République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
École Nationale Supérieure de Technologie
Département de Génie Industriel et Maintenance

Mémoire pour l'obtention du diplôme en vue de l'obtention de

MASTER

Domaine : **SCIENCE ET TECHNOLOGIE**

Filière : **MAINTENANCE INDUSTRIELLE**

Spécialité : **MANAGEMENT ET INGÉNIERIE DE MAINTENANCE INDUSTRIELLE**

- Thème -

Mise en place d'un outil d'aide à la décision d'une fonction de maintenance : basé sur des algorithmes de classification supervisée avec des données à plusieurs dimensions.

Présenté par

GHERIB Imane & GUEHTAR Lina Achouak

Mme H.HACHEMI,	**M.C.B**	ENST, Alger	**Président**
Mme N.SALHI,	**M.A.A**	ENST, Alger	**Examinateur**
Mr W.REZGUI,	**M.C.B**	ENST, Alger	**Examinateur**
Mr A.AMRANE	**M.C.B**	ENST, Alger	**Encadrant**
Mr A.CHABANE	**Doctorant**	ENST, Alger	**Co-Encadrant**

Année Universitaire 2020 / 2021

Remerciements

Nous remercions tout d'abord "ALLAH" tout puissant qui nous a donnée toute la patience et l'aide pour réaliser notre travail.

Nous souhaiterions ici remercier nos professeurs et en particulier notre encadrant de mémoire Monsieur A.AMRANE, ainsi que notre co-encadrant A.CHABANE, qui nous ont permis de mener à bien cette recherche grâce à leurs conseils et à leurs écoutes et leurs confiances qu'ils ont eue à notre égard tout au long de la réalisation de notre mémoire.

Nos remerciements vont également à nos parents ; à tout ceux qui nous ont aidé de près ou de loin à réaliser notre travail "FENINEKH Chaima","CHEBILA Asma", "MAMMAR Khaire Eddine".

Tous nos remerciements aux membres de jury d'avoir accepté d'évaluer ce travail.

Enfin, nous sommes reconnaissantes d'avoir pu mener à bien cette étude conjointement, elle ne s'en trouve que plus riche et plus fournie. Travailler ensemble nous a permis de confronter nos points de vue et sans cela notre recherche n'aurait sans doute pas être arracher.

Résumé

Ce mémoire présente une démarche pour la classification supervisée des paramètres de maintenance de la société GICA, par les méthodes : K-PPV, SVM et FA. La première étape est le prétaitement des données, qui est suivie par l'application des méthodes de la classification. Ensuit, un algorithme comparatif a été proposé afin de choisir la bonne méthode. Enfin, nous avons simplifié l'utilisation de ces méthodes par la proposition d'une interface graphique.

Abstract

This thesis presents an approach for the supervised classification of the maintenance parameters of the GICA company, by the methods : K-PPV, SVM and FA. The first step is the data preparation, which is followed by the application of the classification methods. Then, a comparative algorithm was proposed in order to choose the right method. Finally, we have simplified the use of these methods by proposing a graphical interface.

Abstrakt

Diese Arbeit stellt einen Ansatz zur überwachten Klassifizierung der Instandhaltungsparameter der Firma GICA nach den Methoden K-PPV, SVM und FA vor. Der erste Schritt ist die Datenaufbereitung, gefolgt von der Anwendung der Klassifikationsmethoden. Anschließend wurde ein Vergleichsalgorithmus vorgeschlagen, um die richtige Methode auszuwählen. Schließlich haben wir die Verwendung dieser Methoden vereinfacht, indem wir eine grafische Oberfläche vorgeschlagen haben.

Dédicaces

Dans l'effort que nous faisons pour comprendre le monde, nous ressemblons quelque peu à l'homme qui essaie de comprendre le mécanisme d'une montre fermée. Il voit le cadran et les aiguilles en mouvement, il entend le tic-tac, mais il n'a aucun moyen d'ouvrir le boîtier. S'il est ingénieux, il pourra se former quelque image du mécanisme, qu'il rendra responsable de tout ce qu'il observe, mais il ne sera jamais sûr que son image soit la seule capable d'expliquer ses observations.

Albert Einstein et Léopold Intel

Je dédie ce travail à deux personnes :

A la perssonne, qui sans elle je ne serai jamais arrivée à ce palier, encouragée ...à ma mère.
Et à mon père pour son aide et son soutien.
Mon frère : Zakaria.
Ma soeur : Allae Raoua .
A mon amie GUEHTAR Lina .

Pour leurs affection et leurs sacrifices afin que rien n'entrave le déroulement de mes études.

GHERIB Imane.

Liste des Symboles et Abréviations

Les principales notations et abréviations utilisées sont explicitées ci-dessous, sous leur forme la plus couramment employée dans le domaine du génie électrique.

Liste des Abréviations

AA :	Apprentissage Automatique.
ACP :	Analyse des Composants Principaux .
AD :	Arbre de Décision.
API :	Application Programming Interface.
CH :	classification Hiérarchique.
CT :	Classes Théorique.
Déf :	Défaillance.
D :	Disponibilité.
ERP :	Entreprise Ressources Planing.
FA :	Forêts Aléatoires
GMAO :	Gestion de Maintenance Assistée par Ordinateur.
GUI :	Interfaces Utilisateur Graphiques.
IA :	Intelligence Artificiel.
K-PPV :	K Plus Proche Voisin.
MTBF :	Temps Moyen entre Pannes.
MTTR :	Temps Moyen de Réparation.
NTIC :	Nouvelle Technologie de l'Information et des Communications.
OCR :	Reconnaissance des Caractères Optiques.
P :	Performance.
PGI :	Progiciels de Gestion Integrée.
R :	Réparation.
RH :	Ressources Humaines.
RNA :	Réseaux de Neurones Artificiel.
SVM :	Machine vecteur de support.
SVR :	Vecteur Regression de Support.
TLN :	Traitement du Langage Naturel.

Liste des Symboles

K :	Constante définie par l'utilisateur.
R :	Langage de Programmation.
Θ :	L'ensembles des classes.
λ :	Défaillance.
μ :	Réparation.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

La maintenance industrielle était considérée, au cours du 19ème siècle, comme une activité d'arrière-plan. Les fonctions étaient limitées juste aux interventions pendant des pannes électriques, mécaniques ou de graissage. Les notions de prévisions ou de préventions n'avaient pas encore fait surface, car le monde industriel n'était pas ce qu'il est aujourd'hui, et il les implications étaient bien différent.

En effet, efficacité de la maintenance des systèmes industriels est un enjeu économique majeur pour leur exploitation commerciale. Les principales difficultés et sources d'inefficacité résident dans le choix des actions de maintenance. Une action de maintenance consiste à remplacer les équipements du système qui sont en panne qui ne sont plus capables de réaliser leur fonction. Les opérations de maintenance sont coûteuses pour plusieurs raisons.

Tout d'abord, elles nécessitent souvent un arrêt de fonctionnement du système. Dans ce cas, durant toute la phase de maintenance, le système n'est pas opérationnel. Plus la phase de maintenance est longue, plus elle est coûteuse dû à l'indisponibilité du système. Par conséquent la phase de maintenance doit idéalement être réduite aux opérations consistant à remplacer, sans tâtonnement, les équipements réellement en panne. La décision d'une action de maintenance est très complexe et doit reposer sur une surveillance et une analyse intelligente de l'état du système.

La seconde raison pour laquelle une maintenance peut être coûteuse concerne les cas d'urgence dans lesquels la sécurité ou l'accomplissement de la fonction du système sont mis en jeu. En effet, lorsqu'un équipement tombe soudainement en panne et que le système ne peut plus réaliser sa fonction, des actions de maintenance doivent être automatiquement réalisées pour remettre le système en état de fonctionnement. Ces actions imprévues sont naturellement plus coûteuses car les besoins et services pour la maintenance n'ont pas été anticipés et doivent être rapidement disponibles. L'objectif général de notre recherche en rapport à ce problème est de minimiser l'occurrence de ce genre de situation, une maintenance préventive peut être envisagée.

Outre l'objectif général, l'intérêt de ce travail est d'anticipé et corrigée les pannes des équipements, avant de générer de trop importants dégâts qui pourraient provoquer un arrêt imprévu du système.

Par ailleurs, l'apparition de systèmes complexe nécessite un besoin croissant en traitement

de données. Dans ce contexte, des bases de données ont alors été construites dans le but de gérer toutes les taches du service de maintenance. Au fil des années ; le volume des données augmente en permanence et à une grande vitesse, et pourrait contenir plusieurs milliers d'éléments, avec plusieurs paramètres de classification et ayant des natures très diverses.

Cependant, la variété et la nature des différents éléments et leur évolution dans le domaine de la maintenance industrielle devrez amener à étudier leurs classifications. Pour cela, un accent particulier a été mis dans ce travail sur la classification supervisée, et à la mise en place d'une démarche de classification pour aider le service de maintenances dans la prise de décision.

Dans le contexte précèdent, nous nous intéressons à lister et caractériser les différentes méthodes de classifications existantes afin de proposer une classification des différents paramètres de maintenance. Deux critères de cette classification s'imposent : l'algorithme et les caractéristiques des méthodes utilisées. Deux approches sont identifiées, à savoir, les méthodes supervisé et non supervisés. Les données seront analysées, pour choisir laquelle des classification sera utilisée en se basant valeur des paramètres de l'historique. De plus, la base de données sera multi-paramètres, au minimum quatre paramètres, ce qui rend les méthodes conventionnelles inefficace. Afin de pallier au dernier problème, une étude sur les différentes méthodes susceptibles d'être utilisées sera faite pour réduire et exploiter les données avant de pouvoir les traiter.

Dans le but d'atteindre efficacement les objectifs fixés par le travail, ce manuscrit suit l'organisation suivante :

Le premier chapitre présente le cadre de l'intelligence artificielle et l'apprentissage automatique. Les types de l'apprentissage automatique et de la classification sont introduits. Les différents algorithmes existants pour résoudre ces problèmes sont décrits. Ces algorithmes s'appuient, sur une connaissance approfondie des paramètres de la base de données.

Le deuxième chapitre décrit les méthodes utilisés dans le cadre ce projet ainsi que les étapes à poursuivre pour réaliser le pré-traitement des données collectées.

Le troisième chapitre traite de l'architecture et de la mise en oeuvre des méthodes introduits dans le deuxième chapitre. Ce chapitre consiste à présenter, à analyser et à interpréter les résultats de notre outil. De plus, il permettra d'analyser les résultats et présenter l'interface graphique mise en place.

Enfin, la conclusion générale du mémoire synthétise les apports de notre travail. Les limites et les perspectives de recherche futures sont également discutées.

Chapitre I

Généralité sur la classification

"Quand le bateau est au milieu du fleuve, il est bien tard pour réparer l'avarie".

Proverbe chinois

I.1 Introduction

De nos jours, les entreprises doivent répondre de plus en plus aux exigences en termes de qualité, de quantité de produits et de services, de réactivité et de réduction des coûts. Dans ce contexte, les entreprises doivent disposer d'un système de production fiable, pour ce faire, le service de maintenance nécessite un système efficace et moindre coût. De plus, un service de maintenance performant et organisé contribue au bon fonctionnement du système de production, il permet d'augmenté la durée vie des équipements industriels et ainsi participé à l'optimisation des performances globales.

Depuis quelques décennies, une phase de structuration et de normalisation des services de maintenance se met en place. De plus l'évolution des marchés, leur mondialisation et l'accent sur le profit et la compétitivité de l'entreprise provoquent la mise en place de nouveaux concepts concernant l'organisation de la production en particulier l'organisation de la maintenance.

Les systèmes informatiques industriels regroupent les applications de gestion et leurs éléments d'accompagnement, les logiciels et le matériel. Le développement des systèmes informatiques industriels dans le domaine de la maintenance industrielle a commencé lorsque celle-ci a été reconnue comme fonction fondamentale dans l'entreprise et une réflexion particulière a été mise sur l'étude approfondie et le développement des procédures de cette fonction.

A ce propos, le présent chapitre a pour objet de présenter l'informatisation de la maintenance, l'intelligence artificielle et l'apprentissage automatique, en tenant en compte, les différents algorithme utilisés dans le cadre de l'apprentissage automatique . Pour cela nous proposons l'utilisation de ces algorithmes dans la classification des paramètres de maintenance.

I.2 L'informatisation de la maintenance

Le terme « informatisation de la maintenance » désigne l'utilisation de l'informatique dans la maintenance et regroupe l'ensemble des technique de conception, d'analyse et de programmation des services de maintenance. Constituant un domaine très vaste et avec de multiple applications dans l'industrie. En effet, d'après la littérature existante l'évolution des systèmes informatiques en maintenance ont passé par plusieurs étapes qui sont ([19], [8]) :

- **Informatisation des procédures de maintenance :** Plusieurs procédures de maintenance ont été informatisées grâce à l'informatisation et l'automatisation de la gestion de entreprises. L'intégration des fichiers informatiques des équipements, des interventions, des stocks, des plans et schémas et l'automatisation des activités de la maintenance ont été possibles grâce aux progiciels de GMAO (Gestion de Maintenance Assistée par Ordinateur). Les événements quotidiens de la maintenance ont été traités : la panne, l'exécution du préventif, la gestion des stocks[29].
- **Interfaçage avec des progiciels :** Par la suite, ces progiciels ont dû s'interfacer avec les autres logiciels de l'entreprise tels que les achats et la comptabilité, déjà informatisés précédemment. Les grands progiciels de gestion intégrée (PGI) correspondant au sigle ERP en anglais (Enterprise Resource Planning) représentent une étape suivante dans la rationalisation des processus de l'entreprise et dans l'intégration de la maintenance avec les autres fonctions de l'entreprise [3].
- **Evolution du domaine technique :** L'informatique a aussi progressé dans le domaine technique de la maintenance. Les techniques modernes d'analyse de maintenance et de contrôle ont vu le jour parallèlement à l'informatique : analyse vibratoire, analyse d'huile, thermographie IR, ultrasons à chaud, etc [21].
- **Intégration de modules intelligents en architectures de maintenance :** La présence de ces différents modules intelligents de maintenance nous amène à les faire communiquer et collaborer entre eux. Cette construction de modules ou briques intelligentes doit concourir à donner des indicateurs pour prendre la bonne décision en matière de stratégie et de politique de maintenance [22].
- **Développement des NTIC :** Le développement de nouvelles technologies de l'information et des communications, l'extension d'Internet dans l'entreprise, l'intégration des applications, l'émergence de nouvelles politiques de maintenance indiquent une nouvelle période pour l'informatisation de la maintenance, celle que certains appellent la « maintenance intelligente ». Cela nous amène vers des architectures coopératives et distribuées des systèmes de maintenance communiquant entre eux ou sur une base de réseaux. L'implémentation de ces architectures de maintenance peut se faire à l'aide de plateformes de maintenance dont l'idée majeure est de proposer un service de maintenance via internet. Les plateformes de maintenance proposées dans les projets Proteus peuvent servir d'exemples. [31]

Anticiper une défaillance matérielle, prévoir une interruption de service, détecter un défaut sur une pièce en service offrent aux entreprises ou collectivités la capacité de réagir et d'éviter des incidents aux conséquences parfois dramatiques.

Le calcul du temps moyen entre pannes (MTBF) est classique depuis de nombreuses années et est historiquement basé sur des modèles statistiques. L'apport de l'Intelligence Artificielle est d'offrir une plus grande finesse dans la détermination des pannes, une anticipation prenant en compte des facteurs « temps réels » voire de détecter des signaux faibles ou de découvrir des cas non répertoriés.

I.3 L'intelligence artificielle et l'apprentissage automatique

I.3.1 L'intelligence artificielle

Ces dernière années, les algorithme ont évaluer et la puissance informatique a grimpé. Profitant de ces amélioration les chercheurs obtiennent des résultats étonnants avec les algorithme utilisant des capacités de traitement puissantes.

L'intelligence artificielle (IA), l'une de ces deniers algorithmes, peut être définie comme la capacité d'une machine à exécuter les fonctions cognitives normalement associées à l'esprit humain, comme la perception, le raisonnement, l'apprentissage et la résolution de problèmes. L'IA n'est pas une technologie unique associée à une application particulière, mais un ensemble de technologies possédant des applications générales dans tous les domaines. La robotique, la vision par ordinateur, le traitement du langage naturel (TLN) et l'apprentissage automatique (AA) sont des exemples de technologies permettant à l'IA de résoudre des problèmes de nature professionnelle.

L'apprentissage automatique constitue un sous-ensemble de l'intelligence artificielle qui est elle-même un sous-ensemble de la science des données ; il s'agit en fait simplement d'une technique de réalisation de l'intelligence artificielle([30], [11]).

I.3.2 L'apprentissage automatique

L'apprentissage automatique (machine learning) est un élément à ce point important de l'intelligence artificielle que les deux termes sont souvent utilisés de manière interchangeable.

L'intelligence artificielle comprend toutes les technologies, y compris l'apprentissage automatique, qui permettent à la machine d'imiter l'intelligence humaine.

En fait, L'apprentissage automatique est une méthode d'entraînement d'algorithmes pour permettre à ces derniers d'apprendre à prendre des décisions et à faire des prévisions sans recevoir d'instructions découlant d'une programmation explicite.

Il s'agit donc de fournir de très nombreuses données à un algorithme et de lui permettre d'en apprendre plus sur les informations traitées.

Les algorithmes d'apprentissage automatique recherchent des modèles naturels dans les données afin de produire des informations et de pouvoir prendre de meilleures décisions et faire de meilleures prédictions. Ils servent chaque jour à prendre des décisions critiques en matière de diagnostic médical, d'opérations d ?actions, de prévisions des charges énergétiques, etc.

En ce qui concerne l'apprentissage automatique, la partie la plus importante du processus de formation consiste à acquérir suffisamment de données de qualité pour tester le modèle. De plus, l'entraînement est un processus continu et itératif, fournissant à chaque étape davantage de données ou affinant le modèle. L'ensemble du cycle d'apprentissage automatique peut être résumé comme suit :

1. **Acquisition des données :** La première étape consiste à obtenir des données pertinentes pour l'application à développer. Les données doivent être de grande qualité et détaillées.
2. **Préparation des données :** Cette étape est également appelée nettoyage des données. Les données doivent être précises, propres et sécurisées.
3. **Sélection de l'algorithme :** L'algorithme le plus approprié pour l'application à développer doit être choisi.
4. **Entraînement du modèle :** Un modèle d'apprentissage automatique est une représentation mathématique d'un processus réel. L'algorithme retenu doit être entraîné sur les données pour créer le modèle. Le processus d'entraînement peut être supervisé, non supervisé ou renforcé. La description détaillée de ce processus est fournie dans d'autres sections de la présente monographie.
5. **Évaluation :** Le modèle doit être évalué pour s'assurer que l'algorithme retenu est le mieux adapté.
6. **Déploiement :** Il faut décider si le modèle doit être déployé dans le nuage informatique ou sur place.
7. **Test :**Le modèle doit être testé avec des données nouvelles et pour faire des prédictions.
8. **Évaluation :** La validité des prédictions établies par le modèle doit être évaluée, et le raffinement des données, du modèle et de l'algorithme doit être mis en oeuvre selon qu'il convient. [20]

I.4 Les types d'apprentissage

L'apprentissage automatique est axé sur l'analyse prédictive et prescriptive, en fonction de la nature de l'analyse et des algorithmes utilisés. Cette section donne un aperçu des types les plus courants d'apprentissage automatique, comme le montre la figure.

I.4.1 Apprentissage supervisé

Les algorithmes d'apprentissage supervisé font des prévisions en fonction d'exemples, par exemple un historique de vente pour déterminer des prix futurs. Dans un tel cas, il y a une

variable d'entrée composée de données d'entraînement étiquetées et d'une variable de sortie souhaitée. Un algorithme est utilisé pour analyser les données d'entraînement afin d'apprendre la fonction qui associe l'entrée à la sortie. Cette fonction permet de procéder à une mise en correspondance de nouveaux exemples en généralisant à partir des données d'entraînement pour anticiper les résultats [16].

I.4.1.1 Problème résolu par apprentissage supervisé

L'apprentissage supervisé sert à résoudre les problèmes suivants :

- **Classification** La classification, telle qu'elle est définie en analyse de données, consiste à regrouper des ensembles d'exemples en classes. Ces classes sont souvent organisées en une structure (clustering).
- **Régression :** Lorsqu'on procède à la prédiction de valeurs continues, on parle de régression.

I.4.1.2 Algorithmes d'apprentissage supervisé

Les algorithmes d'apprentissage supervisé les utilisés dans la littérature sont [23] :

- **La régression linéaire (régression) :** Est la régression de base. La régression linéaire simple nous permet de comprendre les relations entre deux variables continues.
- **La régression logistique (classification) :** Consiste à déterminer la probabilité qu'un événement se produise en fonction de données fournies.
- **L'analyse discriminante linéaire (classification) :** Est un algorithme de classification traditionnellement limité aux problèmes de classification à deux classes. Lorsqu'il y a plus de deux classes, l'algorithme d'analyse discriminante linéaire constitue la technique de classification linéaire privilégiée.
- **La classification naïve bayésienne (classification) :** Est basée sur le théorème de Bayes, classe chaque valeur indépendamment de toute autre valeur. Elle permet de prédire une classe / catégorie en fonction d'un ensemble donné de caractéristiques et à l'aide de probabilités.
- **Machine à vecteurs de support (classification) :** La machine à vecteurs de support peut être utilisée à la fois pour les tâches de régression et les tâches de classification. Elle est cependant largement utilisée à des fins de classification.
- **Les arbres de décisions (classification / régression) :** Un arbre de décisions est une structure arborescente de type organigramme qui illustre tous les résultats possibles d'une décision à l'aide d'une méthode de branchement. Chaque noeud de l'arborescence représente un test lié à une variable spécifique, et chaque branche en est le résultat.
- **Les forêts aléatoires (classification / régression) :** La forêt aléatoire est une méthode d'apprentissage qui combine plusieurs algorithmes pour obtenir de meilleurs résultats en matière de classification, de régression et pour d'autres tâches. Individuellement,

les classifieurs sont faibles ; cependant, lorsqu'ils sont combinés les uns aux autres, ils peuvent produire d'excellents résultats.

- **Les plus proches voisins (classification) :** L'algorithme des plus proches voisins évalue la probabilité selon laquelle une donnée appartient à tel ou tel groupe. Il examine des données afin de déterminer le groupe auquel la donnée en question appartient.
- **La méthode AdaBoost (classification) :** La méthode AdaBoost est une technique d'ensemble qui tente de créer un classifieur fort à partir d'un certain nombre de classifieurs faibles en construisant un modèle à partir des données d'apprentissage, puis en créant un second modèle pour essayer de corriger les erreurs du premier modèle.

I.4.2 Apprentissage semi-supervisé

Dans l'apprentissage supervisé, l'étiquetage des données peut être long et coûteux. Si les étiquettes sont limitées, il est possible d'utiliser des exemples non étiquetés pour améliorer l'apprentissage supervisé. Il s'agit donc d'un mixe entre l'apprentissage supervisé et non supervisé en utilisant des données étiquetées et non-étiquetées pour le même ensemble de données.

Étant donné que la machine n'est pas entièrement supervisée, on emploie le terme « semi-supervisé ». [26]

I.4.3 Apprentissage non supervisé

Pour ce type d'apprentissage la base de données d'apprentissage ne contient pas de variable cible (comme on l'a vu en apprentissage supervisé). Il y a seulement un ensemble de données collectées en entrée. L'algorithme doit découvrir par lui-même la structure en fonction des données. On utilise cette technique pour partitionner les données en groupes d'éléments homogènes. La distance est souvent la plus utilisée comme mesure de similarité entre les groupes [28]. La figure (I.1), illustre un exemple d'apprentissge non supervisé.

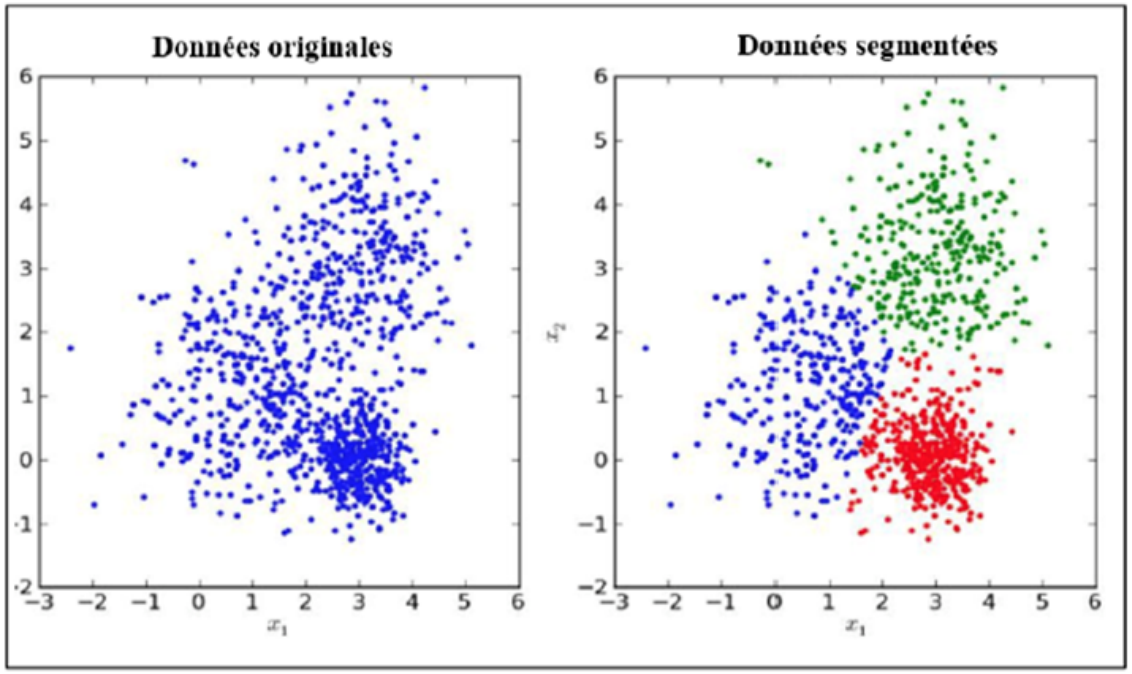


Figure I.1 – *Exemple d'apprentissage non supervisé.*

I.4.3.1 Problème résolu par apprentissage non supervisée

Nous distinguons deux types d'apprentissage non supervisé ; les transformations de l'ensemble de données et le clustering où la transformatios non supervisées est la réduction de la dimensionnalité et l'extraction des données.

- **Le clustering :** consiste à partitionner l'ensemble de données en groupes, appelés sous ensembles (clusters). L'objectif est de diviser les données de telle manière que les points d'un même cluster soient très similaires et que les points des différents clusters soient différents. Similaire à la classification [9].
- **La classification non supervisée** est la recherche des classes principales de la distribution par le processus de l'apprentissage non superviser puisque les classes ne sont pas connue à l'avance (données cachées)[14] .

I.4.3.2 Les algorithmes d'apprentissage non supervisé

Les algorithmes souvent utilisé dans l'apprentissage non supervisé sont les suivants [10] :

- **Algorithme de partitionnement en k-moyennes apprentissage non supervisé groupement (K-means clustering) :** L'algorithme de partitionnement en k-moyennes est un type d'apprentissage non supervisé servant à catégoriser les données non étiquetées, c'est-à-dire les données sans catégories ni groupes définis.
- **Réduction de la dimensionnalité (Dimensionality Reduction) :** Il s'agit de réduire le nombre de variables examinées. Dans de nombreuses applications, les données brutes possèdent de nombreuses caractéristiques dimensionnelles, dont certaines sont super-

flues ou non pertinentes. Réduire les dimensions permet donc de trouver la véritable relation latente.

- **Réseaux de neurones (Neural networks) :** Le réseau de neurones artificiel (RNA) contient un ensemble de neurones artificiels fortement interconnectés inspirées des neurones biologiques du cerveau. Le but est d'imiter certaines fonctions du cerveau humain, tel que la mémorisation par association, l'apprentissage par exemple, etc.
- **Analyse des composants principaux (Principal Component Analysis) :** L'analyse en composantes principales (ACP) est une technique descriptive permettant d'étudier les relations qui existent entre les variables, sans tenir compte, a priori, d'une quelconque structure. Le but de l'ACP est d'identifer la structure de dépendance entre des observations multivariables afin d'obtenir une description ou une représentation compacte de ces dernières.

I.4.4 Apprentissage par renforcement

L'apprentissage par renforcement permet d'analyser et d'optimiser le comportement d'un agent en fonction du retour d'informations de l'environnement (Ceci est appelé le signal de renforcement). Les machines essaient différentes situations pour déterminer les actions les plus avantageuses, plutôt que de simplement recevoir des instructions sur les actions à entreprendre.

Ce qui distingue l'apprentissage par renforcement des autres techniques, ce sont l'apprentissage par essais et erreurs et la récompense différée. Il peut être très avantageux pour la prévision financière à haute fréquence où l'environnement est dynamique et en conséquence, il est difficile de trouver ou d'automatiser manuellement des stratégies efficaces.

Pour ce présent chapitre nous avons se limiter aux algorithmes de classification (apprentissage supervisé) que nous allons détailler dans les sections suivantes.

I.5 Applications de l'apprentissage automatique

L'apprentissage automatique peut être utilisé dans des différents domaines, tel que :

- **Santé :** L'exploration de données offre un potentiel considérable pour améliorer les systèmes de santé. Il utilise des données et des analyses pour identifier les meilleures pratiques permettant d'améliorer les soins et de réduire les coûts.
- **Education :** Voici un nouveau domaine émergent, appelé Educationnel Data Mining, qui concerne le développement de méthodes permettant de découvrir des connaissances à partir de données provenant d'environnements éducatifs.
- **Détection de fraude :** Des milliards de dollars ont été perdus à cause des fraudes. Les méthodes traditionnelles de détection des fraudes prennent du temps et sont complexes. L'exploration de données aide à fournir des modèles significatifs et à transformer les données en informations.

- **Analyse du panier de marché :** L'analyse du panier de marché est une technique de modélisation basée sur une théorie selon laquelle, si vous achetez un certain groupe d'articles, vous êtes plus susceptible d'acheter un autre groupe d'articles. Cette technique peut permettre au détaillant de comprendre le comportement d'achat d'un acheteur.
- **Segmentation de la clientèle :** Les études de marché traditionnelles peuvent nous aider à segmenter les clients, mais l'exploration de données est approfondie et accroît l'efficacité du marché. L'exploration de données aide à aligner les clients dans un segment distinct et peut adapter les besoins en fonction des clients.
- **Banque financière :** Avec les opérations bancaires informatisées partout, une quantité énorme de données doit être générée avec de nouvelles transactions. Data mining peut contribuer à résoudre les problèmes commerciaux des secteurs bancaire et financier en détectant des schémas, des liens de causalité et des corrélations dans les informations commerciales et les prix du marché qui n'apparaissent pas immédiatement aux gestionnaires car les données de volume sont trop volumineuses ou générées trop rapidement pour être analysées par des experts.
- **Ressources Humaines :** Le Datamining est également utilisé dans les ressources humaines (RH) de certains ministères pour identifier les caractéristiques de leurs employés les plus performants. L'information obtenue peut contribuer aux efforts de recrutement des ressources humaines.
- **Applications industrielles :** A partir d'une base de données industrielle, le datamining va permettre de classifier, d'estimer, de segmenter et de décrire ces données. Cela va servir en marketing (étude de marché, service après-vente), en conception (conception du produit, analyse des pratiques de conception) et en fabrication (définition de gammes de production, amélioration des processus de fabrication, gestion ou amélioration de la qualité [5].

I.6 Généralités et types de classification

L'apprentissage d'une règle de classification est l'un des thèmes de l'apprentissage artificiel le plus traité. Il y a plusieurs raisons à cela : d'abord, on sait l'aborder du point de vue des théories de l'apprentissage, la plupart du temps dans le cas de deux classes (mais on peut assez facilement généraliser à un nombre quelconque). Ensuite, un grand nombre de méthodes et d'algorithmes existent, en particulier dans le cas où l'espace de représentation est numérique. On est alors dans le domaine classique de la reconnaissance statistique des formes (statistical pattern recognition). Enfin, apprendre à classer est un problème central de l'intelligence, naturelle comme artificielle.

Intuitivement, une règle de classification est un acte cognitif ou une procédure permettant d'affecter à un objet la famille à laquelle il appartient, autrement dit de le reconnaître. C'est ainsi qu'un enfant apprend à classer les animaux domestiques en « chiens » ou « chats », les plats en « salé » ou « sucré », etc. Par analogie, les ordinateurs de bureau qui reconnaissent

l'écriture manuscrite ont appris (grâce à un programme d'apprentissage automatique) des règles pour distinguer les signes tracés ; d'autres programmes savent classer des sons, des signaux biomédicaux, etc. Toutes les procédures qui simulent des fonctions perceptives doivent évidemment posséder des capacités de généralisation, c'est à dire être munies de la faculté d'induction, sans quoi elles ne seraient capables de reconnaître que les exemples qui ont servi à les entraîner.

I.6.1 Types de classification

On distingue essentiellement deux types de classification : supervisée et non-supervisée.

I.6.1.1 Classification supervisée

Dans le contexte supervisé on dispose déjà d'exemples dont la classe est connue et étiquetée. Les données sont donc associées à des labels des classes notés

$$\Theta = \{q_1, q_2, ..., q_n\} \tag{I.1}$$

L'objectif est alors d'apprendre à l'aide d'un modèle d'apprentissage des règles qui permettent de prédire la classe des nouvelles observations ce qui revient à déterminer une fonction C_l qui à partir des descripteurs (D) de l'objet associe une classe q_i et de pouvoir aussi affecter toute nouvelle observation à une classe parmi les classes disponibles. Ceci revient à la fin à trouver une fonction qu'on note Y_s qui associe chaque élément de X un élément de Q. On construit alors un modèle en vue de classer les nouvelles données.

Parmi les méthodes supervisées on cite : les k-plus proches voisins, les arbres de décision, les réseaux de neurones, les machines à support de vecteurs (SVM) et les classificateurs de Bayes.

I.6.1.2 Classification non-supervisée

Cette classification est aussi appelée "classification automatique", "clustering" ou encore "regroupement". La classification non-supervisée est utilisée lorsque que l'on possède des documents qui ne sont pas classés et dont on ne connaît pas de classification. A la fin du processus de classification non-supervisée, les documents doivent appartenir à l'une des classes générées par la classification [27].

L'absence d'étiquette de classe est un lourd handicap qui n'est que très partiellement surmontable. Seule l'analyse de la répartition spatiale des observations peut permettre de "deviner" où sont les véritables classes. Les deux difficultés essentielles que rencontre la classification non supervisée sont les suivantes :

- S'il est naturel de reconnaître comme "appartenant à une même classe" des observations regroupées dans une même zone de forte densité, il n'en est pas de même dans des zones de faible densité. En particulier, on peut s'attendre à ce que la définition de frontières entre les classes soit sujette à caution, et pour le moins hasardeuse.

- L'oeil humain est un extraordinaire outil de classification non supervisée. Malheureusement, il n'est opérationnel que pour des données bidimensionnelles, alors que les données que rencontre l'analyste sont couramment décrites par des dizaines de variables ou plus. Il s'avère que reproduire les performances de l'oeil humain dans des espaces de grande dimension est un exploit aujourd'hui hors d'atteinte des machines. [24]

La classification non-supervisée est divisée en deux suivants types : ;

- **Classification hiérarchique :** Où les sous-ensembles créés sont emboîtés de manière hiérarchique les uns dans les autres. On distingue la CH descendante qui part de l'ensemble de tous les individus et les fractionnes en un certain nombre de sous-ensembles, chaque sous-ensemble étant alors fractionné en un certain nombre de sous-ensembles, et ainsi de suite. Et la CH ascendante qui part des individus seuls que l'on regroupe en sous-ensembles, qui sont à leur tour regroupés, et ainsi de suite. Pour déterminer quelles classes on va fusionner, on utilise le critère d'agrégation.

  La classification hiérarchique peut être schématisée par l'organigramme montré dans la figure (I.2) :

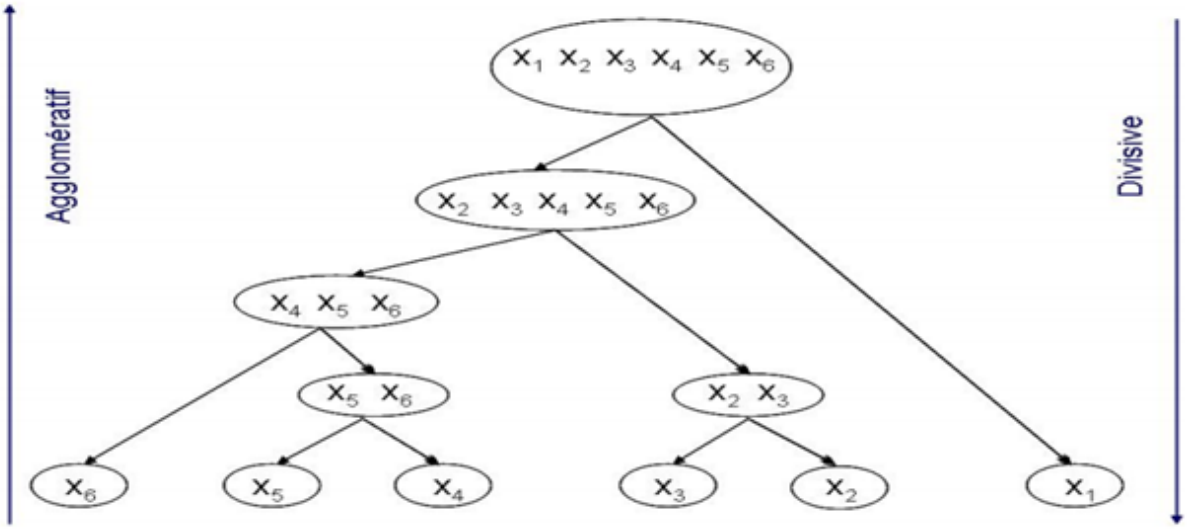


Figure I.2 – *Classification ascendante et descendante.*

- **Classification non-hiérarchique :** Où les individus ne sont pas structurés de manière hiérarchique. Si chaque individu ne fait partie que d'un sous-ensemble, on parle de partition. Si chaque individu peut appartenir à plusieurs groupes, avec la probabilité P_i d'appartenir au groupe i, alors on parle de recouvrement.

I.7 Conclusion

Dans ce chapitre nous avons introduit et expliqué le fonctionnement des algorithmes d'apprentissage automatique les plus fréquent qui peuvent être utilisé dans la classification. Dans

le chapitre 2 et 3 nous allons montrer comment utilisé une démarche basée sur ces derniers et comment choisir l'algorithme qui convient le plus à notre cas d'étude. L'entrainement de ces algorithmes avec des données historiques et le choix des paramètres sera traité.

La importante tache consiste à choisir le modèle qui génère des résultats cohérents avec les résultats réels, c-a-d une comparaison entre les résultats obtenus et les classes définie par les experts.

Le prochains chapitre présentera les principales techniques et métriques d'évaluation des modèles prédictifs dans le contexte de la classification.

Chapitre II

Méthodes de classification

II.1 Introduction

La classification est une méthode mathématique d'analyse des données. Elle facilite l'étude d'une population, en regroupant les individus en plusieurs classes de telle sorte qu'ils soit le plus possible semblable. L'un des classificateurs les plus simples et plutôt triviaux est le classificateur Rote, qui mémorise l'ensemble des données d'apprentissage et effectue la classification uniquement si les attributs de l'objet de test correspondent exactement aux attributs de l'un des objets d'apprentissage.

Il existe plusieurs méthodes de classification. Le choix du bon algorithme de dépend de plusieurs facteurs, notamment la taille, la qualité et la diversité des données, ainsi que les réponses attendues. Il y a bien d'autres aspects à prendre en compte, comme la précision, le temps d'entraînement, les paramètres et les points de données.

Choisir le bon algorithme dépend donc à la fois des besoins, des spécifications, des essais ainsi que du temps disponible. Il est difficile de déterminer l'algorithme le plus efficace d'effecteur des essais.

Le reste de ce chapitre traite de plusieurs méthodes de classification. Le choix c'est posé sur K plus proches voisins (K-PPV), Machine vecteur de support (SVM) et Forêt aléatoire de base. Ainsi, les algorithmes, leurs domaines d'application et leurs avantages seront présentés.

II.2 L'algorithme K plus proches voisins K-PPV

L'algorithme des K plus proches voisins s'écrit en abrégé k-NN ou KNN, de l'anglais k-Nearest Neighbors, appartient à la famille des algorithmes d'apprentissage automatique (machine learning). Le terme de machine learning a été utilisé pour la première fois par l'informaticien américain Arthur Samuel en 1959. Les algorithmes d'apprentissage automatique ont connu

un fort regain d'intérêt au début des années 2000 notamment grâce à la quantité de données disponibles sur internet.

L'algorithme des K plus proches voisins est un algorithme d'apprentissage supervisé, il est nécessaire d'avoir des données labellisées. A partir d'un ensemble E de données labellisées, il sera possible de classer (déterminer le label) d'une nouvelle donnée (donnée n'appartenant pas à E). A noter qu'il est aussi possible d'utiliser l'algorithme des K-PPV à des fins de régression en statistiques (on cherche à déterminer une valeur à la place d'une classe). [25].

La méthode K-PPV a été utilisée par plusieurs auteurs dans différents domaines, parmi ; Eve Mathieu-Dupas qui a présenté le fondement théorique de la méthode K-PPV et a illustré cette technique d'apprentissage statistique au travers du problème diagnostique d'une pathologie complexe [17].

B.Taconet et al ont utilisé la classification des K-PPV par sous-voisinages emboîtés sur des iris de Fisher et sur la base de données mnist [15].

A.M.Qamar et E.Gaussier ont développé deux algorithmes "eSiLA" et "SiLA" dans le but d'améliorer l'algorithme K-PPV ([1],[13]).

II.2.1 Principe de l'algorithme

Une approche plus sophistiquée, la classification K-PPV, trouve un groupe de k objets dans l'ensemble d'apprentissage qui sont les plus proches de l'objet de test, et fond l'attribution d'une étiquette sur la prédominance d'une classe particulière dans ce quartier.

L'algorithme du voisin le plus proche est une technique non paramétrique. Mais avant de détailler l'algorithme, quelques définitions sont nécessaires :

- **La « distance »** entre deux objets est prise comme la distance euclidienne entre eux. Pour calculer cette distance, chaque objet doit être représenté par un vecteur de position dans un espace de caractéristiques multidimensionnel. Soit les vecteurs x_i et x_j être deux échantillons d'entrée (objets) avec p caractéristiques $(x_1, x_2, ..., x_p)$. Un objet est le **« voisin »** d'un autre objet si la distance entre eux est en dessous d'un seuil prédéfini.
- **Le « plus proche voisin »** d'un objet x est l'objet échantillon dont la distance à x est la plus faible parmi tous les échantillons d'entrée.
- **Le « 2e voisin le plus proche »** d'un objet x est l'objet échantillon dont la distance à x est la deuxième plus basse parmi tous les échantillons d'entrée. Le « nième voisin le plus proche » est défini de manière analogue.
- **Les « k voisins les plus proches »** d'un objet x sont la collection d'objets échantillons x_i où $i = \{1, 2, ..., k\}$ et x_i est le ième voisin le plus proche de x.

Les étapes de l'algorithme du voisin le plus proche peuvent être décrites comme suit :

II.2.1.1 Phase de formation

- Un être humain classe un certain nombre d'objets manuellement. C'est l'ensemble d'entraînement. Les vecteurs de caractéristiques et les étiquettes de classe de ces échantillons sont stockés.
- L'ordinateur lit cet ensemble d'objets. La classification correcte de ces objets est connue.

II.2.1.2 Phase de classement

Un nouvel objet d'entrée non classifié (échantillon de test) est classé par un vote majoritaire de ses voisins.

Normalement, la phase d'apprentissage est exécutée une fois, et la phase de classification est exécutée un certain nombre de fois par la suite.

II.2.2 Désavantages du K-PPV

Les inconvénient de la classification par K-PPV sont les suivants :

- S'il existe une classe avec un très grand nombre d'échantillons d'apprentissage par rapport aux autres classes, alors ses échantillons reviennent plus fréquemment parmi les k plus proches voisins d'un nouvel objet lorsque ceux-ci sont calculés. Cette classe domine la classification des nouveaux objets, en écrasant les échantillons appartenant à d'autres classes. Cela peut être évité en augmentant légèrement le vote majoritaire. Par exemple, on peut le modifier pour que la distance de chaque voisin à l'échantillon test détermine la « force » ou la « proximité » de ce voisin. Ainsi, plus la distance est courte, plus l'effet de l'échantillon sur le vote majoritaire est important.
- La précision chute considérablement lorsqu'il y a des caractéristiques bruyantes ou non pertinentes, ou si les échelles des caractéristiques sont incompatibles avec leur importance.
- L'algorithme ne rapporte pas les probabilités de confiance ou de classe.
- Un classement est toujours effectué. Il n'y a pas d'objets qui ne peuvent être affecté à une classe.

II.2.3 Le choix de la valeur de K

La taille de la valeur K est importante. Les petites et grandes valeurs de K ont des caractéristiques différentes. Les petites et grandes valeurs de k sont comparées dans le tableau II.1. Les techniques heuristiques telles que la validation croisée peuvent aider à sélectionner une bonne valeur pour k. En fin de compte, bien sûr, la meilleure valeur de k dépend des données disponibles. S'il n'y a que deux classes différentes, un nombre pair de K peut provoquer une égalité. Le choix d'une valeur impaire pour K évite ce problème.

Tableau II.1 – *Caractéristiques des petites et grandes valeurs de K.*

Petites valeurs de k	**Grandes valeurs de k**
Provoquer un sur-ajustement	Provoquer une généralisation excessive
Augmenter l'effet négatif du bruit	Réduire l'effet négatif du bruit
Créer des limites de classe distinctes	Créer des limites de classe indistinctes

De plus, la figure (II.1) montre l'influence du choix de la valeur de K sur l'apprentissage des données.

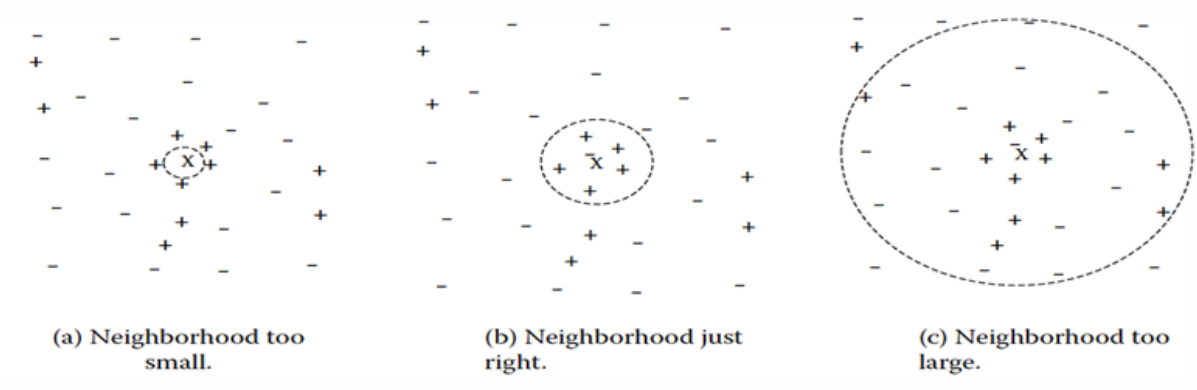


Figure II.1 – *Classification KNN avec petites moyennes et grandes valeurs de K.*

II.2.4 Calcul de la distance

La distance commune pour des variables continues est la distance euclidienne. Pour des variables discrètes, comme en classification de texte, une autre distance peut être utilisée, telle que la distance de recouvrement(ou la distance de Hamming). Dans le contexte de micro-tableau de données génétiques, par exemple, K-PPV a aussi été employée avec des coefficients de corrélation de Pearson et Spearman. Fréquemment, la précision de la classification peut être améliorée de manière significative si la distance est apprise par des algorithmes spécialisés tels que la méthodes du plus proche voisin à grande tolérance ou l'analyse des composantes de voisinage. Afin de trouver les K plus proches d'une donnée continue à classer, on peut choisir la distance euclidienne. Soient deux données représentées par deux vecteurs x_i et x_j , la distance entre ces deux données est donnée par :

$$d(X_i, X_j) = \sqrt{\Sigma(X_{ik} - X_{ij})^2} \tag{II.1}$$

Après il faut implémenter (en Matlab, R, Pyhton ou Octave) l'algorithme des K-plus proches voisins pour prédire les classes de nouvelles données à partir de données étiquetées (données d'apprentissage). Dans notre cas, nous allons implémenter nos données en Python, dont nous allons argumenter notre choix dans ce qui suit dans ce présent chapitre.

L'algorithme des K plus proches voisins peut être présenté :

Algorithm 1 Algorithme des K plus proches voisins (K-NN)

Require:

Données d'apprentissage : $\mathbf{X}_{\text{train}} = (x_1^{\text{train}}, \ldots, x_n^{\text{train}})$
Classes des données d'apprentissage : $\mathbf{z}_{\text{train}} = (z_1^{\text{train}}, \ldots, z_n^{\text{train}})$
Données de test : $\mathbf{X}_{\text{test}} = (x_1^{\text{test}}, \ldots, x_m^{\text{test}})$
Nombre de plus proches voisins : K

Ensure:

Classes des données de test : $\mathbf{z}_{\text{test}} = (z_1^{\text{test}}, \ldots, z_m^{\text{test}})$

for $i \leftarrow 1$ **à** m **do** ▷ Pour chaque exemple de test
  for $j \leftarrow 1$ **à** n **do** ▷ Pour chaque exemple d'entraînement
    Calculer la distance euclidienne d_{ij} entre x_i^{test} et x_j^{train} :
    $$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik}^{\text{test}} - x_{jk}^{\text{train}})^2}$$
  end for
  ▷ Trouver les K plus proches voisins
  Trier les distances d_{ij} par ordre croissant pour $j = 1, \ldots, n$
  Enregistrer les indices IndVoisins correspondant aux K plus petites distances
  ▷ Déterminer la classe majoritaire
  Initialiser un compteur pour chaque classe : $C_h \leftarrow 0$, $h = 1, \ldots, K$
  for $k \leftarrow 1$ **à** K **do**
    $\text{ind\_voisin}_k \leftarrow \text{IndVoisins}_k$
    $h \leftarrow z_{\text{ind\_voisin}_k}^{\text{train}}$ ▷ Classe du voisin
    $C_h \leftarrow C_h + 1$ ▷ Incrémenter le compteur de classe
  end for
  ▷ Attribuer la classe majoritaire
  $z_i^{\text{test}} \leftarrow \arg\max_h C_h$
end for
return $\mathbf{z}_{\text{test}}$

II.3 Forêts aléatoires

Afin de comprendre le fonctionnement de cette méthode,il est important de comprendre le fonctionnement d'un arbre de décision (ou Decision tree en anglais).

II.3.1 Arbre de décision

Avec une racine, des noeuds, des branches et des feuilles, l'arbre de décision est un classifieur en forme d'arbre (en sens informatique). Lorsqu'on entraîne un modèle de type arbre de décision, les features du dataset sont testés au niveau des noeuds de l'arbre, puis pour chaque valeur possible du feature testé, une branche est formée. La procédure se poursuit jusqu'à ce qu'il n'y

ait plus de feature à tester et on obtient ainsi des feuilles représentant les différentes classes de la variable cible (on possède généralement une classe au minimum).

Le schéma dans la figure (II.2) décrit le principe de l'arbre de décision.

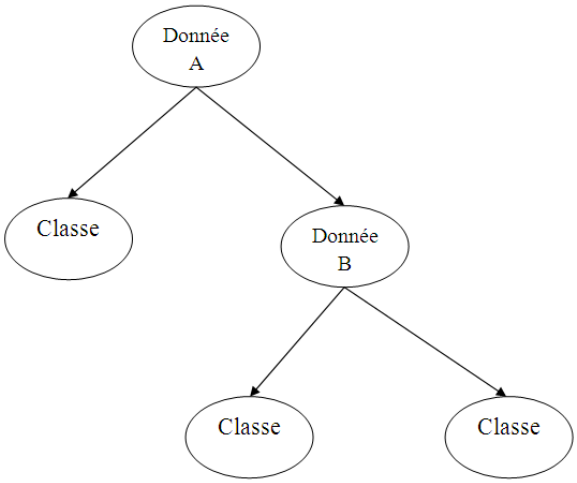


Figure II.2 – *Schéma d'arbre de décision.*

II.3.2 Principe de fonctionnement de Forêts aléatoires

Dans le but de traitement des problèmes de classification, Leo Breiman a proposé l'algorithme des forêts aléatoires (ou Random Forest en anglais) [6]. Comme son nom l'indique, il s'agit d'un algorithme qui combine plusieurs arbres de décisions dans une approche de type bagging (Moyenner les prédictions de plusieurs modèles indépendants afin de réduire l'erreur de prédiction générale).

L'algorithme entraîne plusieurs arbres de décisions sur des samples sélectionnés aléatoirement. La prédiction de l'algorithme est calculée à partir de la moyenne des prédictions de chaque arbre de décision construit avec des données sont quantitatives. Pour les données qualitatives, l'algorithme utilise la moyenne des prédictions des modèles indépendants et procède à un vote pour déterminer sa prédiction, comme l'illustre la figure (II.3).

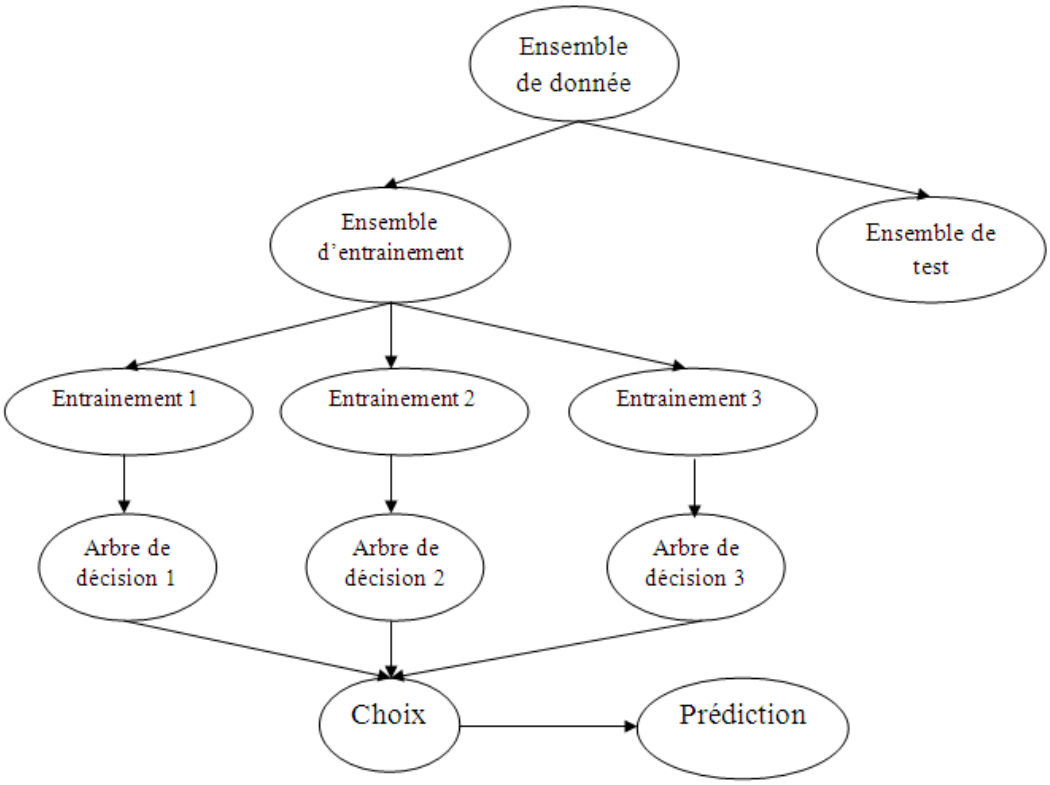


Figure II.3 – *Fonctionnement des Forêts aléatoires(Random Forest).*

Dans le but de confirmer l'efficacité de cet algorithme W.Dazhong et al ont réalisé une étude comparative sur la prédiction de l'usure de l'outil, en utilisant essentiellement FA et d'autres algorithmes tel que SVR et RNA [13].

Algorithm 2 Algorithme des Forêts Aléatoires

Require:

$\mathbf{x}$: observation à prévoir

$\mathcal{D}_n$: échantillon d'apprentissage

B : nombre d'arbres

m : nombre de variables candidates par nœud ($m \leq p$)

Ensure: $\hat{h}(\mathbf{x})$: estimation de la prédiction

for $k = 1$ **à** B **do**

    Tirer un échantillon bootstrap $\mathcal{D}_n^{(k)}$ depuis $\mathcal{D}_n$

    Construire un arbre CART $h(\cdot, \theta_k)$ sur $\mathcal{D}_n^{(k)}$ avec :

        - À chaque nœud : choisir m variables aléatoirement parmi p

        - Sélectionner la coupure minimisant le critère CART

end for

return $\hat{h}(\mathbf{x}) = \dfrac{1}{B} \displaystyle\sum_{k=1}^{B} h(\mathbf{x}, \theta_k)$

II.4 Machine vecteur de support (SVM)

L'algorithme des machines à vecteurs de support est l'un des méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnika en 1995. Initialement, les SVM ont été développé comme un algorithme de classification binaire supervisée.

Le principe des SVM consiste à projeter les données de l'espace d'entrée (appartenant à deux classes différentes) non linéairement séparables dans un espace de plus grande dimension appelé espace de caractéristiques de façon à ce que les données deviennent linéairement séparables. Dans cet espace, on construit un hyperplan optimal séparant les classes tel que :

- Les vecteurs appartenant aux différentes classes se trouvent de différents côtés de l'hyperplan.
- La plus petite distance entre les vecteurs et l'hyperplan (la marge) soit maximal[18].

Cette méthode est donc dans son origine binaire. Cependant, les problèmes du monde réel sont dans la plupart des cas multi classe, l'exemple le plus simple en est la reconnaissance des caractères optiques (OCR). Dans de tels cas, on ne cherche pas à affecter un nouvel exemple à l'une de deux classes mais à l'une parmi plusieurs, c'est à dire que la décision n'est plus binaire et un seul hyperplan ne suffit plus.

Le problème étudie dans ce travail est a multi classe, où les méthodes des machines à vecteur support multi classe seront obligatoirement utilisées, en composant de plusieurs hyperplans bi-classes permettant de tracer les frontières de décision entre les différentes classes. Ces méthodes décomposent l'ensemble d'exemples en plusieurs sous ensembles représentant chacun un problème de classification binaire. Pour chaque problème un hyperplan de séparation est déterminé par la méthode SVM binaire. On construit lors de la classification une hiérarchie des hyperplans binaires qui est parcourue de la racine jusqu'à une feuille pour décider de la classe d'un nouvel exemple[12].

Le principe de l'lgorithme SVM peut être illustré par la figure (II.4) :

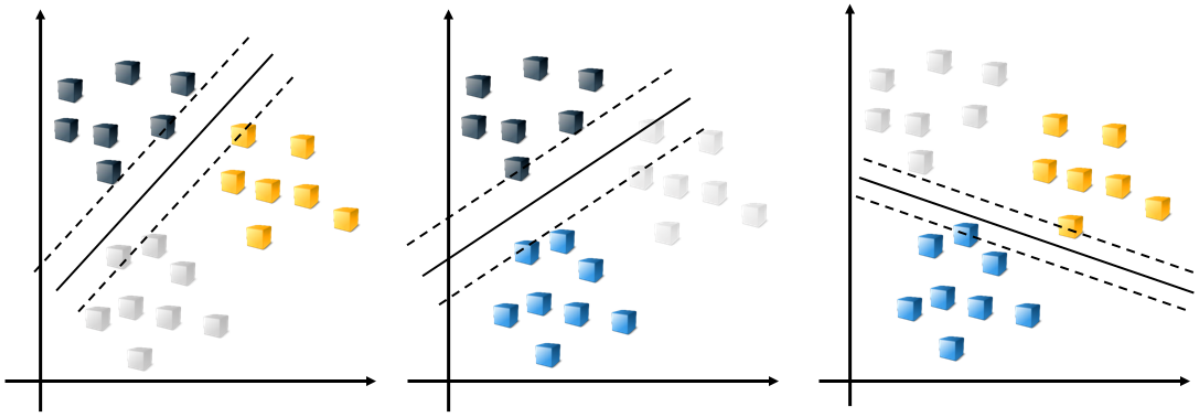


Figure II.4 – *Principe de l'algorithme SVM[2].*

Pour créer un modèle d'apprentissage automatique, il est indispensable de parler de d'une phase importante, qui est bien la phase du « prétraitement des données ».Il est souvent judicieux

d'inspecter les données, pour voir si la tâche est facilement solvable sans apprentissage automatique, ou si les informations souhaitées pourrait ne pas être contenu dans les données. De plus, l'inspection de nos données est un bon moyen de trouver des anomalies et des particularités.

Dans le monde réel, les données sont généralement incomplètes : absence de valeurs d'attribut, absence de certains attributs intéressants ou contenant uniquement des données globales. Bruyant : contenant des erreurs ou des valeurs aberrantes. Incohérent : contient des écarts dans les codes ou les noms.

II.5 Le prétraitement des données

Le prétraitement des données est une technique d'exploration de données qui consiste à transformer des données brutes dans un format compréhensible.

Les données réelles sont souvent incomplètes, incohérentes et / ou absentes de certains comportements ou tendances, et sont susceptibles de contenir de nombreuses erreurs. Le prétraitement des données est une méthode éprouvée pour résoudre ces problèmes.

II.5.1 Étapes du prétraitement des données

La phase du prétraitement des données peut être divisée en 5 étapes suivantes :

1. Importez les bibliothèques : le Python utilise des mot-clé pour l'importation des bibliothèques (importer les bibliothèques les plus populaires utilisées par tout Data Scientist).
2. Importez l'ensemble de données :L'ensemble de données peut etre importé en utilisant l'une de bibliothèques.
3. Pour gérer les données manquantes, il existe plusieurs stratégies appropriées selon le cas étudié. En effet, on peut : ignorer les valeurs manquantes, remplir manuellement, adopter la moyenne générale, ou encore la moyenne par classe, ou bien, élaborer un algorithme d'apprentissage permettant de remplir les valeurs manquantes avec la valeur la plus probable.
4. **C**onsultez les valeurs catégorielles : les variables catégorielles causent problème dans l'ensemble des données, pour éviter cela ils doivent être converties en valeurs numériques.
5. **D**ivision de l'ensemble de données en ensemble d'entraînement et de test : Ensemble d'entraînement et ensemble de test qui vont apprendre des données pour faire prédictions.

II.6 Choix du langage

Python est devenu la lingua franca pour de nombreuses applications de science des données. Il combine la puissance des langages de programmation à usage général avec la facilité d'utilisation des langages de script spécifiques à un domaine comme MATLAB ou R. Python dispose de

bibliothèques pour le chargement de données, la visualisation, les statistiques, le traitement du langage naturel, le traitement d'images, etc. Cette vaste boîte à outils fournit aux scientifiques des données un large éventail de fonctionnalités générales et spéciales.

L'un des principaux avantages de l'utilisation de Python est la possibilité d'interagir directement avec le code, à l'aide d'un terminal ou d'autres outils comme le Jupyter Notebook, que nous verrons sous peu. L'apprentissage automatique et l'analyse des données sont des processus fondamentalement itératifs, dans lesquels les données guident l'analyse. Il est essentiel que ces processus disposent d'outils permettant une itération rapide et une interaction aisée.

En tant que langage de programmation à usage général, Python permet également la création d'interfaces utilisateur graphiques (GUI) et de services Web complexes, ainsi que l'intégration dans des systèmes existants.

II.7 Critères de choix entre les algorithmes

Le tableau suivant donne un résumé non technique sur les algorithmes utilisés pour l'apprentissage supervisé dans notre cas d'étude.

Le choix du bon algorithme dépend de deux critères :

- Le taux d'apprentissage : cela doit être supérieur à 70% pour avoir un modèle valide.
- Le taux de validation : ce dernier doit être inférieur au taux d'apprentissage

Pour comparer entre les algorithme, nous avons proposé un algorithme comparatif qui compare de base entre les critères cités précédemment.

II.8 Interface graphique

Les interfaces graphiques (interface homme-machine) sont des outils permettent à l'utilisateur d'interagir avec une fonctionnalité informatique dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran, de sorte que l'usager peut utiliser en imitant la manipulation physique de ces objets avec un dispositif de pointage, le plus souvent une souris.

II.8.1 Interface graphique Tkinter

Tkinter est une bibliothèque Python pour réaliser les interface graphique en utilisant TK une standard Framework graphique. C'est le plus utiliser en Python et disponible en Licence gratuit.

II.9 Conclusion

Ce chapitre apporte des explication et des datais sur le fonctionnement des algorithmes d'apprentissage automatique utilisés pour faire la classification, pour pouvoir choisir le meilleur

pour cette étude. D'autre part nous, une présentation des étapes à poursuivre pour faire le prétraitement de nos données, poursuit par l'argumentation du choix du langage utilisé au cours de cette étude sont présenté.

Par contre, dans le prochain chapitre les étapes de mise en place de notre outil seront misent en évidence. En commençant l'application des algorithmes et l'interprétation des résultats obtenus par et en terminant par le choix du bon algorithme et la proposition d'une interface graphique.

Chapitre III

Mise en oeuvre des méthodes de classification (application de la démarche)

III.1 Introduction

Ce chapitre aborde la méthodologie poursuite qui peut être utilisés pour tester et évaluer la performance des modèles d'apprentissage automatique, pour les problèmes de classification. Nous commençons par une brève description de l'entreprise où nous avons collecté les données utilisées. Ensuite, nous présentons le outils et technologies de programmation utilisés pour trainer nos model, avant de résumer les étapes poursuites par un synoptique descriptif. Enfin, nous conclurons le chapitre par l'interprétation des résultats obtenus.

III.2 Présentation de l'entreprise

Dans le cadre de préparation de notre mémoire de fin d'étude, nous avons choisi d'effectuer notre stage au niveau de la cimenterie d'AIN TOUTA.

La cimenterie d'AIN TOUTA est une usine à voie sèche. Les matières premières sont constituées de calcaire et d'argile extraits d'une carrière située à proximité de l'usine. Elle a été mise en service le 07 Septembre 1986. Sa capacité de production annuelle est d'un million de tonnes. Le coût de l'investissement initial s'élève 1149000 000 DA clé en main.

III.3 Outils et technologies de programmation

Afin d'atteindre notre objectif de classification et de trainer nos modelés, plusieurs outils et technologies ont été utilisés.

III.3.1 Outils

Les principaux logiciels utilisés sont :

- **Microsoft Office 2008 :** Excel pour une meilleure visualisation des données.

III.3.2 Technologies

Plusieurs choix techniques ont été faits. Dans ce qui suit, nous illustrons les plus importants :

- **Python 3.9.2 :** Le choix du langage de programmation est Python. Il s'agit d'un langage orienté objet, facile à prendre en main, l'un des plus utilisé pour faire du machine learning. Python possède un grand nombre de librairies : d'analyse de données, de calcul scientifique, de visualisation et d'apprentissage automatique.
- **Scikit-Learn :** Scikit-learn est l'un des projets les plus prépondérant sur Github derrière (TensorFlow). Il s'agit d ?une librairie d'apprentissage automatique qui permet de faire de la classification, régression, du clustering, de la réduction de dimension, de la sélection de modèle et du prétraitement de données.
- **Pandas :** Pandas est une librairie Python qui nous a permis de manipuler facilement nos données pour l'analyse et le prétraitement.
- **NumPy :** La bibliothèque NumPy permet d'effectuer des calculs numériques avec Python.
- **Matplotlib :** Matplotlib est une bibliothèque Python qui permet de tracer et de visualiser des données sous plusieurs formes de graphiques.
- **Seaborn :** Seaborn est une librairie qui vient compléter certaines fonctionnalités non accessibles avec Matplotlib.
- **Jupyter Notebook :** Nous avons utilisé les notebooks Jupyter, cahiers électroniques qui, dans le même document, peuvent rassembler du texte, des images, des formules mathématiques et du code Python exécutable.
- **PyCharm :** Nous avons également utilisé PyCharm, un IDE pour la programmation Python, simple d'utilisation. Il nous a permis de déboguer des scripts afin de pouvoir nous situer dans une boucle 'for' par exemple.
- **Anaconda3 :** Nous avons utilisé Anaconda, une plateforme libre qui intègre un grand nombre de packages dont des librairies Python, pour l'installation des librairies citées ci-dessus.

III.4 Les étapes de la démarche :

Le synoptique décrit dans la figure (**??**) résume les étapes de la démarche de classification des données poursuite dans le cadre de notre projet

III.4.1 Les données utilisées

Les données utilisées pour entrainer et évaluer la performance du modèle utilisé dans ce projet sont des indicateurs de suivie utilisés par le service méthode. Ces paramètres sont présentés comme suit :

- **Disponibilité** Aptitude d'un bien à être en état d ?accomplir une fonction requise dans des conditions données, à un instant donné ou durant un intervalle de temps donné, en supposant que la fourniture des moyens extérieurs nécessaires est assurée. (Norme NF EN 13306).
  Taux de Disponibilité :
  $$D = \frac{MTBF}{MTBF \ + \ MTTR} \tag{III.1}$$
- **Performance :** L'organisme ISO défini la performance comme un résultat mesurable. Ce résultat mesurable peut porter sur des éléments quantitatifs ou qualitatifs. Tout en restant basé sur des mesures .

  Taux de Performance : Il mesure les écarts de performance du moyen dus aux variations de cadence et aux micros arrêts :
  $$P = \frac{Temps\ net}{Temps\ de\ fonctionnement} \tag{III.2}$$
- **La Défaillance :** Cessation de l'aptitude d'un bien à accomplir une fonction requise. (Norme NF EN 13306).
  Taux de Défaillance : est un indicateur de fiabilité représenté par la formule
  $$\lambda = \frac{1}{MTBF} \tag{III.3}$$
- **La réparation :** C'est l'ensemble des actions physiques exécutées pour rétablir la fonction requise d'un bien en panne. (Norme NF EN 13306).
  Taux de Réparation :
  $$\mu = \frac{1}{MTTR} \tag{III.4}$$

Dans notre cas, nous allons travailler sur les données d'une des zones de l'entreprise, qui est la zone 200 (broyeur crue), dont les données sont représentées par Excel.

III.4.2 Prétraitement des données

Suite au prétraitement réalisé en suivant les étapes citées précédemment avec l'utilisation de Jupyter notebook, nous avons pu afficher le tableau suivant :

- **Importation des bibliothèques :** En utilisant le mot clé import, les bibliothèques les plus populaires ont été importées.

- **Importation de l'ensemble de données :** En utilisant Pandas, nous avons pu importer le fichier Excel et l'afficher sur Jupyter comme montré dans la figure (III.1) :

```
Entrée [2]: dataset=pd.read_excel('C:\\Users\\Nacer\\Desktop\\Classeur12.xlsx')
            dataset
```

Out[2]:

	Mois	Taux_de_Disponibilité	Taux_de_Performance	Taux_de_Défaillance	Taux_de_Réparation	classes
0	51	2	78.00	2.84	6.29	C5
1	52	4	89.00	3.47	14.83	C5
2	53	6	85.00	3.02	11.83	C5
3	54	8	76.00	5.62	12.65	C5
4	55	10	89.00	4.52	23.05	C5
...	...	...	...	...	...	...
135	221	78	52.00	1.66	21.00	C1
136	222	65	54.00	1.80	22.00	C7
137	223	55	43.00	2.15	23.00	C7
138	224	57	73.83	2.26	24.00	C2
139	225	78	90.00	1.22	25.00	C1

140 rows × 6 columns

Figure III.1 – *Importation des données.*

- **Gérer les données manquantes :** cela a été fait manuellement.
- **Consultation des valeurs catégorielles :** Touts les paramètres utilisés sont déjà numériques, sauf celles des classes, et cela est affiché dans la figure (III.2)

```
Entrée [18]: numeric_features = dataset.dtypes[dataset.dtypes != "object"].index
             print(numeric_features)

Index(['Taux_de_Disponibilité', 'Taux_de_Performance', 'Taux_de_Défaillance',
       'Taux_de_Réparation'],
      dtype='object')

Entrée [19]: categorical_features = dataset.dtypes[dataset.dtypes == "object"].index
             print("categorical_features are :",categorical_features)

categorical_features are : Index(['classes'], dtype='object')
```

Figure III.2 – *Catégorisation des données.*

- **Division de l'ensemble de données en ensemble d'entraînement et de test :** cette étape était réalisé par le code dans la figure (III.3) :

```
Entrée [43]: all_features = dataset.drop("classes",axis=1)
             Targeted_feature = dataset["classes"]
             X_train,X_test,y_train,y_test = train_test_split(all_features,Targeted_feature,test_size=0.3,random_state=None)
             X_train.shape,X_test.shape,y_train.shape,y_test.shape

Out[43]: ((98, 4), (42, 4), (98,), (42,))
```

Figure III.3 – *Division des données.*

III.4.3 Apprentissage

Cette phase consiste à estimer un modèle à partir de données, elle est généralement réalisée préalablement à l'utilisation pratique des modèles. Avant de commencer cette phase nous étions dans l'obligation de choisir les classes de sortie selon nos paramètres, nous avons obtenue les

classes suivantes :

- Classe 1 : D= {70 ; 100} Déf= {0 ; 5}
- Classe 2 : D= {40 ; 70} P= {70 ; 100}
- Classe 3 : D= {70 ; 100} Déf = {5 ; 10}
- Classe 5 : D= {0 ; 40}
- Classe 6 : D= {40 ; 70} Déf= {10 ; 15}
- Classe 7 : D= {40 ; 70} Déf= {0 ; 10}

Après le choix des classes, nous avons préparé notre base de données (dataset), pour entrainer les algorithmes par la suite. La base de données obtenue a données les résultats dans la figure (III.4) :

```
Entrée [5]: # Target variable
            dataset.classes.value_counts()

Out[5]: C1    62
        C2    23
        C5    22
        C7    18
        C3    14
        C6     1
        Name: classes, dtype: int64
```

Figure III.4 – *L'affichage du nombre de classes.*

III.4.4 Classification des données (résultats et interprétations)

Afin de classifier nos données, nous avons appliqué les 3 algorithme comme suit :

III.4.4.1 L'algorithme des K plus proches voisins (KNN)

A partir de la bibliothèque SKLEARN, nous avons fait appel à l'algorithme KPPV comme le montre la figure (III.5) :

```
Entrée [9]: from sklearn.neighbors import KNeighborsClassifier
            model = KNeighborsClassifier(n_neighbors = 3)
            model.fit(X_train,y_train)
            prediction_knn=model.predict(X_test)
            print('-------------The Accuracy of the model----------------------------')
            print('The accuracy of the K Nearst Neighbors Classifier is',round(accuracy_score(prediction_knn,y_test)*100,2))
```

Figure III.5 – *Appel à l'algorithme du K-PPV.*

- **Comparaison du taux d'erreur avec la valeur K :** En calculant la moyenne d'erreur pour des valeurs prédites de K qui varient de 1 à 40, nous pourrions estimer la

meilleure valeur de K dans notre ensemble d'essai.
Cela le montre la figure (III.6) :

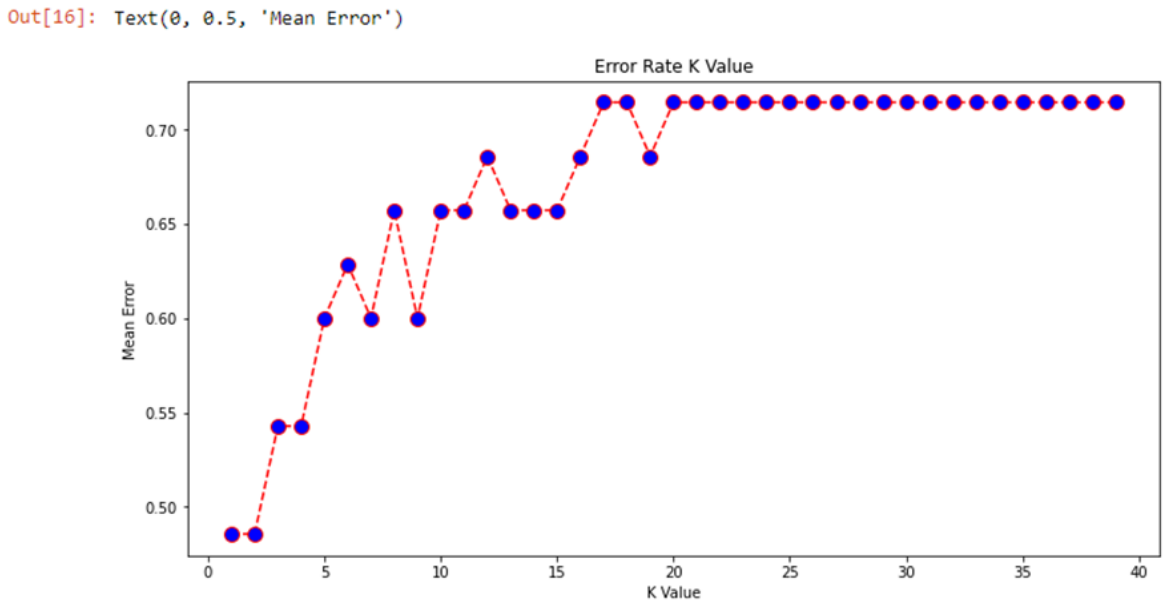


Figure III.6 – *La moyenne d'erreur en fonction de valeur du K.*

D'après la sortie, nous pouvons voir que l'erreur moyenne est presque nulle lorsque la valeur du K est comprise 1 et 2.

- **Résultats et interprétation :** La figure (III.7) représente la matrice de confusion :

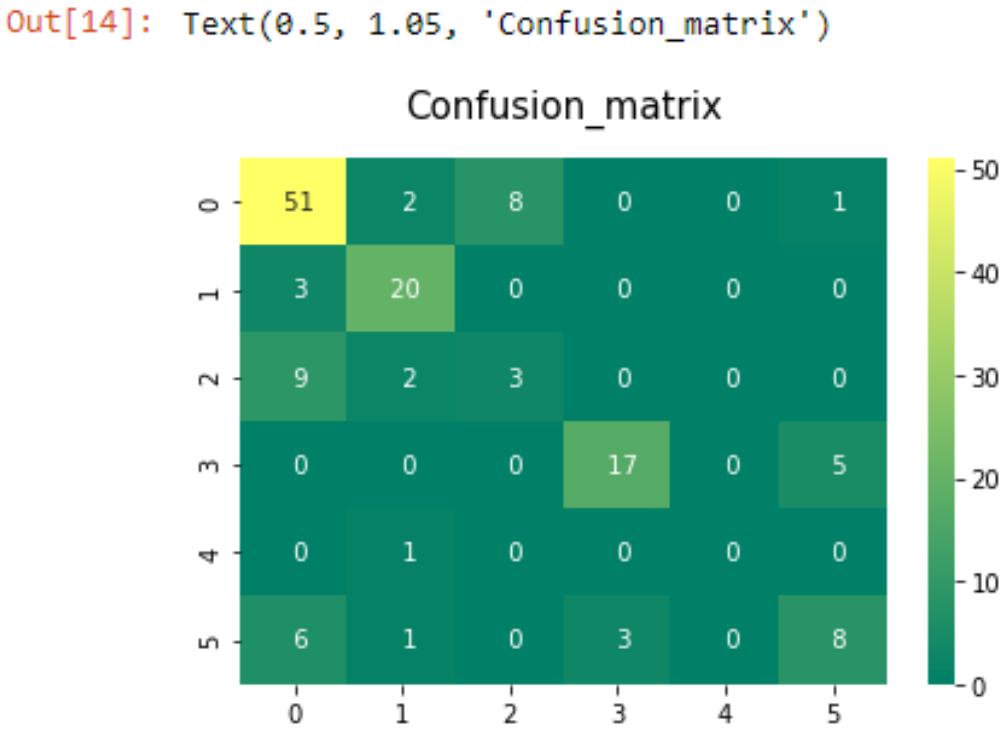


Figure III.7 – *Matrice de confusion du K-PPV.*

L'explication des valeurs de la matrice de confusion peut être faite de cette manière :

- Élément(0,0) : 51 des données de la classe 1 sont correctement classées par le K-PPV dans la classe 1.

- Élément(4,4) : 0 des données de la classe 6 sont classées correctement par le K-PPV dans la classe 6.

  Le reste de l'interprétation se trouve dans l'annexe (C.1.1.1).

- La figure (III.8) représente les taux de trainement et de validation obtenus :

```
Entrée [15]:
print('--------------The Accuracy of the model----------------------------')
print('The accuracy of the K Nearst Neighbors Classifier is',round(accuracy_score(prediction_knn,y_test)*100,2))
print('The cross validated score for K Nearest Neighbors Classifier is:',round(result_knn.mean()*100,2))

--------------The Accuracy of the model----------------------------
The accuracy of the K Nearst Neighbors Classifier is 78.57
The cross validated score for K Nearest Neighbors Classifier is: 70.61
```

Figure III.8 – *Résultats obtenus par l'algorithme KPPV.*

- Le taux de trainement appelé aussi taux d'apprentissage (the accuracy) doit être supérieur à 70%.
  D'après les résultats obtenus, on peut juger la performance de notre base de données avec un taux d'apprentissage égale à 78,57%, ce qui implique que 78,57% de notre prédiction est correct.
- Le taux de validation (the cross validation) doit être inférieur à celui d'apprentissage, ce qui est réalisé dans notre cas avec 70,61%.
  Nous avons remarqué que les taux précédents sont influencés par la variation des K et la validité de notre dataset.

III.4.4.2 Machine vecteur de support (SVM)

A partir de la bibliothèque SKLEARN, nous avons fait appel à l'algorithme KPPV comme le montre la figure suivante :

- **Résultats et interprétation :** La figure (III.9) représente la matrice de confusion :

```
Out[15]: Text(0.5, 1.05, 'Confusion_matrix')
```

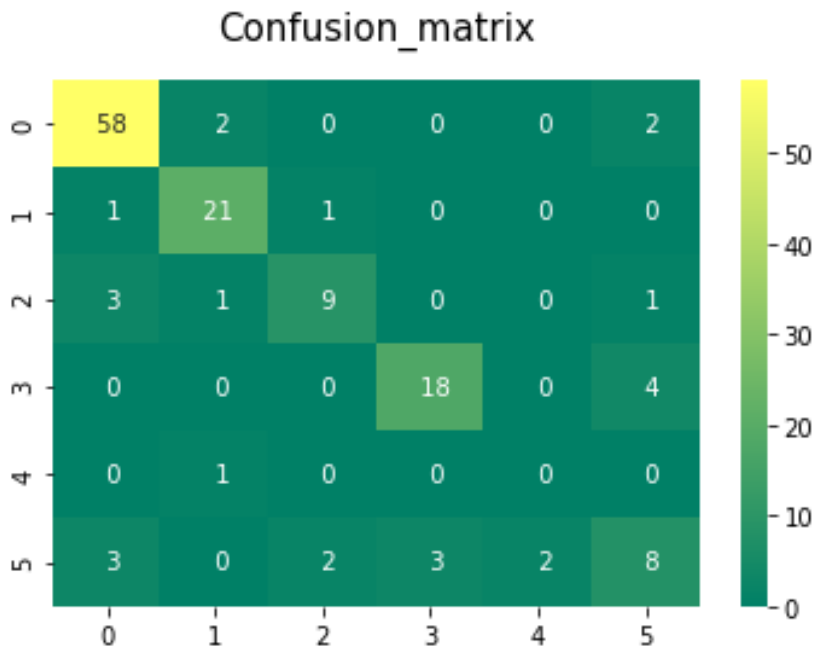


Figure III.9 – *Matrice de confusion du SVM.*

L'explication des valeurs de la matrice de confusion peut être faite de cette manière :

- Élément(0,0) : 58 des données de la classe 1 sont correctement classées par le SVM dans la classe 1.
- Élément(4,4) : 0 des données de la classe 6 sont classées correctement par le SVM dans la classe 6.

  Le reste de l'interprétation se trouve dans l'annexe (C.1.1.2).

- La figure (III.10) représente les taux d'apprentissage et de validation obtenus :

```
Entrée [8]: from sklearn.svm import SVC, LinearSVC

Entrée [9]: model = SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
                        decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
                        max_iter=-1, probability=True, random_state=None, shrinking=True,
                        tol=0.001, verbose=False
                        )
            model.fit(X_train,y_train)
            prediction_svm=model.predict(X_test)
```

Figure III.10 – *Appel à l'algorithme du SVM.*

- D'après les résultats obtenus, on peut juger la performance de notre base de données avec un taux d'apprentissage égale à 85.71%.
- Le taux de validation (the cross validation) obtenu est égal à 81.39%.

III.4.4.3 Forêts aléatoires

Nous présentons l'appel de l'algorithme Forêts aléatoires dans la figure (III.11).

```
Entrée [15]: # Random Forests
             from sklearn.ensemble import RandomForestClassifier

Entrée [16]: model = RandomForestClassifier(criterion='gini', n_estimators=700,
                                             min_samples_split=10,min_samples_leaf=1,
                                             max_features='auto',oob_score=True,
                                             random_state=1,n_jobs=-1)

Entrée [17]: model.fit(X_train,y_train)
             prediction_rm=model.predict(X_test)
```

Figure III.11 – *Appel à l'algorithme du Forêts aléatoires.*

- **Résultats et interprétation :** La figure (III.9) représente la matrice de confusion :

```
Out[23]: Text(0.5, 1.05, 'Confusion_matrix')
```

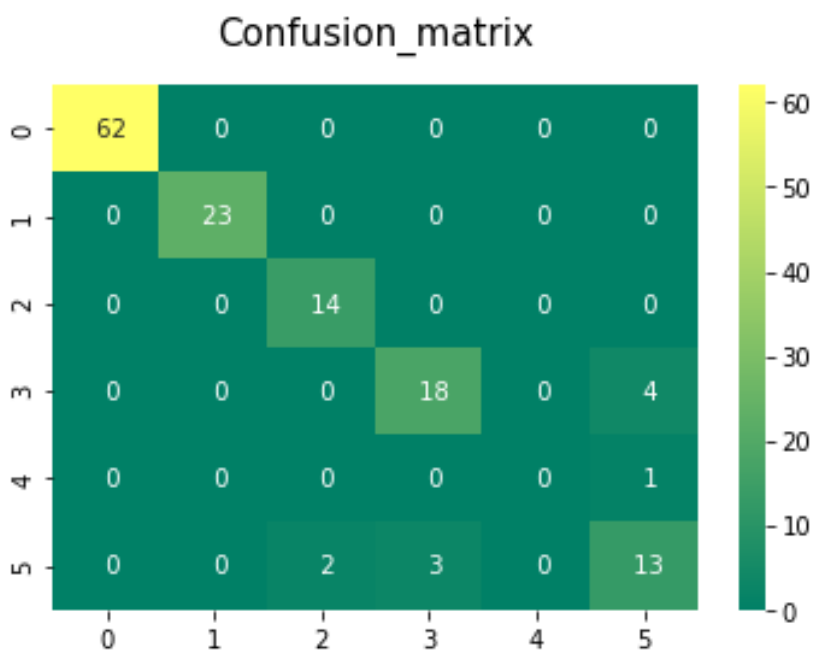


Figure III.12 – *Matrice de confusion du Forêt aléatoires.*

L'explication des valeurs de la matrice de confusion peut être faite de cette manière :

- Élément(0,0) : 62 des données de la classe 1 sont correctement classées par le FA dans la classe 1.
- Élément(4,4) : 0 des données de la classe 6 sont classées correctement par le FA dans la classe 6.

  Le reste de l'interprétation se trouve dans l'annexe (C.1.1.3).

- La figure (III.13) représente les taux de apprentissage et de validation obtenus :

```
Entrée [24]: print('--------------The Accuracy of the model----------------------------')
             print('The accuracy of the Random Forest Classifier is',round(accuracy_score(prediction_rm,y_test)*100,2))
             print('The cross validated score for Random Forest Classifier is:',round(result_rm.mean()*100,2))

--------------The Accuracy of the model----------------------------
The accuracy of the Random Forest Classifier is 92.86
The cross validated score for Random Forest Classifier is: 92.82
```

Figure III.13 – *Résultats obtenus par l'algorithme Forêts aléatoire.*

- D'après les résultats obtenus, on peut juger la performance de notre base de données avec un taux d'apprentissage égale à 92.86%, représente le meilleur résultats parmi les trois algorithmes.
- Le taux de validation (the cross validation) obtenu est égal à 92.82%, nous somme donc dans un bon état.

On peut noter donc que l'algorithme le plus efficace est celui qui a donné le taux le plus élevé, dans notre cas, l'algorithme Forêts aléatoire est alors le plus efficace pour notre dataset.

Pour confirmer cela, on propose d'utiliser un algorithme comparatif proposé dans l'étape suivante.

III.4.5 Mise en place d'un algorithme comparatif

Le principe de cet algorithme est principalement la comparaison entre les algorithmes utilisés au préalable afin de choisir le plus convenable à notre cas d'étude, la figure (III.14) montre les resultats obtenus.

```
RandomForest Test Accuracy is: 0.75
KNN Test Accuracy is: 0.7142857142857143
SVM Test Accuracy is: 0.6428571428571429
Classifier with best accuracy :RandomForest
```

Figure III.14 – *L'affichage de l'algorithme comparatif.*

D'après les résultats obtenus par l'algorithme comparatif, l'hypothèse précédente est donc confirmée et le meilleur choix est le classifieur Forêts aléatoire.

III.5 Vérification des résultats

Afin de vérifier les résultats obtenus précédemment, nous avons appliqué les trois algorithmes sur un cas réel au cours de 12 mois, où nous avons classé nos paramètres manuellement après faire une comparaison entre la classe théorique (obtenue manuellement) et les classes obtenues par les algorithmes. Nous avons obtenu les résultats illustrés dans les figures (III.15,III.16,III.18,III.19) :

Out[3]:

	CT	K-PPV	SVM	FA
0	C1	C1	C1	C1
1	C7	C7	C7	C7
2	C2	C2	C2	C2
3	C5	C5	C2	C5
4	C6	C7	C6	C7
5	C3	C3	C3	C3
6	C2	C2	C2	C2
7	C1	C1	C1	C1
8	C5	C2	C2	C5
9	C7	C1	C7	C7
10	C7	C2	C2	C2
11	C5	C5	C2	C5

Figure III.15 – *Tableau représentatif des classes.*

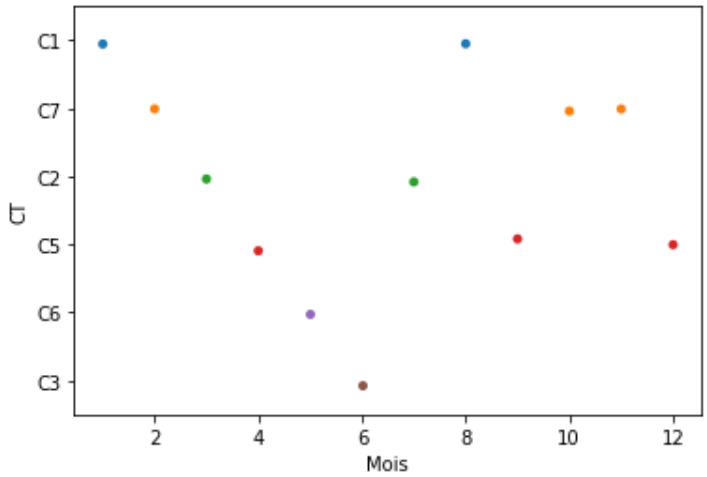


Figure III.16 – *Visualisation des classes Théoriques.*

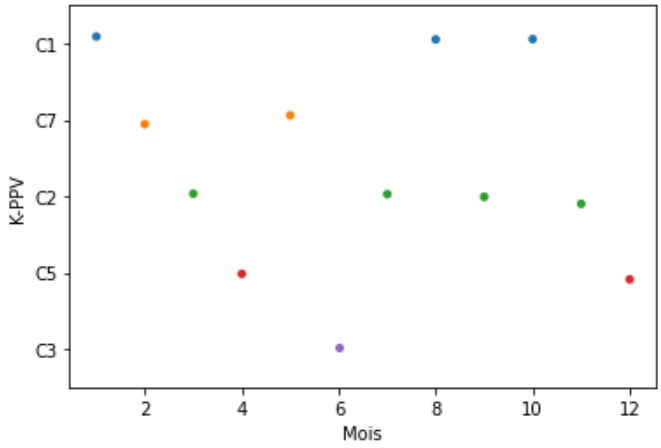


Figure III.17 – *Classification par K-PPV.*

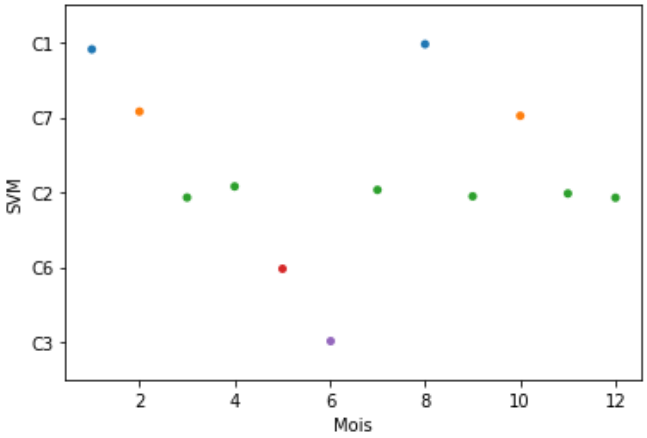


Figure III.18 – *Classification par SVM.*

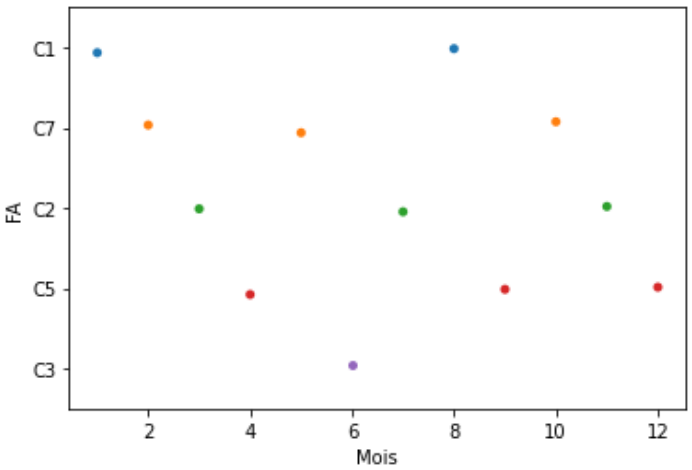


Figure III.19 – *Classification par FA.*

Après avoir visualisé les résultats de classification dans des graphe, il est clair que l'algorithme FA est plus efficace avec seulement deux erreurs commises, suivi par le SVM avec

4 erreurs, enfin le K-PPV avec 5 erreur commises, ce qui confirme les résultats obtenus au préalable.

III.6 Interface graphique

L'interface graphique est une fenêtre qui permet à l'utilisateur d ?entrer les paramètres de système recommandé (Taux de défaillance, taux de performance, taux de réparation, taux de disponibilité). Et après le choix de l'algorithme (SVM, KNN ou Random Forest), l'interface affiche les résultats du système sélectionné sous une classe.
L'ouverture de l'interface affiche la fenêtre montrée dans la figure(III.20).

Figure III.20 – *La fenêtre principale de l'interface graphique.*

La figure(III.21) montre comment l'utilisateur peut faire entrer les valeurs des paramétrés.

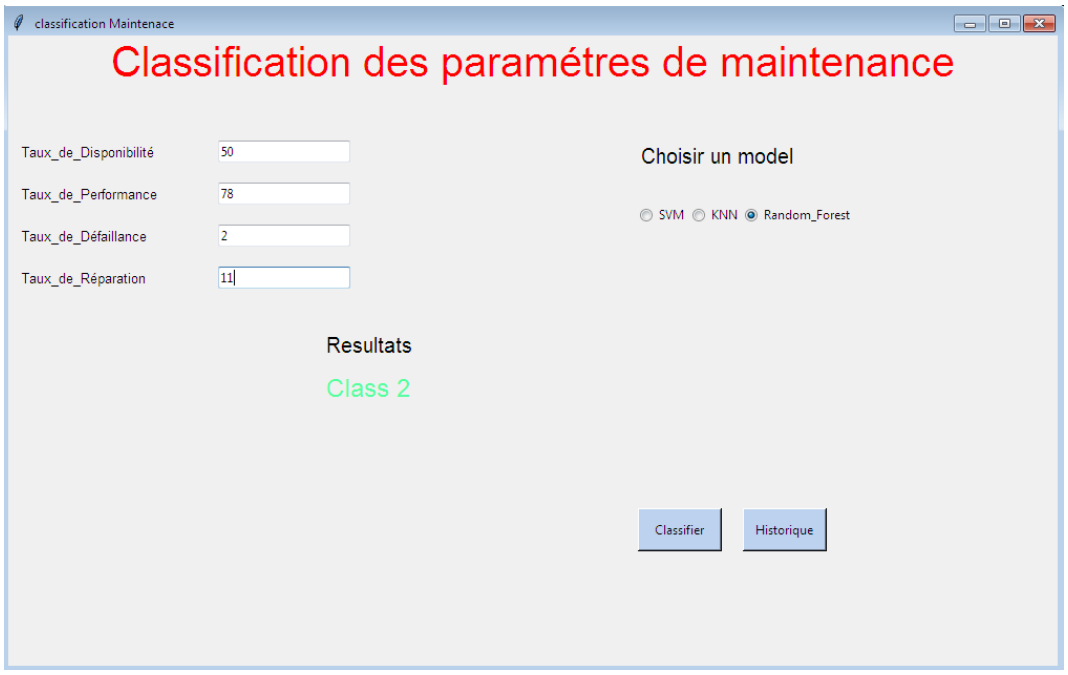


Figure III.21 – *Exemple de classe dans l'état normal.*

En cas où le système détecte la classe 5 il affiche un danger, dans le cas de détection des classes 6 ou la classe 7 il affiche une fenêtre indique une Alerte à l'utilisateur .

La figure(III.22) illustre l'affichage de la classe 5.

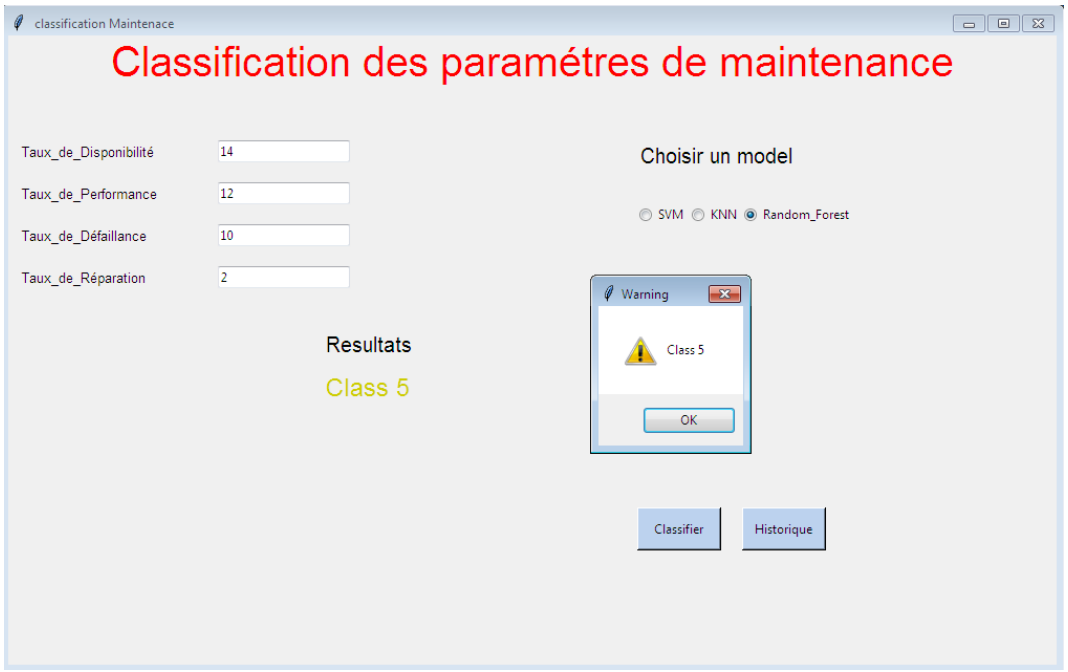


Figure III.22 – *Exemple de classe dans l'état de danger.*

La figure(III.23) illustre l'affichage de la classe 6.

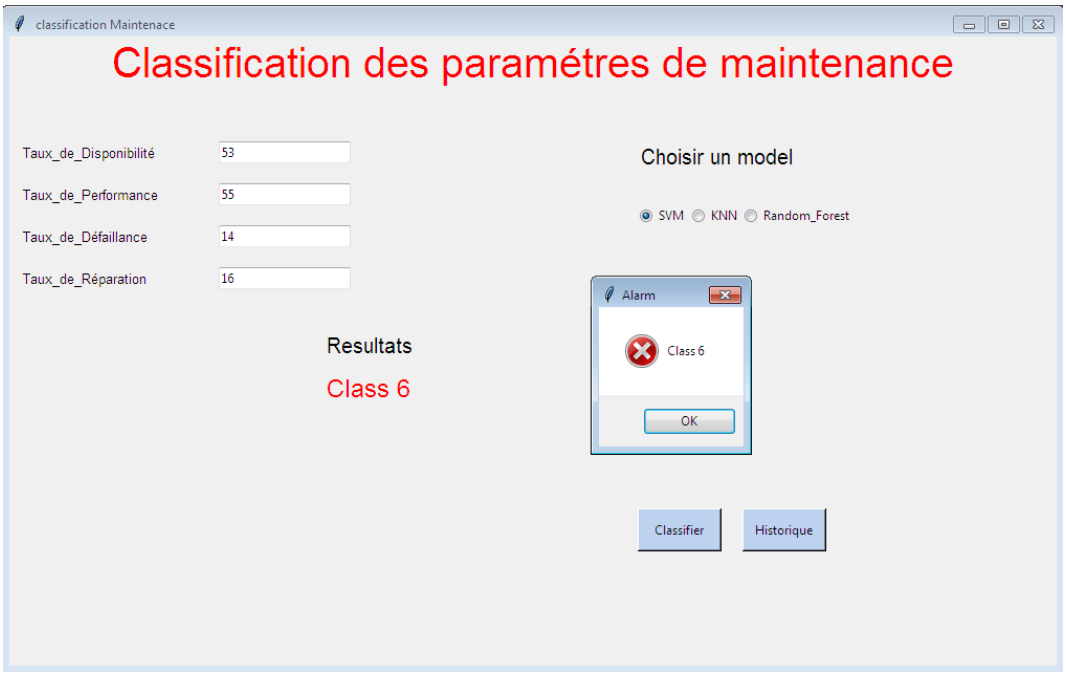


Figure III.23 – *Exemple de classe dans l'état d'alerte.*

La figure(III.24) illustre l'affichage de la classe 7.

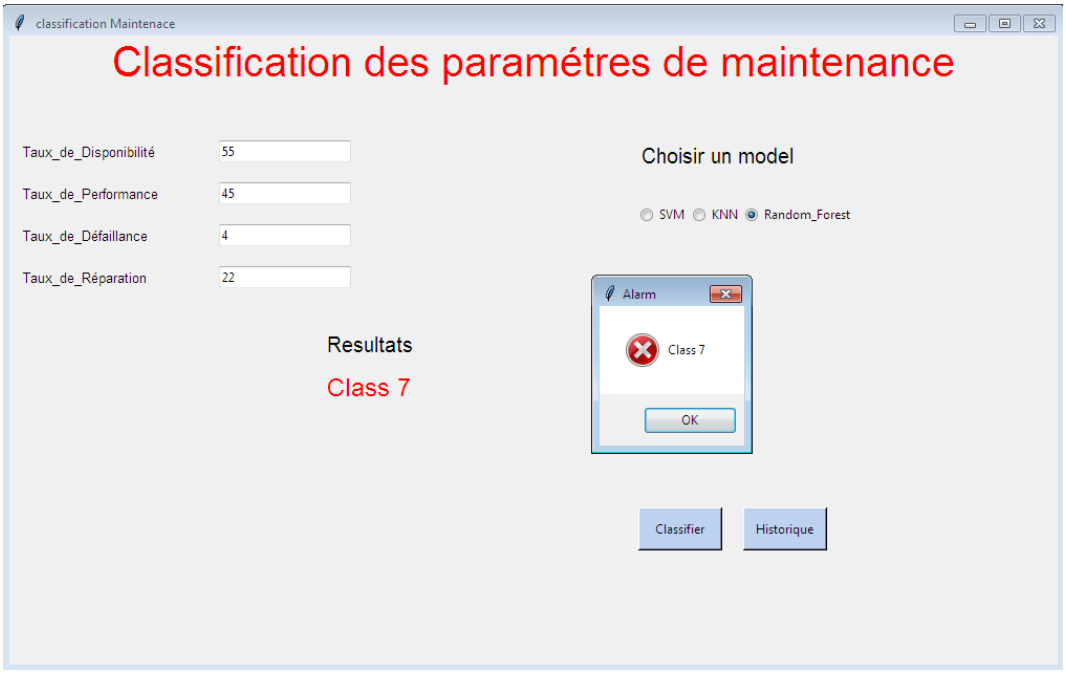
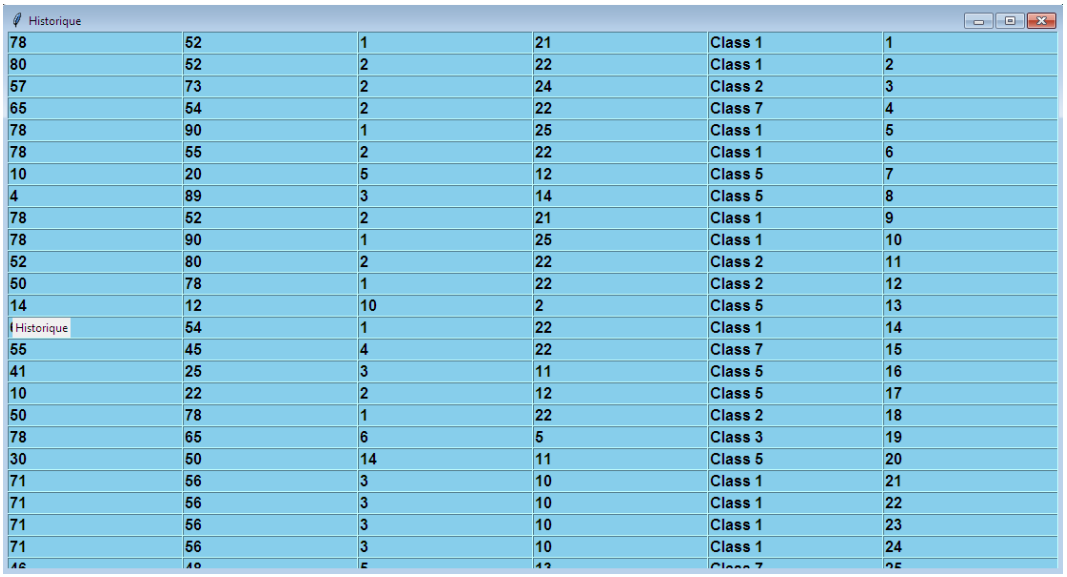


Figure III.24 – *Exemple de classe dans l'état d'alerte.*

Pour rendre l'expérience utilisateur plus rentable, une base des données a été créé en affichant l'historique de la classification. La figure(III.25) représente l'historique.


Historique

78	52	1	21	Class 1	1
80	52	2	22	Class 1	2
57	73	2	24	Class 2	3
65	54	2	22	Class 7	4
78	90	1	25	Class 1	5
78	55	2	22	Class 1	6
10	20	5	12	Class 5	7
4	89	3	14	Class 5	8
78	52	2	21	Class 1	9
78	90	1	25	Class 1	10
52	80	2	22	Class 2	11
50	78	1	22	Class 2	12
14	12	10	2	Class 5	13
Historique	54	1	22	Class 1	14
55	45	4	22	Class 7	15
41	25	3	11	Class 5	16
10	22	2	12	Class 5	17
50	78	1	22	Class 2	18
78	65	6	5	Class 3	19
30	50	14	11	Class 5	20
71	56	3	10	Class 1	21
71	56	3	10	Class 1	22
71	56	3	10	Class 1	23
71	56	3	10	Class 1	24

Figure III.25 – *Exemple d"historique de la classification.*

III.7 Conclusion

Ce chapitre a poursuis une certain démarche d'application des méthodes de classification supervisée, afin de classifier des paramètres de maintenance.

Après avoir appliquée cette démarche nous avons obtenu des résultats qui nous ont permet de choisir la méthode la plus convenable à notre cas d'étude.

Enfin, pour simplifier l'utilisation de ces méthodes à l'utilisateur, nous avons proposer une interface graphique.

"Si on savait ce qu'on savait, on serait trois fois plus efficace ".

Conclusions générales et perspectives

Dans le cadre de notre mémoire de fin d'étude, nous nous sommes intéressés à la classification supervisée des paramètres de maintenance. Notre objectif consiste à classifier des paramètres collectés dans l'entreprise GICA Ain Touta. Pour ce faire, nous avons proposé d'appliquer trois différents algorithme de classification supervisée. Nous avons par la suite choisit l'algorithme le plus précis , En se basant sur plusieurs critères et sur un algorithme comparatif, afin d'aider l'utilisateur à savoir l'état du système pour décider quels sont les actions de maintenance à appliquer.

Avant d'aborder notre démarche, nous avons présenté des généralités sur la classification et les algorithmes les plus connues qui ont été citées dans la littérature scientifique. Ensuite, nous avons présenté les méthodes et les outils pour décider lequel adopter dans notre mémoire.

Finalement, nous avons proposé et appliqué une démarche afin de choisir le meilleur algorithme. Nous avons également élaboré une interface graphique pour simplifier à l'utilisateur la classification. Ce qui simplifiera galamment les taches des opérateurs du service de maintenance et optimisera le temps des interventions quotidienne.

À travers les essais qui ont été effectués dans le cadre de ce mémoire, nous avons pu choisir l'algorithme qui convient le mieux avec notre cas d'étude. en l'occurrence la méthodes des foret aléatoire qui donne un tau d'apprentissage et tau de validation les plus optimum

Malgré les bons résultats obtenus, on remarque que le choix des paramètres et des classes, reste une étape critique et décisive, pour cela, et dans une perspective d'amélioration de notre approche, nous nous proposons :

- Formation d'un groupe de maintenance spécialisé dans le domaine de l'intelligence artificielle (un groupe avec un retour d'expérience).
- Utilisation des données volumineuse (riches) et plus performantes pour enrichir le dataset et améliorer la précision.
- Accompagner chaque classe par l'intervention nécessaire à prendre dans l'interface.
- La réalisation d'une application web API ou une application sur les smartphones qui met en oeuvre une interface utilisateur permet de rendre l'utilisation de l'application plus facile.
- Intégration de l'interface graphique dans la GMAO.

Bibliographie

[1] Ali Mustafa Qamar, Eric Gaussier Apprentissage de différentes classes de similarité dans les K-PPV 2016.

[2] Introduction aux méthodes d'agrégation : boosting, baggin et forêts aléatoires. Illustrations avec R. université Rennes 2.

[3] Baïna S., Panetto H. et Benali K., Apport de l'approche MDA pour une interopérabilité sémantique : Interopérabilité des systèmes d'information d'entreprise, Processus d ?entreprise et SI, RSTI-ISI, pp.11-29, novembre 2006.

[4] BARIGOU Fatiha CONTRIBUTION À LA CATÉGORISATION DE TEXTES ET À L'EXTRACTION D'INFORMATION.

[5] BENCHETTOUH Salah Eddine Elaboration d'un système de prédiction des pannes et de planification des maintenances.

[6] Leo Breiman, Random Forests 1999-2001.

[7] Bruno Taconet, Abderrazak Zahour, Saîd Ramadane, Wafa Boussella Cllassification des K-PPV par sous-voisinages embloîtés 2006.

[8] Boucly F., Le management de la maintenance : Evolution et mutation, Editions Afnor, 1998.

[9] Chapman et Hall DATA Classification Algorithms and Applications.

[10] Christophe Chesneau éléments de classification 2016.

[11] A.Cornuéjols, L.Miclet, Y.Kodratoff Apprentissage artificiel EYROLLES.

[12] Abdelhamid DJEFFAL.Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données Université Mohamed Khider, Biskra Abdelhamid DJEFFAL.

[13] Ali Mustafa Qamar, Eric Gaussier Similarity learning in Nearest Neighbor and Application to information Retrieval 2014.

[14] Ethem Alpaydin Introduction to machine learning.

[15] Eve Mathieu-Dupas Algorithme des k plus proches voisins pondérés et application en diagnostic 2010.

[16] Fabien Torre un algorithme stochastique pour l'apprentissage supervisé et non-supervisé.

[17] Faïcel Chamroukhi.Classification supervisé : les K-plus proches voisins, université de Caen.

[18] M. ZAIZ Faouzi Les Supports Vecteurs Machines (SVM) pour la reconnaissance des caractères manuscrits arabes université Mohamed Khider Biskra

[19] Francastel J.C., Externalisation de la maintenance : Stratégies, méthodes et contrats. Dunod, Paris, 2003.

[20] K.Gosalia,Ph.D., CFA, CPA, CGA Rock Lefebvre, MBA, FCIS, FCPA, FCGA Introduction à l'apprentissage automatique ; MONOGRAPHIE DE CPA NOUVEAU-BRUNSWICK ;

[21] Kaffel H., La maintenance distribuée : concept, évaluation et mise en oeuvre. Thèse de doctorat, Université Laval, Quebec, 2001.

[22] Kahn J., Overview of MIMOSA and the Open System Architecture for Enterprise Application Integration. Proc. of COMADEM 2003, pp. 661-670, Växjö University, Sweden, 2003.

[23] kamalesh Gosalia, Roch Lefebvre Introduction à l'apprentissage automatique.

[24] F.Karem, M.Dhibi, A.Martin Combinaison de classification supervisée et non-supervisée par la théorie des fonctions de croyance.

[25] A.LOUGHANI + JNB cours.

[26] Introduction to Machine Learning with Python Andreas C. Müller Sarah Guido.

[27] Norman Matloff Statistical Regression and Classification.

[28] Rachid MIFDAL PReSENTe a L'eCOLE DE TECHNOLOGIE SUPeRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE AVEC MeMOIRE EN GeNIE, CONCENTRATION PERSONNALISeE M. Sc. A.

[29] F.Monchy, J.Vernier MAINTENANCE Méthodes et organisations DUNOD.

[30] Rudolph Russell Machine learning.

[31] I.Rasovska, B.Chebel-Morello, N.Zerhouni. Classification des différentes architectures en maintenance.