algorithme (65%) prouve que l'IA peut bien être utilisée en industrie pour prédire une panne, comme en médecine pour diagnostiquer une maladie. En général, pour une prédiction en IA (machine learning qui est la branche utilisée pour notre travail), il serait plus intéressant d'avoir une précision de l'ordre de 80% . Mais ce premier pas est pour nous une avancée. Toutefois, en vue de perfectionner ce travail et le rendre d'avantage exploitable, Nous suggérons donc que des études soient menées en utilisant d'autres branches de l'IA, telles que l'apprentissage profond, qui certainement pourrait nous donner de meilleurs résultats. Pour ce qui est de l'apprentissage automatique, nous suggérons que pour une meilleure précision, on pourrait revoir la méthode de récolte des données, c'est-à-dire affiner les informations à recueillir pour étudier la fiabilité du système.

REFERENCES BIBLIOGRAPHIQUES

[1] www.axiocode.com, consulté le 30 Mars 2021

[2] www.experiences.microsoft.fr, consulté le 30 Mars 2021

[3] www.towardsdatascience.com

[4] www.cbinsights.com

[5] https://theconversation.com/

[6] www.sas.com

[7] www.diegocalvo.es

[8] datakeen.co

[9] https://towardsdatascience.com/

[10] https://datascientest.com/

[11] pythonistaplanet.com

[12] https://www.ibm.com/cloud/learn/unsupervised-learning

[13] https://www.ibm.com/cloud/learn/unsupervised-learning

[14] Stuart Russell and Peter Norvig ; Artificial Intelligence: A Modern Approach

[15] Aurélien Géron ; Hands-On Machine Learning withScikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition

[16] Chistopher Bishop ; Pattern Recognition And Machine Learning - Springer 2006

[17] Richard S. Sutton and Andrew G. Barto ; Reinforcement Learning: An Introduction Second edition, in progress 2014, 2015

[18] Agathe Mercante, Les échos,Décembre 2017

[19] Docteur Voufo, Diagnostic des défauts dans les réseaux HTA des pays subsahariens en présence de GED

[20] Avrim Blum, John Hopcroft, and RavindranKannan, Foundations of Data Science

[21] Elizabeth Matsui and Roger D. Peng ,The Art of Data Science: A Guide for Anyonewho Works ,2016

[22] Jake VanderPlas, Python Data Science Handbook: Essential Tools for Workingwith Data ,2016

[23] Andrew Ng, Machine Learning Engineering for Production,2021

[24] www.un.org

[25] Carlos Lopes,L'Afrique est l'avenir du monde :Repenser le développement, Mars 2021