République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira de Bejaia

Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Recherche en Informatique

Option : Intelligence Artificielle

Méthodes d'apprentissage automatique pour la détection des défaillances d'oléoducs

---

Réalisé par :
M^{lle} Lydia Souici
M^{lle} Sofia Chergui

Encadré par :
M Kamal Amroun
Co-encadré par :
M Kamal Souadih

Soutenu le 27 Septembre 2022, Devant le jury composé de :

Mme. Soraya Tighidet : - Présidente
M. Zoubeyr Farah : - Examinateur
Mme. Fatima Ait Hatrit : - Examinatrice

Promotion : 2021/2022

Remerciements

En préambule à ce mémoire, nous tenons tout d'abord à remercier ALLAH le tout puissant et miséricordieux, qui nous aide et qui nous a donné la force, le courage et la patience d'accomplir ce Modeste travail

Nous tenons à remercier tout particulièrement notre encadrant **M.Amroun kamal**, pour l'aide compétente qu'il nous a apportée et pour sa patience et son encouragement. Son œil critique nous a été très précieux pour structurer le travail et pour améliorer la qualité des différentes sections.

Nous désirons remercier également notre maître de stage **M.Souadih Kamal**, pour sa confiance et les connaissances qu'il a su partager avec nous. Nous le remercions aussi pour sa disponibilité et la qualité de son encadrement en entreprise.

Que les membres de jury trouvent, ici, l'expression de nos sincères remerciements pour l'honneur qu'ils nous font en prenant le temps de lire et d'évaluer ce travail.

Un grand merci à nos familles et plus particulièrement à nos parents, nos frères et nos soeurs pour leur amour, leur confiance, leurs conseils ainsi que leur soutien inconditionnel qui nous ont permis de réaliser les études pour lesquelles nous nous destinons et par conséquent ce mémoire.

Nous souhaitons particulièrement remercier nos amies de toujours, **Lilia** ,**Nassima**, **Wissam** et **Yamina** pour leurs accompagnements, leurs soutiens et amitiés durant toutes ces années .

Résumé

Les oléoducs représentent le moyen le plus sûr de transport des hydrocarbures. Cependant, des accidents peuvent se produire, la maintenance est ainsi nécessaire pour parvenir à limiter les dégâts. La détection de défaillances est une tâche importante dans la maintenance des oléoducs. En connaissant le type de panne à l'avance, les ingénieurs pourraient prévoir le processus de réparation. Le modèle que nous proposons pourrait servir d'aide à la décision. Plusieurs algorithmes ont été appliqués sur le dataset issu du CONCAWE (CONservation of Clean Air and Water in Europe). Deux métriques de performances ont été employées dans le réglage des paramètres ; la moyenne géométrique et l'accuracy. La moyenne géométrique a été utilisée car elle essaie de maximiser la précision des classes tout en gardant ces précisions équilibrées. Le meilleur algorithme par rapport à l'accuracy est SVM qui a atteint 65% et Random Forest qui a atteint 64%. Les résultats ont aussi montré une meilleure classification pour la classe *Third party* qui est de 87% de Recall et de 69% de précision pour l'algorithme SVM.

Mots clés : apprentissage automatique, MLOps, hydrocarbure, panne d'oléoduc, classification, pétrole, industrie pétrolière

Abstract

Pipelines are the safest means of transporting hydrocarbons. However, accidents can occur, therefore maintenance is necessary to limit the damage. Failure detection is an important task in pipeline maintenance. By knowing the type of failure in advance, engineers can overcome serious issues in the repair process. The models we propose could serve as a decision aid. Several algorithms were applied to the dataset from CONCAWE (CONservation of Clean Air and Water in Europe). Two performance metrics were employed in parameter tuning ; geometric mean and accuracy. Geometric mean was used because it tries to maximize the accuracy of the classes while keeping these accuracies balanced. The best algorithms are SVM which reached an accuracy of 65% and Random Forest which reached 64%. The results also showed a better classification for the class *Third party* which is 87% of Recall and 69% of precision with the SVM algorithm.

Keywords : oil, machine learning, classification, petroleum industry, MLOps, hydrocarbons, pipeline failure.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

ML	Machine Learning
AI	Artificial Intelligence
IA	Intelligence Artificielle
SVM	Support Vector Machines
RF	Random Forest
KNN	K Nearest Neighbors
DT	Decisio Tree
NB	Naive Bayes
MLOps	Machine Learning Operations
DevOps	Developpement Operations
RN	Réseau de Neurones
TP	True Positive
TPR	True Positive Rate
TN	True Negative
FP	False Positive
FN	False Negative
ETL	Extract Trasform Load
CI	Continuous Integration
CD	Continous Delivery
CT	Continuous Training
CONCAWE	CONservation of Clean Air and Water in Europe
RTC	Région de Transport Centre
ILI	In Line Inspection
G-Mean	Geometric Mean
SMOTE	Synthetic Minority Oversampling Technique

Introduction Générale

Une fois le pétrole brut séparé du gaz naturel, des pipelines véhiculent le pétrole vers un autre transporteur ou directement vers une raffinerie. Les produits pétroliers voyagent ensuite de la raffinerie au marché par camion-citerne, camion, wagon-citerne ou pipeline. Les pipelines, ou oléoducs, sont constitués de tubes en acier ou en plastique qui sont généralement enterrés. Le pétrole est déplacé à travers les pipelines par des stations de pompage le long du pipeline. Les pipelines sont l'un des moyens les plus sûrs de transporter des matériaux comparativement à la route ou au rail. Cependant, des accidents se produisent chaque année et certains de ces accidents ont un impact catastrophique sur l'environnement et entraînent de grandes pertes économiques. Afin de maintenir la sécurité des pipelines, plusieurs techniques d'inspection ont été développées au cours des dernières décennies. Malgré l'efficacité de ces techniques, elles sont très coûteuses et chronophages. De même, plusieurs modèles de prévision de défaillances et d'évaluation de l'état des oléoducs ont été développés au cours de la dernière décennie ; cependant, la plupart de ces modèles sont limités à un type de défaillance, comme la défaillance par corrosion, ou dépendent principalement de l'opinion d'experts, ce qui rend leur sortie subjective.

Lors de notre stage effectué au sein de Sonatrach, on nous a confié la tâche d'intégration de l'apprentissage automatique en entreprise. Nous nous sommes ainsi intéressées à la possibilité de développement d'un modèle objectif de prédiction des défaillances des oléoducs en fonction des données historiques disponibles sur les accidents de pipelines. Ce modèle servira d'aide à la décision pour la prévention de ces pannes. Dans le but d'intégration de ce modèle en entreprise, nous avons proposé de le développer en utilisant un framework de gestion de cycle de vie ; MLflow.

Du au manque de documentation en langue française, nous avons introduit certains termes en anglais, pour d'autres, en revanche, nous avons fait de notre mieux pour donner une traduction fiable.

Notre mémoire est organisé comme suit. Dans le premier chapitre, nous abordons les principaux concepts de l'apprentissage automatique. Ce chapitre est nécessaire et servira de référence pour expliquer les techniques d'apprentissage automatique employées dans notre étude. Comme nous proposons notre modèle à une entreprise, le deuxième chapitre présentera les meilleures pratiques à considérer lors du développement d'un modèle ML pour une entreprise. Dans le troisième chapitre, nous présenterons notre étude et dans le quatrième chapitre, nous donnons des détails de notre implémentation et des résultats obtenus.

Chapitre 1

Concepts généraux d'apprentissage automatique

1.1 Introduction

L'avènement de l'informatique a commencé au $18^{ème}$ siècle avec l'apparition de l'industrialisation en Europe. Vers 1950, une génération de mathématiciens, de scientifiques et de philosophes est apparue avec le concept d'intelligence artificielle assimilé dans leurs esprits. La prolifération des smartphones et la numérisation de tant d'éléments de la vie quotidienne ont créé d'énormes quantités de données. Dans le même temps, la poursuite de la loi de Moore [64], l'idée que l'informatique augmenterait considérablement en puissance et diminuerait en coût relatif au fil du temps, a rendu la puissance de calcul bon marché largement disponible. La science des données existe comme le lien entre ces deux innovations. Jusqu'à la fin des années 1970, l'apprentissage automatique, ou Machine Learning, faisait partie de l'évolution de l'Intelligence Artificielle [15]. Ensuite, elle a bifurqué pour évoluer comme une branche à part. Avec ses applications nombreuses et variées telles que les moteurs de recherche et de la reconnaissance de caractères, la recherche en génomique, l'analyse des réseaux sociaux et la vision par ordinateur, le machine learning permet d'automatiser des processus que des humains avaient l'habitude d'exécuter.

Dans les sections suivantes, nous donnons un aperçu sur quelques concepts d'apprentissage automatique. En premier lieu, nous définissons celle-ci. Ensuite, pour éclaircir d'avantage le sujet, nous citons les principales catégories de l'apprentissage automatique. Dans le processus de résolution d'un problème donné, plusieurs algorithmes ML sont disponibles. Ce processus de résolution est décrit par un modèle qui devrait être bien adapté au type de problème. La section suivante, met en avant des considérations à prendre lors de la formation de ces modèles. Avant de parler des algorithmes ML, nous évoquons quelques notions d'apprentissage automatique en relation avec la manipulation des données et les propriétés des algorithmes pour enfin, décrire ces algorithmes.

1.2 Définition de l'apprentissage automatique

L'apprentissage automatique également appelé apprentissage machine ou apprentissage artificiel et en anglais Machine Learning, est défini selon Arthur Samuel [79], comme le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmé. Comme le montre la figure 1.1 l'apprentissage automatique s'appuie sur différents algorithmes,pour résoudre problèmes de données.Le type d'algorithme utilisé dépend du type de problème à résoudre, le nombre de variables, le type de modèle qui lui conviendrait le mieux, etc [57].

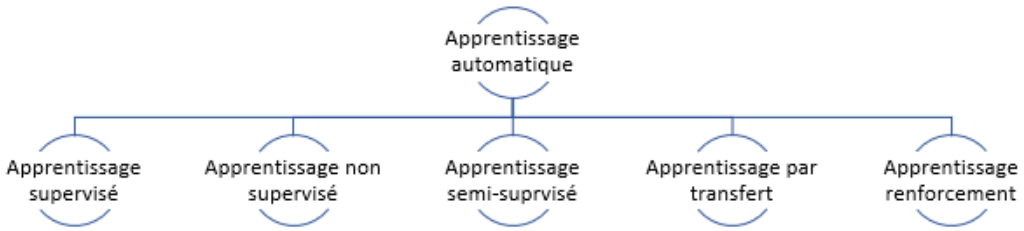


FIGURE 1.1 – Types d'algorithme Machine Learning
[57]

Avec les progrès de la puissance de calcul et des machines très efficaces, il y a eu une amélioration significative de l'efficacité des algorithmes d'apprentissage automatique, au fur et à mesure que ces derniers utilisent plus de données il devient possible de créer des modèles plus précis basés sur ces données.

Le Machine Learning est présent dans plusieurs champs d'études, comme la médecine et les systèmes d'apprentissage automatique médicaux en particulier. Les arbres de décisions donnent de bons résultats dans le domaine de la cardiologie ,exactement la prédiction de risque de mortalité [19]. de façon globale, les algorithmes d'apprentissage automatique et d'apprentissage en profondeur appliqués dans le domaine de la santé permettent aux personnel médical de surveiller, diagnostiquer, cibler et mettre en évidence la région du problème et de proposer la solution requise et précise dans les plus brefs délais [29].

1.3 Catégories d'apprentissage automatique

Les algorithmes d'apprentissage automatique peuvent être divisés en 5 grandes catégories [70].

1.3.1 Apprentissage supervisé

L'apprentissage supervisé commence généralement par un ensemble de données bien défini et une certaine compréhension de la façon dont ces données sont classifiée [42]. Dans ce type d'apprentissage , la variable de sortie ou de réponse souhaitée est connue, et l'algorithme d'apprentissage automatique fournit un mappage entre les caractéristiques d'entrée et la variable de sortie (voir la figure 1.2. Les deux principales sous-catégories de l'apprentissage supervisé sont les problèmes de régression et de classification [70], qui sont dictés par le type de variable de sortie. Lorsque la variable de sortie est continue, elle entre dans la catégorie de régression. D'autre part, pour les problèmes de classification, la variable de sortie contient plusieurs classes ou étiquettes. Dans l'apprentissage supervisé, le processus d'apprentissage du modèle se poursuit avec l'évaluation de l'erreur et la améliorations jusqu'à ce que le niveau de précision souhaité soit atteint.

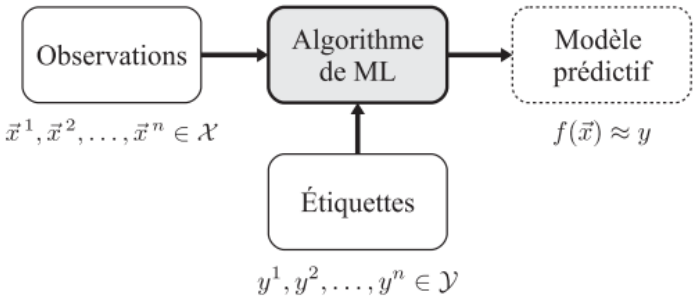


FIGURE 1.2 – Apprentissage supervisé [77].

1.3.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé, il n'y a pas de variable de sortie explicite et les relations sont générées en fonction des données fournies à l'algorithme. Certains des algorithmes appartenant à cette catégorie peuvent révéler des structures et des relations cachées entre les entités d'entrée. Certains des exemples d'apprentissage non supervisé comprennent le regroupement, les algorithmes de réduction de dimensionnalité et l'apprentissage de règles associatives. Son objectif étant de classifier et d'inférer des connaissances à partir de ces données selon des similarités, des différences ou d'autres motifs qu'il découvre au fur et à mesure à partir de nouvelles données, sans entraînements préalables[67].

1.3.3 Apprentissage par renforcement

Cette catégorie d'algorithmes est conçue de manière à ce qu'il y ait une récompense ou une pénalité associée à la séquence de décisions prises par l'algorithme. Une récompense ou une pénalité aide l'algorithme à apprendre l'ensemble des décisions

qu'il doit prendre pour atteindre un objectif défini. L'objectif de ce type est de permettre à un agent (machine ou robot etc.) d'apprendre à l'aide de ses interaction avec son environnement et de tirer profit des expériences passées afin d'optimiser sa performance et de réduire ses erreurs suivant un algorithme bien précis [67]. La robotique pour l'automatisation industrielle est un exemple d'applications d'apprentissage par renforcement

1.3.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé est une approche de l'apprentissage automatique avec objective de résoudre des problèmes similaires à ceux de l'apprentissage supervisé.C'est une combinaison de l'apprentissage supervisé et non supervisé où dans les données d'entraînement on retrouve des données étiquetées et non étiquetée [67].

Ce type d'apprentissage consiste à apprendre des étiquettes à partir d'un jeu de données partiellement étiqueté.L'avantage de cette approche est qu'elle permet d'éviter d'avoir à étiqueter l'intégralité des exemples d'apprentissage, ce qui est pertinent quand il est facile d'accumuler des données mais que leur étiquetage requiert une certaine quantité de travail humain [6].

1.3.5 Apprentissage par transfert

L'apprentissage par transfert, ou bien Transfert Learning, se concentre sur le stockage des connaissances acquises lors de la résolution d'un problème pour résoudre un problème différent, mais qui présente des similitudes [69].

Exemple 1.3.1. *Les connaissances acquises par un algorithme d'apprentissage automatique pour reconnaître des voitures peuvent ensuite être transférées pour être utilisées dans un autre modèle d'apprentissage automatique créé, pour reconnaître d'autres types de véhicules, tels que des camions [89]*

1.4 Considérations sur la formation des modèles

Avec la disponibilité d'un grand nombre d'algorithmes dans les boîtes à outils de science de données, il devient parfois difficile de comprendre quel algorithme doit être utilisé pour résoudre un problème. De nombreux algorithmes peuvent résoudre le même problème et apprendre la relation entre les caractéristiques d'entrée et la variable de sortie. Cependant, la technique et le processus d'apprentissage adoptés par différents algorithmes peuvent être sensiblement différents. Un algorithme peut surpasser d'autres algorithmes lorsque certains paramètres du modèle sont modifiés. Le processus de formation du modèle implique des paramètres supplémentaires appelés hyperparamètres, qui peuvent inclure le nombre d'itérations pour la formation du modèle, la fraction de données d'apprentissage (taille du lot ou batch size en anglais) lors de chaque itération et la fraction d'erreur d'estimation propagée pour modifier

les paramètres du modèle (taux d'apprentissage). Le processus itératif de réglage de ces hyperparamètres pour apprendre les paramètres de modèle optimaux est connu sous le nom d'optimisation des hyperparamètres.

En plus des paramètres de modèle optimaux, la sélection du nombre et du type optimaux d'entités en entrée peut également améliorer la précision d'un modèle. Cela fait de l'ingénierie et de la sélection des fonctionnalités des aspects très importants du processus d'apprentissage automatique.

Chaque algorithme d'apprentissage automatique est associé à trois composants principaux : la représentation, l'optimisation et l'évaluation [27].

Les fonctions sont représentées sous la forme de modèles de graphes numériques, symboliques, basés sur des instances ou probabilistes [63]. Pour améliorer les performances de l'algorithme, des méthodes d'optimisation, telles que la descente de gradient, la programmation dynamique [30] ou le calcul évolutif [61], sont employées.

L'évaluation de ces modèles est effectuée au moyen de métriques statistiques, qui peuvent inclure des calculs de précision, de rappel et d'erreur quadratique moyenne.

Les modèles peuvent être soit linéaires ou non linéaires. Pour la classification, le modèle est linéaire si toutes les n caractéristiques peuvent être tracées dans un espace à n dimensions, et qu'il existe une "ligne" dimensionnelle (n-1) (ou plan, ou hyperplan), qui sépare (ou majoritairement sépare) différentes classes.

En régression, un modèle linéaire signifie que si on trace en fonction des caractéristiques un résultat numérique, il existe une ligne (ou un hyperplan) qui estime approximativement le résultat

Tous les autres modèles sont "non linéaires". Ceux-ci sont divisés en deux types. Le premier en classification, où on trouve une fonction à n dimensions (qui n'est pas un hyperplan).

Exemple 1.4.1. *Un modèle linéaire à deux caractéristiques est représenté par une ligne. Un modèle non linéaire à 2 caractéristiques peut être représenté par une courbe (ou n'importe quelle autre dessin géométrique différent d'une ligne droite)*

Les réseaux de neurones et plusieurs autres algorithmes sont non linéaires, mais avec des limites ou des "lignes de meilleur ajustement" potentiellement très complexes/courbées. La deuxième version des modèles non linéaires est non paramétrique. K-plus proches voisins en est un exemple. Il ne recherche pas du tout une ligne/courbe discriminante, et regarde simplement les classes de ses voisins les plus proches.

1.5 D'autres notions d'apprentissage automatique

Pour la suite de ce mémoire il est nécessaire de définir quelques concepts clés dans le domaine d'apprentissage automatique. Dans cette section, nous définissons ces concepts de base sur lesquels repose ces algorithmes.

1.5.1 Préparation des données

La préparation des données est une transformation des data brutes en informations utiles et exploitables. Cette étape permettent une analyse efficace et une limitation des erreurs qui peuvent survenir dans les données lors du traitement et rend toutes ces données traitées plus accessibles aux utilisateurs. La normalisation est une des méthodes qui assure le bon fonctionne de cette étape .

Normalisation est un processus de prétraitement des données, il vise a diminuer la complexité des modèles et mettre sur une même échelle toutes les variables quantitatives. Cette méthode peut être appliquée lorsque les limites supérieures et inférieures approximatives des données sont connues [70]. En termes plus simples, cette procédure implique la suppression des données non structuré ainsi que de la redondance pour assurer le stockage logique des données. Lorsque la normalisation des données est effectuée correctement,la saisie de données normalisée en est le résultat. Ceux-ci sont quelques types de la normalisation :

Normalisation Min-Max : dans ce type l'idée est de ramener toutes les valeurs de la variable dans l'intervalle $[0, 1]$ et des fois $[-1, 1]$, tout en conservant le rapport des distances entre ces valeurs [4].

$$X_{norm]} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1.1}$$

Standardisation (Z-Score Normalization) : signifie généralement Le redimensionnement les données pour avoir une moyenne μ de 0 et un écart-type σ de 1.

$$X_{stand} = \frac{X - \mu}{\sigma} \tag{1.2}$$

1.5.2 Sous-ajustement / sur-ajustement

Le sur-échantillonnage et le sous-échantillonnage sont des techniques utilisées pour ajuster la distribution des classes d'un ensemble de données (c'est-à-dire le rapport entre les différentes classes/catégories représentées). Ces termes sont utilisés à la fois dans l'échantillonnage statistique, la méthodologie de conception d'enquête et l'apprentissage automatique. Le sur-échantillonnage et le sous-échantillonnage sont des techniques opposées et à peu près équivalentes. Il existe également des techniques de sur-échantillonnage plus complexes, notamment la création de points de données artificiels avec des algorithmes tels que la technique de sur-échantillonnage de minorité synthétique.

Techniques de sur-échantillonage

Sur-échantillonage aléatoire : Le sur-échantillonnage aléatoire consiste à compléter les données de formation avec plusieurs copies de certaines des

classes minoritaires. Le sur-échantillonnage peut être effectué plus d'une fois (2x, 3x, 5x, 10x, etc.). C'est l'une des premières méthodes proposées, qui s'est également avérée robuste [27]. Au lieu de dupliquer tous les échantillons de la classe minoritaire, certains d'entre eux peuvent être choisis au hasard avec remise.

SMOTE : ou Synthetic Minority Over-sampling Technique [5]. Pour illustrer le fonctionnement de cette technique, considérons certaines données d'apprentissage qui ont s échantillons et f caractéristiques dans l'espace des caractéristiques des données. Notez que ces caractéristiques, pour simplifier, sont continues. Prenons l'exemple d'un ensemble de données d'oiseaux à classer. L'espace des caractéristiques pour la classe minoritaire pour laquelle nous voulons suréchantillonner pourrait être la longueur du bec, l'envergure et le poids (tous continus). Pour ensuite suréchantillonner, prenez un échantillon de l'ensemble de données et considérez ses k voisins les plus proches (dans l'espace des caractéristiques). Pour créer un point de données synthétique, prenez le vecteur entre l'un de ces k voisins et le point de données actuel. Multipliez ce vecteur par un nombre aléatoire x compris entre 0 et 1. Ajoutez ceci au point de données actuel pour créer le nouveau point de données synthétique. De nombreuses modifications et extensions ont été apportées à la méthode SMOTE depuis sa proposition [27].

ADSYN : L'approche d'échantillonnage synthétique adaptatif, ou algorithme ADASYN,[27] s'appuie sur la méthodologie de SMOTE, en déplaçant l'importance de la frontière de classification vers les classes minoritaires qui sont difficiles. ADASYN utilise une distribution pondérée pour différents exemples de classes minoritaires en fonction de leur niveau de difficulté d'apprentissage, où des données plus synthétiques sont générées pour les exemples de classes minoritaires plus difficiles à apprendre.

Augmentation : L'augmentation des données dans l'analyse des données sont des techniques utilisées pour augmenter la quantité de données en ajoutant des copies légèrement modifiées de données déjà existantes ou de données synthétiques nouvellement créées à partir de données existantes. Il agit comme un régularisateur et aide à réduire le surajustement lors de la formation d'un modèle d'apprentissage automatique [27].

Techniques de sous-échantillonage

Sous-échantillonage aléatoire : Il consiste à retirer au hasard des échantillons de la classe majoritaire, avec ou sans remplacement. C'est l'une des premières techniques utilisées pour atténuer le déséquilibre dans l'ensemble de données, cependant, elle peut augmenter la variance du classificateur et est très susceptible de rejeter des échantillons utiles ou importants.[27]

Groupe : Les centroïdes de cluster sont une méthode qui remplace le cluster d'échantillons par le centroïde de cluster d'un algorithme K-means, où le

nombre de clusters est défini par le niveau de sous-échantillonnage.

1.5.3 Fractionnement de données

Données d'apprentissage : est un ensemble d'échantillons de données initial utilisées pour entraîner les algorithmes et construire les modelés d'apprentissage automatique .

Données de validation : Les données de validation fournissent une première vérification que le modèle peut renvoyer des prédictions utiles dans un environnement réel, ce que les données d'apprentissage ne peuvent pas faire.

Données de test : est un ensemble de données distinct utilisé pour tester et évaluer a nouveau les performances du modèle construit ,par exemple si ce dernier peut faire des prédictions précises.

1.5.4 Apprentissage automatique reproductible

Dans l'industrie , il est important de s'assurer que le modèle d'apprentissage automatique et les prédictions générées par les modèles sont exactement reproductibles [70]. Dans le monde des machines learning , La reproductibilité est la capacité d'exécuter un algorithme à plusieurs reprises, sur des ensembles de données différent et obtenir a chaque fois les mêmes résultats .Elle importante non seulement parce qu'elle garantit que les résultats sont corrects, mais aussi parce qu'elle assure la transparence et donne confiance dans la compréhension exacte de ce qui a été fait [96].

1.5.5 Bagging vs Boosting

Le Bagging et le Boosting sont des techniques qui sert à améliorer les règles de prédiction.Ils aident à réduire le bruit,éviter le sur-ajustement et le sous-ajustement [70] .

Bagging : est une methode appliquée pour améliorer la stabilité du modèle [94].Elle crée des classificateurs pour l'ensemble, en formant chaque classificateur sur une redistribution aléatoire de l'ensemble d'apprentissage [9].

Boosting :Le boosting est une technique générale utilisée pour améliorer les performances des algorithmes d'apprentissages qui génère des classificateurs avec des erreurs de classification inférieures à 50% sur un problème donné [9].

Alors que Les algorithmes de Boosting sont considérés comme plus puissants que le bagging sur des données sans bruit, le Bagging est beaucoup plus robuste que le boosting dans des environnements bruyants [51].

1.5.6 Interprétabilité d'un modèle

Le type de décisions et de prédictions prises par les systèmes basés sur l'apprentissage automatique devient critique pour la vie et le bien-être personnel. La nécessité

de faire confiance à ces systèmes basés sur l'IA est primordiale. Cependant, un problème clé qui se pose surtout à mesure que la complexité du modèle augmente est l'interprétabilité. Il est difficile de définir (mathématiquement) l'interprétabilité d'un modèle d'apprentissage automatique. Miller (2013) [62] donne une définition (non mathématique) qui est la suivante : l'interprétabilité est le degré auquel un humain peut comprendre la cause d'une décision d'un classifieur ML. Une autre est : l'interprétabilité est le degré auquel un humain peut prédire de manière cohérente le résultat du modèle [50]. Un modèle est mieux interprétable qu'un autre modèle si ses décisions sont plus faciles à comprendre pour un humain que les décisions de l'autre modèle.

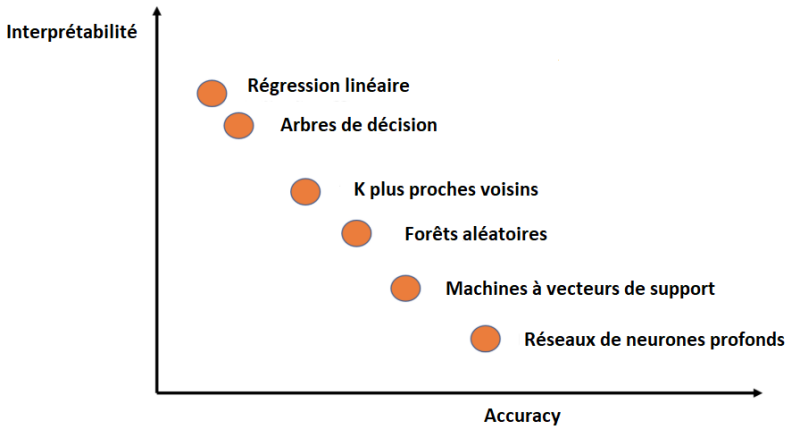


FIGURE 1.3 – L'interprétablité en fonction du taux de précision de quelques algorithmes d'apprentissage automatique [21]

L'apprentissage automatique interprétable est un terme générique utile qui capture "l'extraction de connaissances pertinentes à partir d'un modèle d'apprentissage automatique concernant les relations contenues dans les données ou apprises par le modèle". La figure 1.3 montre le classement de quelques algorithmes d'apprentissage automatique selon leur interprétabilité. On remarque que les réseaux de neurones sont les modèles les moins interprétables. En effet, ceux-ci sont une boite noire ; on ne sait pas comment tous les neurones individuels travaillent ensemble pour arriver à la sortie finale du réseau. Souvent, on ne sait même pas ce que fait un neurone particulier par lui-même.

1.6 Quelques algorithmes d'apprentissage automatique

Les algorithmes décris dans cette section seront testés dans la partie pratique. Le choix de ces algorithmes a été fait selon leur interprétabilité et précision.

1.6.1 Arbres de décision

Un arbre de décision est un outil d'aide à la décision qui utilise un modèle arborescent de décisions et leurs conséquences possibles, y compris les résultats d'événements fortuits, les coûts des ressources et l'utilité. C'est une façon d'afficher un algorithme qui ne contient que des instructions de contrôle conditionnelles. L'arbre de décision peut être linéarisé en règles de décision, [75] où le résultat est le contenu du nœud feuille, et les conditions le long du chemin forment une conjonction dans la clause if. En général, les règles ont la forme : "si condition1 et condition2 et condition3 alors résultat". Les règles de décision peuvent être générées en construisant des règles d'association avec la variable cible à droite. Ils peuvent également désigner des relations temporelles ou causales. [49] De ce fait, les arbres de décision sont très interprétables – tant qu'ils sont courts. Le nombre de nœuds terminaux augmente rapidement avec la profondeur. Plus il y a de nœuds terminaux et plus l'arbre est profond, plus il devient difficile de comprendre les règles de décision d'un arbre.

1.6.2 K-Means

Le k-means clustering est une méthode qui vise à partitionner n observations en k clusters dans lesquels chaque observation appartient au cluster de moyenne la plus proche (centres de cluster ou centroïde de cluster), servant de prototype de la grappe. Il en résulte un partitionnement de l'espace de données en groupes de points de données. Le clustering k-means minimise les variances intra-cluster (distances euclidiennes au carré). Étant donné un ensemble d'observations $(x1, x2, ..., xn)$, où chaque observation est un vecteur réel d-dimensionnel, le clustering k-means vise à partitionner les n observations en $k(\leq n)$ ensembles $S = S_1, S_2, ..., S_k$ de manière à minimiser la somme des carrés intra-cluster (la variance). Formellement, l'objectif est de trouver :

$$arg_s min \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 = arg_S min \sum_{i=1}^{k} |S_i| Var S_i \tag{1.3}$$

Tel que μ_i est la moyenne des points de S_i. Cela équivaut à minimiser les écarts au carré deux à deux des points dans le même cluster :

$$arg_S min \sum_{i=1}^{k} \frac{1}{|S_i} \sum_{x,y \in S_i} ||x - y||^2 \tag{1.4}$$

L'équivalence peut être déduite de l'identité

$$|S_i| \sum_{x \in S_i} ||x - \mu_i||^2 = \sum_{x \neq y \in S_i} ||x - y||^2 \tag{1.5}$$

Puisque la variance totale est constante, cela équivaut à maximiser la somme des écarts au carré entre les points de différents clusters (somme des carrés inter-cluster).

1.6.3 KNN

L'algorithme des k plus proches voisins, également connu sous le nom de KNN, est un classificateur d'apprentissage supervisé non paramétrique et non linéaire, qui utilise la proximité pour effectuer des classifications ou des prédictions sur le regroupement d'un point de données individuel. Bien qu'il puisse être utilisé pour des problèmes de régression ou de classification, il est généralement utilisé comme algorithme de classification, en partant de l'hypothèse que des points similaires peuvent être trouvés les uns à côté des autres. Pour les problèmes de classification, une étiquette de classe est attribuée sur la base d'un vote majoritaire, c'est-à-dire l'étiquette la plus fréquemment représentée autour d'un point de données donné est utilisée.[11]

Métriques utilisées

Afin de déterminer quels points de données sont les plus proches d'un point de requête donné, la distance entre le point de requête et les autres points de données devra être calculée. Ces métriques de distance aident à former des limites de décision, qui partitionnent les points de requête en différentes régions. Dans ce qui suit nous citons quelques métriques.

La distance euclidienne , il s'agit de la mesure de distance la plus couramment utilisée, et elle est limitée aux vecteurs à valeurs réelles. En utilisant la formule ci-dessous, elle mesure une ligne droite entre le point de requête et l'autre point mesuré.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{1.6}$$

La distance de Manhattan , il s'agit également d'une autre mesure de distance populaire, qui mesure la valeur absolue entre deux points. Elle est également appelée distance en taxi ou distance d'un pâté de maisons, car elle est généralement visualisée avec une grille, illustrant comment on peut naviguer d'une adresse à une autre via les rues de la ville. Sa formule est de la forme suivante :

$$d(x, y) = \sum_{i=1}^{m} |x_i - y_i| \tag{1.7}$$

La distance de Minkowski , cette mesure de distance est la forme généralisée des métriques de distance euclidienne et Manhattan. Le paramètre, p, dans la formule ci-dessous, permet la création d'autres mesures de distance. La distance euclidienne est représentée par cette formule lorsque p est égal à deux, et la distance de Manhattan est notée avec p égal à un.

$$d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|)^{1/p} \tag{1.8}$$

La distance de Hamming , cette technique est généralement utilisée avec des vecteurs booléens ou de chaînes de caractères, identifiant les points où les vecteurs ne correspondent pas. En conséquence, elle a également été appelée la métrique de chevauchement. Ceci peut être représenté par la formule suivante :

$$D_H = \sum_{i=1}^{k} |x_i - x_i|, \;\; si\ x = y\ alors\ D = 0\ et\ si\ x \neq y\ alors\ D = 1 \tag{1.9}$$

1.6.4 Forêts d'arbres aléatoires

Les forêts aléatoires ou forêts de décision aléatoires sont une méthode d'apprentissage d'ensemble pour la classification, la régression et d'autres tâches qui fonctionnent en construisant une multitude d'arbres de décision au moment de la formation. Pour les tâches de classification, la sortie de la forêt aléatoire est la classe sélectionnée par la plupart des arbres. Pour les tâches de régression, la prédiction moyenne ou moyenne des arbres individuels est renvoyée.[40] [41] Les forêts de décision aléatoires corrigent l'habitude des arbres de décision de sur-ajuster à leur ensemble d'entraînement.[39] Cependant, les caractéristiques des données peuvent affecter leurs performances.[72] [56] Les forêts aléatoires sont fréquemment utilisées comme modèles de "boîte noire" dans les entreprises, car elles génèrent des prédictions raisonnables sur une large gamme de données tout en nécessitant peu de configuration.[53]

1.6.5 Machines à vecteurs de support

Les machines à vecteurs de support ou SVM, de l'anglais Support Vector Machines, sont une familles d'algorithmes de classification supervisés très efficaces dans le cas de problèmes linéaires et non linéaires [11]. Dans ce qui suit, nous allons définir les deux types de SVM. Les explications données sont issues du livre de (Bonaccorso Guiseppe, 2017) [11].

Machines à vecteurs de support linéaires

Étant donné un ensemble de points de deux classes dans un espace de dimension N, l'algorithme SVM génère un hyperplan de dimension N-1 pour séparer ces points en deux groupes. Dans le cas de plusieurs classes, une extension peut être considérée. Soit un ensemble de vecteurs de caractéristiques

$$X = \{\bar{x_1}, \bar{x_2}, ..., \bar{x_n}\}\ tel\ que\ \bar{x_i} \in \mathbb{R}^m \tag{1.10}$$

Pour simplifier, nous supposons qu'il s'agit d'une classification binaire. Dans tous les autres cas, il est possible d'utiliser automatiquement la stratégie du un contre tous définie dans la définition suivante

La stratégie du un contre tous (OvR en abrégé, également appelé One-vs-All ou OvA) est une méthode heuristique permettant d'utiliser des algorithmes de classification binaire pour la classification multi-classes. Cela implique de diviser l'ensemble

de données multi-classes en plusieurs problèmes de classification binaire. Un classificateur binaire est ensuite entraîné sur chaque problème de classification binaire et des prédictions sont faites à l'aide du modèle le plus fiable.

$$Y = \{y_1, y_2, ..., y_n\} \; tel \; que \; y_n \in \{-1, 1\} \tag{1.11}$$

L'objectif de l'algorithme est de retrouver l'équation du meilleure plan séparateur des deux classes. Celle-ci est de la forme

$$\bar{w}^T \bar{x} + b = 0 \; tel \; que \; \bar{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} et \; \bar{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \tag{1.12}$$

Une représentation bidimensionnelle de cet hyperplan est illustrée dans la figure suivante

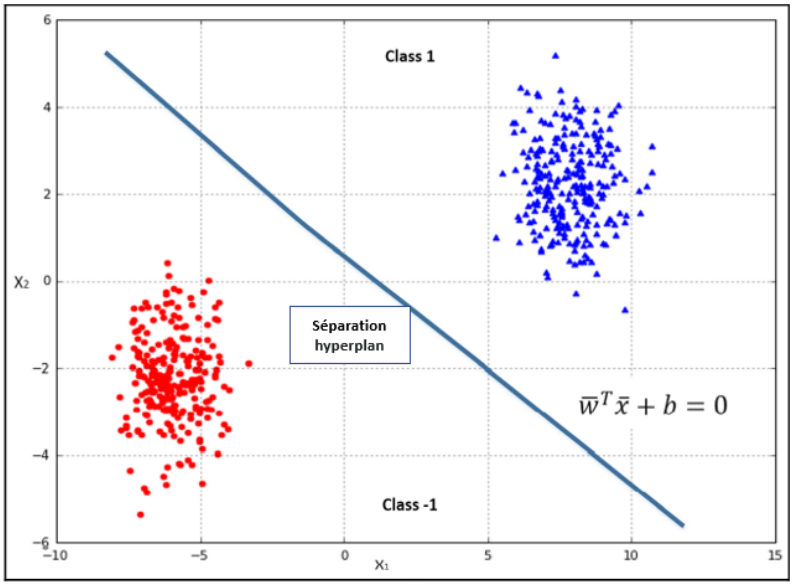


FIGURE 1.4 – Droite séparatrice de deux classes [11]

Le classificateur peut être écrit de la manière suivante

$$\tilde{y} = f(\bar{x}) = sgn(\bar{w}^T \bar{x} + b) \tag{1.13}$$

Machines à vecteurs de support non linéaires

Avant d'introduire les machines à vecteurs de support non linéaires il est important de définir les méthodes noyaux. Les svm non linéaires sont basées sur celles-ci.

Définition 1.6.1. *Les noyaux , ou kernels en anglais, sont des méthodes d'utilisation d'un classificateur linéaire pour résoudre un problème non linéaire, cela se fait en transformant une donnée linéairement inséparable en une donnée linéairement séparable.*

Les SVM adoptent également cette approche, même s'il y a maintenant un problème de complexité qu'il faut surmonter.

Chaque vecteur de caractéristiques est désormais filtré par une fonction non linéaire qui peut complètement remodeler le scénario. Cependant, l'introduction d'une telle fonction a augmenté la complexité de calcul d'une manière qui pourrait apparemment décourager cette approche. Du à la grande complexité de cette procédure, elle est inacceptable pour les gros problèmes. Pour régler le problème de complexité, il existe des fonctions particulières (appelées noyaux) qui ont la propriété suivante :

$$K(\bar{x}_i, \bar{x}_j) = \phi(\bar{x}_i)^T \phi(\bar{x}_j) \tag{1.14}$$

En d'autres termes, la valeur du noyau pour deux vecteurs de caractéristiques est le produit des deux vecteurs projetés. Avec cette astuce, la complexité de calcul reste quasiment la même, mais on peut bénéficier de la puissance des projections non linéaires même dans un très grand nombre de dimensions. Dans ce qui suit, nous citons quelques types de noyaux.

Fonction de base radiale , celle-ci est basée sur la fonction suivante

$$K(\bar{x}_i, \bar{x}_j) = e^{-\gamma|\bar{x}_i - \bar{x}_j|^2} \tag{1.15}$$

Le paramètre gamma détermine l'amplitude de la fonction, qui n'est pas influencée par la direction mais uniquement par la distance.

Noyau polynomial , qui est basé sur la fonction

$$K(\bar{x}_i, \bar{x}_j) = (\gamma \bar{x}_i^T . \bar{x}_j + r)^c \tag{1.16}$$

Cette fonction peut facilement étendre la dimensionnalité avec un grand nombre de variables de support et surmonter des problèmes non linéaires ; cependant, les exigences en termes de ressources sont normalement plus élevées. Considérant qu'une fonction non linéaire peut souvent être assez bien approximée pour une zone délimitée (en adoptant des polynômes), il n'est pas surprenant que de nombreux problèmes complexes deviennent facilement solubles à l'aide de ce noyau.

Fonction sigmoïde , qui est représentée par la fonction suivante

$$K(\bar{x}_i, \bar{x}_j) = \frac{1 - e^{-2(\gamma \bar{x}_i^T \bar{x}_j + r)}}{1 + e^{-2(\gamma \bar{x}_i^T \bar{x}_j + r)}} \tag{1.17}$$

1.6.6 Réseaux de neurones

Les neurones formels Un "neurone formel" (ou simplement "neurone") est un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de logiciel, en d'autres termes un neurone est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés

coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie" [28].
La figure suivante 1.5 présente un neurone formel :

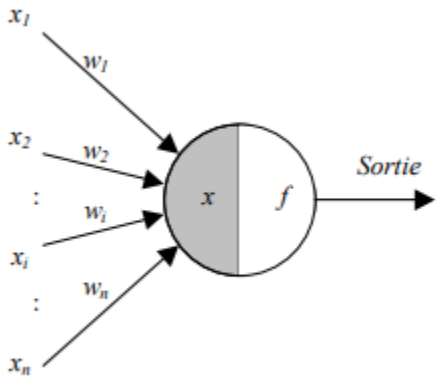


FIGURE 1.5 – Représentation d'un neurone
[8]

x_i : le paramètre d'entrée ;

w_i : le poids qui relie le paramètre d'entrée au neurone

x : l'état d'activation du neurone

f :la fonction d'activation du neurone qui est une fonction de transfert qui relie la sommation pondéré au signal de sortie [65]. Les fonctions les plus connues sont la fonction signe, la fonction linéaire saturée et la fonction sigmoïde.

La sortie du neurone est donnée par :

$$Sortie = f(x) = \sum_{i=1}^{n} x_i w_i \tag{1.18}$$

Le réseau de neurones fait partie des Réseaux Adaptatifs Non-linéaires, cela signifie que ses agents (neurones) s'organisent et modifient leurs liens mutuels lors d'une procédure fondamentale qu'est l'apprentissage. Pour une tâche précise, l'apprentissage du réseau de neurones consiste donc à adapter les différents poids [8].

1.6.7 Réseaux bayésiens

Les modèles graphiques probabilistes, et plus précisément les réseaux bayésiens, initiés par Judea Pearl dans les années 1980, se sont révélés des outils très pratiques pour la représentation de connaissances incertaines, et le raisonnement à partir d'informations incomplètes [54].

un réseau bayésien (RB) est un modèle graphique et probabiliste représentant un ensemble de variables de probabilités conditionnelles et aléatoires relié entre elles, constituant un graphe orienté acyclique [58].Les BN ont également une architecture modulaire qui facilite le développement itératif de modèles. Pour qu'un modèle soit utile dans la production et le partage de connaissances ou dans l'aide à la prise de décision, il doit être construit en utilisant de bonnes de modélisation [20].

1.7 Pipeline d'apprentissage automatique

Un pipeline d'apprentissage automatique est une série d'éléments de traitement, qui comprennent des processus, des threads, des routines et des fonctions, disposés sous la forme d'un organigramme pour transformer les données d'une représentation à une autre. L'objectif de la création d'un pipeline d'apprentissage automatique est d'améliorer la modularité tout en se concentrant sur la répétabilité et la flexibilité.

Un projet de machine learning prêt pour la production ou dans le domaine industriel nécessite un pipeline de machine learning soigneusement architecturé. Le modèle lui-même n'est qu'un des nombreux composants du pipeline de bout en bout. Certains des autres composants du flux de travail d'apprentissage automatique comprennent la collecte de données, la vérification et le prétraitement des données, l'extraction de caractéristiques, la sélection de modèles, la formation et la validation, la prédiction, l'évaluation et le déploiement. Nous définissons ces concepts en détails dans le chapitre suivant.

1.8 Conclusion

Dans ce chapitre, nous avons donné un aperçu des principales idées qui figurent dans le domaine de l'apprentissage artificiel. Actuellement, des spécialistes et des acteurs d'apprentissage automatique ouvrent à créer un processus qui assurerait l'industrialisation du déploiement de l'apprentissage automatique dans les entreprises. Parmi ce qui a été proposé, le processus MLOps (Machine Learning Operations). Pour le reste de ce mémoire noust discutons ce concept et présentons chaque étape et ses constituants granulaires

Chapitre 2

Apprentissage automatique en production

2.1 Introduction

Beaucoup d'entreprises rencontrent des difficultés lorsqu'elles veulent transférer des modèles de Machine Learning de laboratoire vers des environnements de production. Afin de résoudre ce problème le Mlops est apparu.

Dans cette section, nous commençons par montrer quelques défis de l'utilisation continue de modèles ML tel que la dérive conceptuelle (Concept drift) qui peut considérablement nuire à l'exactitude des prédictions. Ensuite, nous parlerons de DevOps, opérations sur lesquelles les MLOps sont basées. Nous continuons avec la définition des opérations ML (MLOps). Par la suite, nous mettons en évidence les composants des pipelines ML et leur maniabilité pour la meilleure gestion du cycle de vie d'un système d'apprentissage automatique. Et enfin, nous soulignons les différentes étapes du cycle de vie et les avantages du MLOps.

2.2 Défis de l'utilisation continue de l'apprentissage automatique

Le développement de modèles d'apprentissage automatique efficaces et productifs comprend plusieurs difficultés, ces difficultés proviennent

Il est difficile de suivre les expériences menées par les datascientists ; savoir quelles données, codes et paramètres ont permis d'obtenir un résultat particulier. Il est difficile de reproduire le code, par exemple si on souhaite exécuter le même code sur une autre plateforme. Il n'existe pas de méthode standard pour empaqueter et déployer des modèles d'apprentissage automatique. Il n'y a pas de magasin central pour gérer les modèles (leurs versions et transitions d'étape). Une équipe de science des données crée de nombreux modèles. En l'absence d'un lieu central pour collaborer et gérer le cycle de vie des modèles, les équipes de science des données sont confrontées

à des défis dans la façon dont elles gèrent les étapes des modèles : du développement au déploiement, et enfin, à l'archivage ou à la production, avec les versions, les annotations et l'historique respectifs. La dégradation des performances d'un modèle au fil du temps est un sérieux problème qui menace la qualité d'un produit ML. Une des principales raisons pour cette dégradation est la dérive de modèle, ou Model Drift. Dans ce qui suit nous définissons les 3 types de dérive de modèle qui sont : la dérive de données, la dérive conceptuelle et l'Upstream Data Changes.

2.2.1 Dérive de données

La dérive des données, ou Data Drift, est définie comme une variation des données de production par rapport aux données qui ont été utilisées pour tester et valider le modèle avant de le déployer en production. De nombreux facteurs peuvent entraîner une dérive des données, l'un des facteurs clés étant la dimension temporelle. Dans le diagramme de la figure 2.1 qui montre les étapes de haut niveau dans le développement du modèle d'apprentissage automatique, il est évident qu'il existe un écart important entre le moment où les données sont recueillies et le moment de déploiement. Cet écart peut aller de quelques semaines à plusieurs mois ou années, selon la complexité du problème. Plusieurs autres facteurs peuvent également entraîner une dérive comme des erreurs dans la collecte de données, la saisonnalité, par exemple si les données sont collectées avant le covid et que le modèle est déployé après le covid [55].

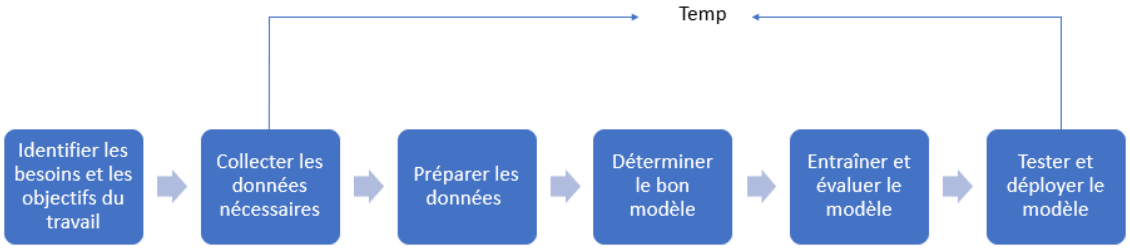


FIGURE 2.1 – Étapes de développement d'un modèle d'apprentissage automatique [55]

Les dérives de données peuvent être identifiées à l'aide de méthodes d'analyse séquentielle, de méthodes basées sur des modèles et de méthodes basées sur la distribution temporelle. Les méthodes d'analyse séquentielle telles que DDM (méthode de détection de dérive)/EDDM (première DDM) s'appuient sur le taux d'erreur pour

identifier la détection de dérive, une méthode basée sur un modèle utilise un modèle personnalisé pour identifier la dérive et les méthodes basées sur la distribution temporelle utilisent la distance statistique comme méthode de calcul pour calculer la dérive entre les distributions de probabilité [55].

2.2.2 Dérive de concept

La dérive des concepts se produit lorsque la cible prédite par le modèle ou ses propriétés statistiques changent au fil du temps. Pendant la formation, le modèle apprend une fonction qui met en correspondance la variable cible, mais au fil du temps, il la désapprend ou est incapable d'utiliser ces modèles dans un nouvel environnement [26]. Par exemple, comme la définition du spam a évolué, les modèles doivent procéder à des ajustements. La dérive de concept se produit également de manière saisonnière, soudaine ou progressive. Par exemple, le comportement des consommateurs après la pandémie de Covid est une dérive soudaine alors que les changements dans les tendances de la mode sont graduels.

La dérive des concepts peut être mesurée en surveillant continuellement les données de formation et en identifiant les changements dans les relations entre les ensembles de données. Parmis les algorithmes populaires de détection de dérive, le test de Kolmogorov-Smirnov [60] qui est un test statistique qui nous permet de vérifier si deux échantillons proviennent de la même distribution [26].

2.2.3 Upstream Data Changes

Les modifications de données en amont, ou Upstream Data Changes, font référence aux modifications de données opérationnelles dans le pipeline de données. Par exemple, lorsqu'une caractéristique n'est plus générée, ce qui entraîne des valeurs manquantes. Un autre exemple est un changement de mesure (par exemple, de miles à kilomètres) [26].

Il est important de créer un processus reproductible pour l'identification de ces dérives, définir des seuils sur le pourcentage de dérive, configurer des alertes proactives afin que les mesures appropriées soient prises. Les frameworks MLOps fournissent de tels outils.

2.3 DevOps

DevOps (Developpement Operations) est un ensemble de pratiques dans le monde du développement logiciel traditionnel qui permet de réduire le temps entre le développement et l'exploitation de logiciel sans nuire à sa qualité [35]. Il s'appuie sur l'automatisation, les outils et les flux de travail pour éliminer la complexité accidentelle et permettre aux développeurs de se concentrer sur des problèmes plus critiques. La majorité des publications précisent que DevOps est un terme utilisé pour souligner la collaboration entre le développement de logiciels et les opérations [59]. Cette

méthodologie est composée de deux pratiques essentielles : l'intégration continue (CI) et la livraison continue (CD) (Figure 2.2).

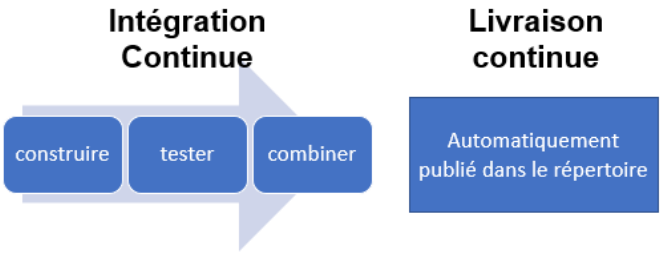


FIGURE 2.2 – Relation entre CD et CI [77]

Intégration Continue (CI) : le CI est une pratique de développement de logiciel dont les membres d'une équipe intègrent et fusionnent le travail de développement, par exemple le code,fréquemment [82],.

Livraison continue (CD) : Le CD est un processus qui déploie automatiquement toutes les modifications de code dans un environnement de test et/ou de production après l'étape de construction [82] .

2.4 MLOps

Machine learning opérationnelle -MLOps- est une approche systématique [76] représentant l'ensemble de pratiques combinant l'apprentissage automatique, le pratiques DevOps et le Data Engineering qui qui vise à déployer et raccourcir le temps de développement et de publication des logiciels et modèles ML/AI en production de manière fiable et efficace.

La méthode MLOps est une solution aux défis posés par les méthodes de développement logiciel traditionnelles en ce qui concerne les applications ML. En l'utilisant, les données et le code progressent dans le temps dans une direction avec un objectif de construction et de maintien d'un système ML robuste et évolutif [76]. La figure suivante 2.3 montre la façon dont les données et le code progressent ensemble.

FIGURE 2.3 – MLOps – data and code progressing together [76]

En plus des deux principales pratiques de DevOps : l'intégration continue (CI) et la Livraison continue (CD),Le MLops introduit une nouvelle pratique celui de l'entraînement continu (CT) dont le but est de résoudre le problème de changement de données en recyclant automatiquement le modèle si nécessaire.Par rapport a Devops, les machines learning opérationnelles sont beaucoup plus complexes et incorporent des procédures supplémentaire impliquant des données et des modèles [88].

2.5 Cycle de vie

L'opérationnalisation du ML comprend quatre phases principales : l'analyse commerciale, le développement du modèle, la vérification du modèle et la gestion des opérations sur le modèle (par exemple, surveillance du modèle en production)[47] [5].

Dans cette section, nous détaillons ces quatre phases en montrant les bonnes pratiques servant à leur exécution. La bonne exécution de ces composants est primordiale pour bien mener l'évolution des applications ML tout en réduisant le coût de production. Pour parvenir à une compréhension commune de l'évolution des applications ML dans l'équipe de développement, les pratiques MLOps aident à décomposer chaque composant dans ses activités respectives et discuter des méthodologies, des approches et des opportunités présentées par chacun. La figure 2.4 montre les principaux composants de chaque étape du cycle de vie.

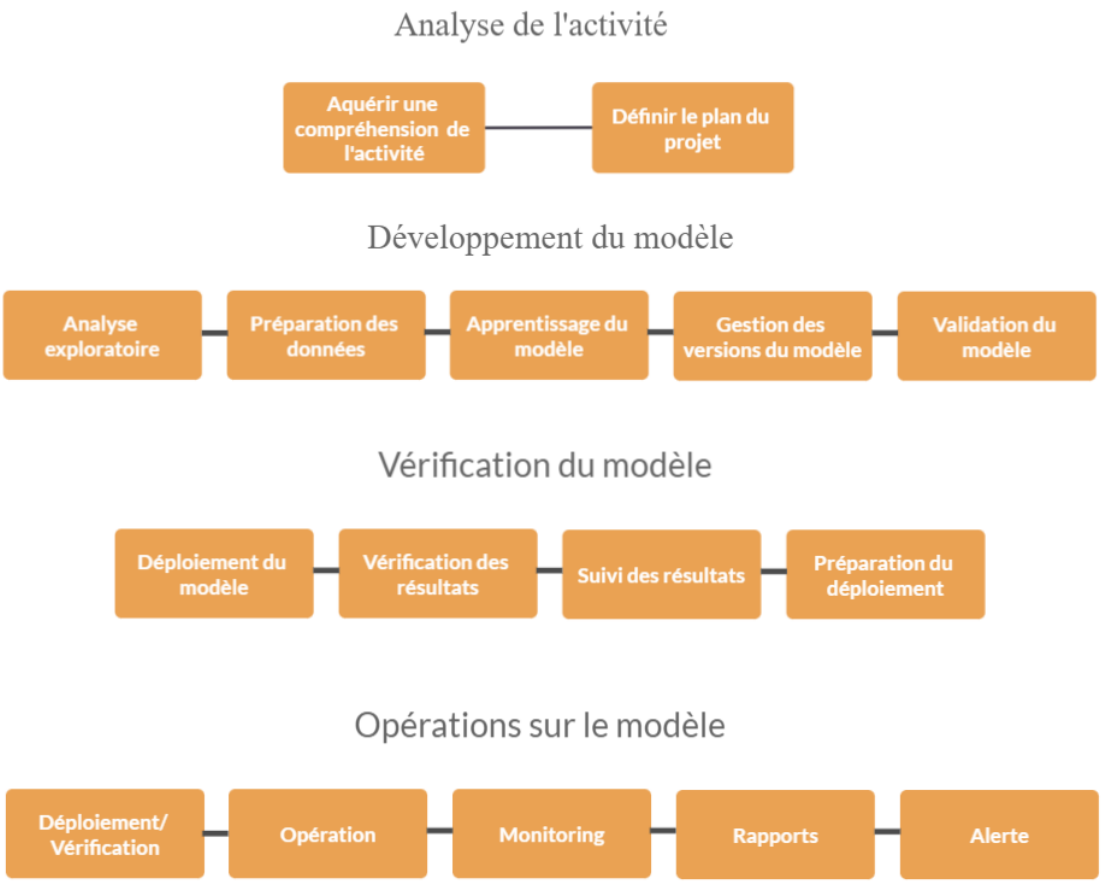


FIGURE 2.4 – Cycle de vie du développement d'un système ML [47]

Il est important de souligner que la séquence apparente dans cette description n'est pas nécessairement la norme dans un scénario réel. Il est parfaitement possible que ces étapes fonctionnent en parallèle et s'informent mutuellement via des boucles de rétroaction.

2.5.1 Analyse de l'activité (Business Analysis)

Les unités commerciales investissent de plus en plus dans les techniques de ML et les identifient comme des domaines de grande expansion [32]. Ces unités commerciales cherchent à exploiter les données et à découvrir des modèles pour améliorer l'efficacité et permettre une prise de décision plus éclairée. Un projet d'apprentissage automatique réussi, analyse les objectifs commerciaux pour révéler les facteurs importants qui influencent les livrables des résultats d'apprentissage automatique. Cette étape est essentielle pour s'assurer que les projets de Machine Learning retournent les bons résultats, ceux qui répondent directement au problème métier qu'on a initialement prévu de résoudre. Le travail doit être fait en étroite collaboration avec des

experts en la matière et des utilisateurs finaux pour mieux comprendre comment ils fonctionnent dans l'environnement actuel. Un plan de projet doit ensuite être élaboré pour définir comment le modèle ML répond aux objectifs commerciaux. Il est important que toutes les parties prenantes s'accordent sur des critères qui précisent comment les résultats doivent être mesurés et ce qui constitue un succès commercial. Les hypothèses doivent être documentées pour aider à guider le processus de développement du modèle ultérieur dans la phase suivante. Toutes les étapes doivent être définies, y compris les détails du projet sur la portée, la durée, les ressources requises, les intrants, les extrants et les dépendances [47].

2.5.2 Développement du modèle

Le développement du modèle pour une utilisation professionnelle nécessite des étapes incrémentales. Étant donné que ce processus de développement implique de délimiter le problème, mener des expériences, examiner les résultats et affiner l'approche, il est difficile de le décrire de manière simple. Bien que chacune des cinq étapes qui seront définies dans la section suivante puisse se produire à différents moments, leur contribution au processus global reste consistante[47].

Analyse exploratoire des données

Une fois les objectifs du projet définis et le problème commercial compris, une analyse exploratoire est nécessaire pour identifier et collecter toutes les données pertinentes et évaluer leur capacité à répondre à l'objectif commercial visé. Ce processus est conçu pour créer une vue complète des données disponibles et utiliser des techniques analytiques pour explorer les relations importantes dans les données. L'analyse exploratoire sert également à identifier les lacunes dans les données qui peuvent empêcher le modèle de répondre à l'objectif commercial prévu. L'exploration de données analyse les données dont nous disposons et nous informe également sur le type de données nécessaires pour développer un modèle utile. Il existe 4 principaux types d'analyse exploratoire de données [31] [78] :

Univariée non graphique : Les données analysées ne comportent qu'une seule variable unique qui ne traite pas des causes ou des relations, l'objectif étant de chercher des modèles (patterns) existant au sein de ces données.

Univariée graphique : Pour avoir une image plus complète des données des méthodes graphiques sont nécessaires. Les types courants des graphiques univariés incluent :

- **Le diagramme tige et feuille** , qui est une technique utilisée pour classer des variables discrètes ou continues. Il sert à organiser des données d'une série statistique au fur et à mesure de leur collecte [3]. Dans la première colonne, la " tige ", on écrit les nombres de milliers, de centaines, ou de dizaines dont peut être constituée chacune des valeurs, et sur chacune

des lignes, les " feuilles ", on écrit les chiffres des unités de chacune des valeurs.

Exemple 2.5.1. *Données :* $18, 19, 20, 48, 49, 47, 24, 27, 16$
Diagramme tige-feuilles :
1 | 8 9 6
2 | 0 4 7
4 | 8 9 7*Pré*

- **L'histogramme**
- **Le diagramme à surface** , qui représente graphiquement le résumé en cinq chiffres du minimum, du premier quartile, de la médiane, du troisième quartile et du maximum, comme le montre la figure 2.5

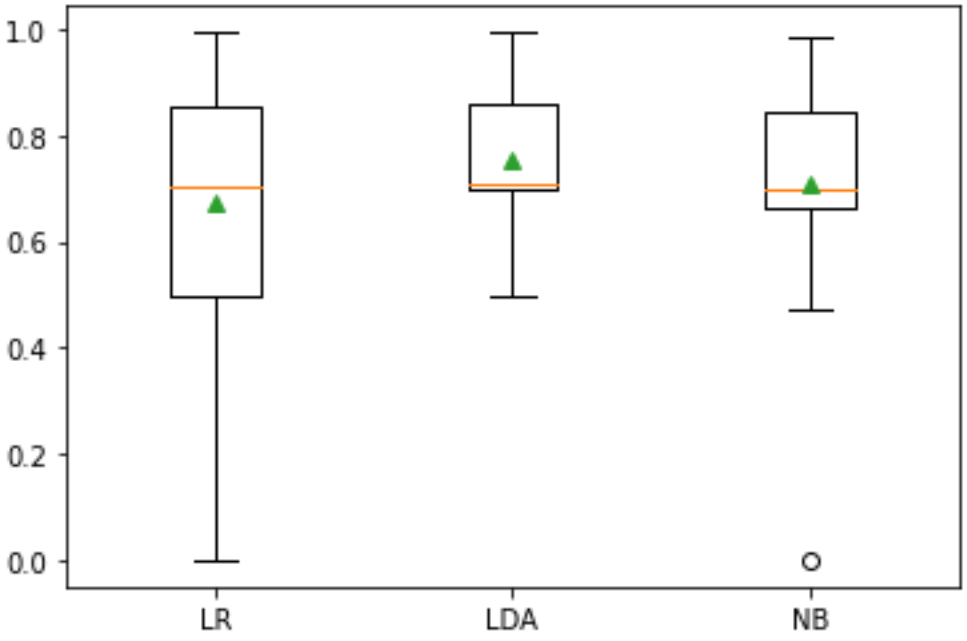


FIGURE 2.5 – Diagrammes à surface montrant les performances de 3 modèles de classification soit la régression logistique, l'analyse discriminante linéaire et les naïves Bayes sur l'ensemble de données "Oil Spill Dataset" avec une méthode de validation croisée à k = 10 découpes (Stratified k-fold validation).

Multivariée non graphique , qui sert à montrer la relation entre deux variables ou plus, par le biais de tableaux croisés ou de statistiques.

Multivariée graphique , qui utilise des graphiques pour afficher les relations entre deux ensembles de données ou plus, comme le diagramme à barres groupées. Il existe d'autres types de graphiques multivariés comme les nuages de points, les diagrammes d'exécution qui incluent une variable temporelle, les

graphiques à bulles. [97] ont utilisé une technique de réduction de caractéristiques sur des ensembles de données sur le trafic réseau, afin de pouvoir représenter graphiquement ces données dans un espace 3D et les visualiser. La visualisation des différents comportements du trafic réseau (les différents types d'attaques) a permis de comprendre la provenance des différentes erreurs de classification.

Préparation des données

Cette étape consiste à explorer au mieux la structure de données du problème afin de l'exposer aux algorithmes d'apprentissage. Dans cette étape, les problèmes de manque d'informations, données en double / invalides, données déséquilibrées (imbalanced data) sont traitées. Cette étape comprend également des techniques d'aide à la préparation des données, notamment : la suppression des données en double/invalides de l'ensemble de données ; remédier au déséquilibre des classes (imbalanced classification lorsque le nombre total d'une classe de données "positif" est bien inférieur au nombre total d'une autre classe de données "négatif") [13] ; enrichissement à l'aide de sources de données supplémentaires ; et la sélection des caractéristiques de données et l'exécution de l'ingénierie des caractéristiques (y compris la combinaison de plusieurs fonctionnalités pour des avantages de formation supplémentaires). Ce processus est conçu pour aider à déterminer les variables d'entrée requises lors de l'élaboration d'un modèle prédicatif. Les experts en la matière sont inestimables au cours de cette phase de développement du modèle [14].

Apprentissage du modèle

L'apprentissage du modèle commence une fois que les données ont été préparées. Dans cette étape, les données d'apprentissage et de validation sont utilisées pour calculer les paramètres du modèle qui optimisent les performances. Avant d'initialiser le processus d'apprentissage du modèle, il est nécessaire de définir le modèle à entraîner et ses configurations correspondantes. Un processus général de formation de modèle peut généralement être utilisé sur différents algorithmes. Pour la plupart des types de modèles ML, la procédure d'apprentissage générale commence par une initialisation aléatoire du modèle et de ses paramètres associés. Ce modèle est ensuite formé à l'aide de données pour l'affiner, en générant et en comparant des prédictions pour chaque entrée dans l'ensemble de données de formation. À l'aide de ces prédictions, la formation du modèle calcule l'erreur et d'autres mesures de performance entre la prédiction du modèle et la vrai valeur pour chaque entrée (par exemple, cas, enregistrement, image). L'erreur calculée, souvent appelée «perte», est ensuite utilisée pour ajuster les paramètres du modèle afin d'optimiser les mesures souhaitées. Le script d'apprentissage continuera à parcourir les étapes mentionnées jusqu'à ce qu'il atteigne les performances souhaitées ou jusqu'à ce que le nombre maximal d'étapes d'entraînement ait été atteint.

Gestion des versions du modèles

L'étape suivante, la gestion des versions du modèle, est cruciale pour s'assurer qu'un flux de travail est explicable et peut être reproduit. Dans cette étape, les modèles sont versionnés en fonction des données utilisées pour l'apprentissage, des spécifications sélectionnées (par exemple, le taux d'apprentissage, la taille du lot, le nombre d'époques) et la sortie résultante. Un identifiant doit être attribué à chaque modèle pour fournir un contexte [12]. La gestion des versions permet à l'équipe de développement de suivre la lignée et le pedigree du modèle, ce qui améliorera les résultats au fur et à mesure que le projet progresse.

Validation du modèle

La dernière étape du développement, la validation du modèle, consiste à vérifier la formation d'un modèle en utilisant un ensemble de données de validation pour déterminer si des ajustements de performances supplémentaires sont nécessaires. Il est important que cet ensemble de données de validation n'inclue aucune donnée utilisée dans la phase d'apprentissage du modèle pour garantir que les métriques de validation soient totalement indépendantes de la procédure d'entraînement [36]. Lors de la phase de validation, le modèle créé est utilisé pour faire des prédictions pour chaque entrée de l'ensemble de données de test. L'erreur de prédiction et d'autres mesures de performances du modèle sont calculées pour garantir que les performances observées du modèle répondent aux exigences définies par l'utilisateur avant de déployer le modèle.

2.5.3 Vérification du modèle

La vérification d'un modèle ML entraîné est une étape cruciale, mais sous-estimée, sur la voie de l'opérationnalisation de l'apprentissage automatique. Une vérification rigoureuse des modèles permet à l'équipe locale de science des données (DataScience) de découvrir des bugs coûteux et de les corriger tout en restant dans un environnement à faible risque. Une fois la vérification exécutée localement, elle est ensuite répliquée dans l'environnement de production. Le déploiement de modèles fonctionnant correctement renforcera la confiance organisationnelle, non seulement dans le système déployé, mais dans les projets d'IA dans leur ensemble [47].

Déploiement du modèle

Les modèles entraînés qui répondent aux exigences de performances initiales sont déployés pour être testés dans un environnement de type production pour une étape de test et d'évaluation rigoureuse. Cette étape consiste à effectuer des tests de calcul et de performance pour s'assurer que le modèle répond aux exigences. Dans un souci de cohérence, le déploiement d'un modèle dans l'environnement de test doit suivre les étapes suivies pour le déployer dans un environnement de production réel [68].

L'équipe de développement doit travailler en étroite collaboration avec des experts en la matière et des utilisateurs fonctionnels pour s'assurer que l'environnement de test reflète l'environnement de production réel [47].

Vérification des résultats

Comme pour les étapes de développement du modèle, il est important de vérifier les résultats. Cette étape garantit que les performances du modèle sont conformes aux attentes et n'ont pas subi de dégradation de performances. Il est important de noter que, sur la base de notre expérience, la plupart des problèmes de modèle de pré-production sont découverts au cours de cette étape. Comme indiqué précédemment, il est crucial que les données envoyées au modèle au cours de cette étape soient représentatives des données que le modèle analysera dans un environnement de production. Une partie du processus de vérification des résultats consiste à explorer la manière dont le modèle gère les données mal formées ou autrement incorrectes. Le modèle doit être capable de gérer des données incorrectes et de s'assurer que les données importantes ne sont pas perdues, qu'une notification d'erreur appropriée est envoyée à l'équipe de développement logiciel et que le modèle reste fonctionnel après avoir rencontré l'erreur.

Suivi des résultats

Le suivi des résultats est essentiel pour garantir que modèle est effectivement examiné et audité. La possibilité de stocker et d'examiner la sortie du modèle historique permet aux experts en la matière et aux utilisateurs fonctionnels d'effectuer des examens périodiques, en s'assurant que le modèle fait des prédictions acceptables et appropriées.

Préparation du déploiement

La dernière étape, la préparation du déploiement, prépare le modèle à déplacer vers l'environnement de production. Une fois le modèle déployé dans l'environnement de production, le modèle est considéré comme "actif" et sera disponible pour tous les utilisateurs finaux et systèmes autorisés. Les datascientits, les ingénieurs logiciels et tous les experts appropriés doivent être consultés pour s'assurer que le modèle est déployé d'une manière qui correspond aux attentes dans l'environnement de production. Le déploiement peut être défini comme un aspect de divers degrés d'automatisation qui dépend du cas d'utilisation, des performances du modèle et du niveau de confort de l'entreprise.

2.5.4 Opérations sur le modèle

L'objectif ultime du workflow d'opérationnalisation en ML est le déploiement réussi et le bon fonctionnement d'un modèle. Le modèle idéal fournit aux utilisateurs

finaux des prédictions lorsqu'elles sont demandées en temps opportun, effectue une surveillance automatique des performances et de l'utilisation du modèle et informe les utilisateurs de l'activité du modèle pertinent par le biais de mécanismes de rapport et d'alerte. En bref, un déploiement réussi signifie que l'application ML fonctionne comme prévu et alerte lorsque l'attention de l'ingénieur est requise [47].

Déploiement/ Vérification

La première étape de l'opérationnalisation d'un modèle consiste à déployer le modèle dans l'environnement de production et à vérifier que le modèle a été transféré avec succès et qu'aucun problème n'existe avec l'exécution de celui-ci dans l'environnement prévu [85]. La vérification du déploiement est généralement effectuée en surveillant étroitement le modèle immédiatement après le déploiement pour s'assurer qu'il reçoit, analyse et renvoie les données appropriées. Cette étape implique également de vérifier que le formatage des données d'entrée est correct et d'analyser la sortie pour s'assurer qu'aucune dégradation des performances du modèle ne s'est produite. Il est préférable d'effectuer ces tests sur l'environnement d'hébergement de production par l'équipe d'ingénierie logicielle immédiatement après le déploiement.

Exploitation

L'étape d'opération consiste à exécuter le modèle lorsque cela est nécessaire. Au cours de cette étape, l'équipe d'ingénierie logicielle doit s'assurer que tous les aspects du modèle déployé fonctionnent comme prévu. Tous les problèmes du système doivent être résolus en fonction de leur ordre d'importance pour garantir que le modèle continue de fonctionner comme prévu. Ce processus nécessite généralement l'utilisation de tests automatisés réguliers du système, la validation de la réactivité du modèle, que les modèles soient recyclés si nécessaire et la vérification que les modèles résistent à l'ingestion de données mal formées ou incorrectes. Ces événements entraînent des alertes système pour demander une inspection humaine si nécessaire et un système de demande d'assistance qui permet aux utilisateurs finaux de soumettre des tickets d'assistance. Le bon fonctionnement d'un modèle déployé nécessite des capacités de surveillance robustes. Un système de surveillance déployé doit être capable de suivre les entrées et les sorties du modèle, de capturer tout message d'avertissement ou d'erreur produit par l'environnement logiciel, de calculer l'activité anormale du modèle (par exemple, l'analyse de détection de dérive, la détection d'attaques d'IA adverses) et de transmettre toutes les informations pertinentes au interfaces de rapport et d'alerte du système.

Monitoring

La surveillance efficace d'un système déployé nécessite plusieurs systèmes de composants pour fonctionner ensemble de manière transparente. Pour ce faire, mieux vaut automatiser autant que possible le système. Compte tenu de la vitesse et de l'échelle

auxquelles de nombreux systèmes d'IA fonctionnent, il ne suffit pas de se fier uniquement à la surveillance et à l'analyse humaines pour capturer tous les détails nécessaires en temps opportun. En tant que telles, des tâches de surveillance automatisées sont requises pour l'activité de surveillance souhaitée et configurées pour s'exécuter selon les besoins. Tous les résultats de ces tâches de surveillance automatisées sont signalés et envoyés sous forme d'alerte, lorsque cela est jugé nécessaire.

Rapports

Les rapports permettent une inspection humaine du modèle déployé. La vitesse et l'échelle auxquelles de nombreux systèmes d'IA fonctionnent peuvent rendre impossible pour un humain d'examiner les performances du modèle en temps réel. Cependant, des rapports automatisés à intervalles réguliers permettront aux utilisateurs de vérifier que le système fonctionne comme prévu. Il est également possible que ces rapports soient utilisés par d'autres systèmes automatisés pour surveiller le modèle déployé. Ces rapports sont examinés à l'aide de formats visuellement attrayants tels qu'un tableau de bord, une feuille de calcul ou un document pour fournir des informations aux utilisateurs.

Alerte

Les modules d'alerte sont chargés de créer des notifications en temps réel sur l'état du système déployé. Ces notifications sont destinées soit à signaler aux personnes un examen immédiat du système, soit à créer des micro-rapports à inclure dans les journaux système automatisés. Les événements qui nécessitent des notifications d'alerte sont initialement signalés par le module de surveillance du modèle. Ces événements signalés sont reçus par le module d'alerte, où les alertes sont rédigées sur la base de modèles de notification pré-écrits. Ces notifications sont ensuite envoyées via un service de messagerie au destinataire. En fonction des besoins, les notifications peuvent être envoyées sous forme de SMS, d'e-mails, de messages Slack, de notifications push d'application ou d'appels téléphoniques automatisés.

2.6 Pipelines

Dans toute équipe d'ingénierie logicielle, un pipeline est un ensemble de processus automatisés qui permettent aux développeurs et aux professionnels DevOps de compiler, créer et déployer de manière fiable et efficace leur code sur leurs plates-formes de calcul de production. Un autre type de pipeline ML est l'art de diviser ses flux de travail d'apprentissage automatique en parties indépendantes, réutilisables et modulaires qui peuvent ensuite être regroupées pour créer des modèles. Ce type de pipeline ML rend la création de modèles plus efficace et simplifiée, en supprimant les tâches redondantes. Cela va de pair avec la récente poussée des architectures de

microservices [18], partant de l'idée principale selon laquelle en divisant son application en parties de base et cloisonnées, on peut créer des logiciels plus puissants au fil du temps. Les systèmes d'exploitation comme Linux et Unix sont également fondés sur ce principe.

Exemple 2.6.1. *Les fonctions de base telles que "grep" et "cat" peuvent créer des fonctions impressionnantes lorsqu'elles sont regroupées. "grep" recherche les occurrences de chaînes correspondant à une expression régulière dans certains textes et imprime les lignes contenant l'une de ces occurrences. "cat", d'autre part, imprime toutes les lignes à partir des sources d'entrée spécifiées dans la commande [25].*

Dans ce qui suit, nous présentons quelques concepts clés qui seront utilisés dans notre projet.

2.6.1 Les Artefacts

Artefacts est un terme ML courant utilisé pour décrire la sortie créée par le processus de formation de modèle. La sortie peut être un modèle entièrement entraîné, un point de contrôle de modèle (pour reprendre la formation plus tard) ou simplement un fichier créé pendant le processus de formation, les artefacts du modèle sont les poids formés stockés dans un format binaire [92].

2.6.2 Les processus ETL (Extract, Transform, Load)

ETL est un processus qui extrait, transforme et charge des données provenant de plusieurs sources vers un entrepôt de données (Data Warehouse) [34] ou un autre référentiel de données, comme les DataLakes unifiés [33].

2.7 Avantages du MLOps

Les opérations d'apprentissage automatique fournissent une technologie qui aide à déployer, surveiller et gérer la production d'un certain modèle ML. Alors que l'emploi MLOps aide à faire évoluer les opérations ML en automatisant, validant et testant les processus,afin de développer un processus reproductible. Dans ce qui suit, nous donnons les principaux avantages de MLops :

Collaboration : Le MLops donne l'opportunité aux équipes qui mettent en oeuvre un projet ML, telle que les professionnels de l'analyse, les ingénieurs informatiques et les équipes de traitement, de travailler d'une manière collaborative comme un seul groupe non corrélé et isolé avec diverses compétences et expertises.

Innovation rapide : Grâce à la gestion robuste du cycle de vie de MLops les solutions ML sont développées plus rapidement et plus efficacement.

Déploiement simple et rapide : En utilisant une méthodologie DevOps rationalisée, MLOps simplifie le déploiement d'un algorithme d'apprentissage automatique. Il facilite l'intégration, le déploiement et la diffusion continue des modèles [10].

Gestion efficace du cycle de vie : Le MLops gère d'une manière efficace de l'ensemble du cycle de vie des modelés ML,ce qui permet, non seulement la planification, l'automatisation et la gestion efficace des workflows,mais aussi l'utilisation des CI / CD pour simplifier le recyclage et intégrer facilement l'apprentissage automatique dans les processus de publication existants.

2.8 Conclusion

Comme indiqué dans ce chapitre, la conception, la formation, les tests, le déploiement et la surveillance des modèles basés sur l'apprentissage automatique comportent de nombreuses étapes. Dans de nombreux cas, ces étapes sont exécutées manuellement. Cependant, cela réduit considérablement la capacité de mettre à l'échelle le processus de développement du modèle. Les MLOps (Machine Learning Operations) fournissent une solution à ce problème grâce à l'intégration et à l'automatisation pour permettre une efficacité accrue dans le développement et la formation des modèles ML. Grâce à l'intégration de ces processus à l'aide de solutions open source et commerciales en pleine expansion, MLOps permettra à une organisation d'adapter ses pipelines de développement ML pour répondre à la demande future et fonctionner au niveau de l'entreprise.

Chapitre 3

Présentation de l'étude

3.1 Introduction

Dans ce chapitre, nous présentons notre étude. Pour ce faire, nous commençons par introduire l'industrie du pétrole et du gaz. Ensuite, l'organisme Sonatrach qui nous a accueilli pour notre stage. Par la suite, nous mettons en évidence notre problématique qui est la détection des défaillances des oléoducs. Pour bien situer notre sujet d'étude, nous poursuivons par l'expression du contexte de notre étude. Et enfin, nous donnons un état de l'art sur les méthodes d'apprentissage automatique pour la détection des défaillances.

3.2 Présentation de l'industrie du pétrole et du gaz

Depuis la fin de la Seconde Guerre mondiale et aujourd'hui plus que jamais, Le pétrole et le gaz naturel sont des industries majeures sur le marché de l'énergie générant environ 5 000 milliards de dollars de revenus mondiaux en 2022 [43], et jouent un rôle influent dans l'économie mondiale en tant que principales sources de carburant dans le monde.

Dans cette section, nous décrivons l'industrie pétrolière et gazière. Notre objectif est de fournir une source d'informations pour aider les lecteurs à mieux comprendre le sujet de notre projet qui est la prédiction des défaillances des oléoducs. Nous donnons ainsi un aperçu générale de l'industrie en commençant par un bref historique qui est suivie d'une définition des principales activités de celle-ci.

3.2.1 Les hydrocarbures

Le pétrole et le gaz, autrement dit les hydrocarbures, sont des substances qui se forment à l'issue de la décomposition des restes végétaux et animales ayant été soumis aux facteurs environnementaux sous terre pour une longue durée. Ces combustibles fossiles, résultat de milliers d'années de décomposition, ont une composition chimique constituée uniquement d'atomes de carbone et d'hydrogène [45]. Les hydrocarbures

sont généralement sous la forme suivante : C_nH_m, $netm$ étant deux nombres entiers naturels non nuls [93].

3.2.2 Activités

L'exploitation du pétrole se subdivise schématiquement en deux étapes : l'amont et l'aval [22] :

Amont : Production et exploration

Le secteur pétrolier amont couvre l'exploration et la production d'hydrocarbures, ainsi que les services associés à ces activités. La phase d'exploration comprend la recherche de champs potentiels, tandis que la phase de production et plus précisément d'extraction du pétrole implique l'exploitation ultérieure des puits dans le but de produire du pétrole brut et du gaz naturel en utilisant des techniques complexes comme le maillage du réservoir par des multiples puits, le maintien de la pression du réservoir par injection d'eau et/ou de gaz, la séparation pétrole/gaz en surface et l'expédition vers les marchés [37]. Jusque là 'exploration et la production a été en grande partie terrestre pour cause de facilité d'accès. Toutefois, ces derniers temps, le forage en mer (offshore) est de plus en plus courant et progresse vers des méthode plus avancées sur des eaux plus profondes [22].

Aval : Raffinage et Marketing

Plus une société pétrolière et gazière est proche de l'approvisionnement des consommateurs en produits pétroliers, plus on dit qu'elle se trouve en aval de l'industrie. Les opérations en aval sont des processus pétroliers et gaziers qui fournissent le plus de produits étroitement liés aux consommateurs comme le gaz naturel liquéfié, l'essence, le mazout, le caoutchouc synthétique, les plastiques, les lubrifiants, l'antigel, les engrais et les pesticides [48].

De façon globale, les compagnies pétrolières en amont sont principalement impliquées dans la découverte, l'extraction et la production de pétrole. Les compagnies pétrolières en aval s'occupent plutôt du raffinage et de la livraison des produits pétroliers aux consommateurs [90].

3.3 Organisme d'accueil

3.3.1 Historique de Sonatrach

L'entreprise SONATRACH (société nationale pour le transport et la commercialisation des hydrocarbures) a été créé le 31/12/1963 par le décret N°63-491.

Le 22/09/1966, les statuts ont été modifiés par le décret N° 66/292 et la SONATRACH devient (société nationale pour la recherche, la production et la transformation des hydrocarbures). Le 24/02/1971, la nationalisation s'étendra à tous les

secteurs des hydrocarbures cela a conduit à une restructuration et sa réorganisation en 1985, SONATRACH s'est recentrée sur ses métiers de base que constituent les activités suivantes.

3.3.2 Activités de Sonatrach

Amont

L'Amont a en charge la recherche, le développement des gisements découverts, l'amélioration du taux de récupération et la mise à jour des réserves, l'exploitation et la production des hydrocarbures.

Aval

L'aval a en charge l'élaboration et la mise en œuvre des politiques de développements et d'exploitation de l'aval pétrolier et gazier. Elle a pour mission essentielles l'exploitation des installations liquéfaction de gaz naturel et de séparation GPL, de raffinage, de pétrochimie et de gaz industriels (hélium et azote).

Transport par canalisation

L'activité transport est confiée à la branche (TRC) dont les missions essentielles sont : l'assurance de transport par canalisation des hydrocarbures, le développement, l'exploitation et la gestion de la maintenance.

Commercialisation

La commercialisation a en charge le management des opérations de vente et de shipping dont les actions sont menées en coopération avec les filiales telles que NAFTAL pour la distribution des produits pétroliers. La première exploitation des gisements en Algérie commence à la fin de l'année 1890 dans le bassin de Chleff, mais le premier gisement ne fût trouvé qu'en 1948 dans la région d'Oued-Guetrini à 150 Km d'Alger. Dans les années 50 les travaux d'exploitation s'étendirent au SAHARA où plusieurs découvertes de pétrole et de gaz ont été faites. Entre 1953 et 1956 furent découverts les champs d'huile d'Edjeleh de Hassi Mesaoud, les champs de gaz Hassi-Messaoud ainsi que les champs de Hassi-R'mel et de Ain Amenas.

3.3.3 Missions et objectifs de Sonatrach

Sonatrach est un groupe pérolier gazier et un acteur majeure de l'industrie pétrolière. De ce fait, Sonatrach a plusieurs missions et objectifs. Dans ce qui suit, nous commençons par citer ces missions puis les objectifs.

Missions de Sonatrach

Dès sa création, SONATRACH avait pour mission de prendre en charge le transport et la commercialisation des hydrocarbures. Après avoir vu en 1966 ses mission étendues à l'ensemble des activités pétrolières, SONATRACH est confirmée dans son rôle d'outil privilégié de la politique nationale dans le domaine des hydrocarbures, le 24/02/1971. Elle s'étale dès lors à consolider le processus de récupération totale des richesses pétrolières et gazières, ainsi que la maitrise technologique, tout en sauvegardant l'approvisionnement énergétique du pays, et en pourvoyant aux recettes en devises nécessaires à son développement.

Aujourd'hui, SONATRACH assure des missions stratégiques centrées sur la recherche, la production, le traitement et la liquéfaction du GPL, de l'approvisionnement du marché national et la commercialisation des hydrocarbures liquides et gazeux sur le marché international.

Objectifs de Sonatrach

Les objectifs de SONATRACH durant les 25 années à venir consistent à doubler le rythme de la production afin d'atteindre la barre de 100 TEP (tonne équivalent pétrole) annuellement, ce qui donnera une production cumulée prévisionnelle de 2,5 milliards de TEP à la fin de l'année 2020. Si parallèlement les efforts d'exploitation et de prospection des hydrocarbures ne suffisent pas à renouveler la totalité des réserves mises en place par la découverte de nouveaux gisements compte tenu du volume d'hydrocarbures qui pourront être récupérées du sous-sol, SONATRACH ira irrémédiablement vers un déséquilibre énergétique très grave. Aujourd'hui, l'évolution de l'économie mondiale des hydrocarbures ne laisse d'autres alternatives à SONATRACH que l'adaptation, l'amélioration et modernisation des conditions de travail et de son outil de production. Pour cela, SONATRACH s'appuie sur les valeurs fondamentales de la culture de l'entreprise.

3.3.4 La région de transport centre Bejaia RTC

La direction régionale de transport de Bejaia (TRC) est l'une des directions régionales de transport des hydrocarbures de la SONATRACH (TRC) elle a pour mission :

- Le transport, le stockage et la livraison des hydrocarbures liquides et gazeux.
- La gestion et l'exploitation des différents ouvrages, les stations de pompage et terminaux ainsi que d'un port pétrolier, d'un gazoduc, et de deux oléoducs.

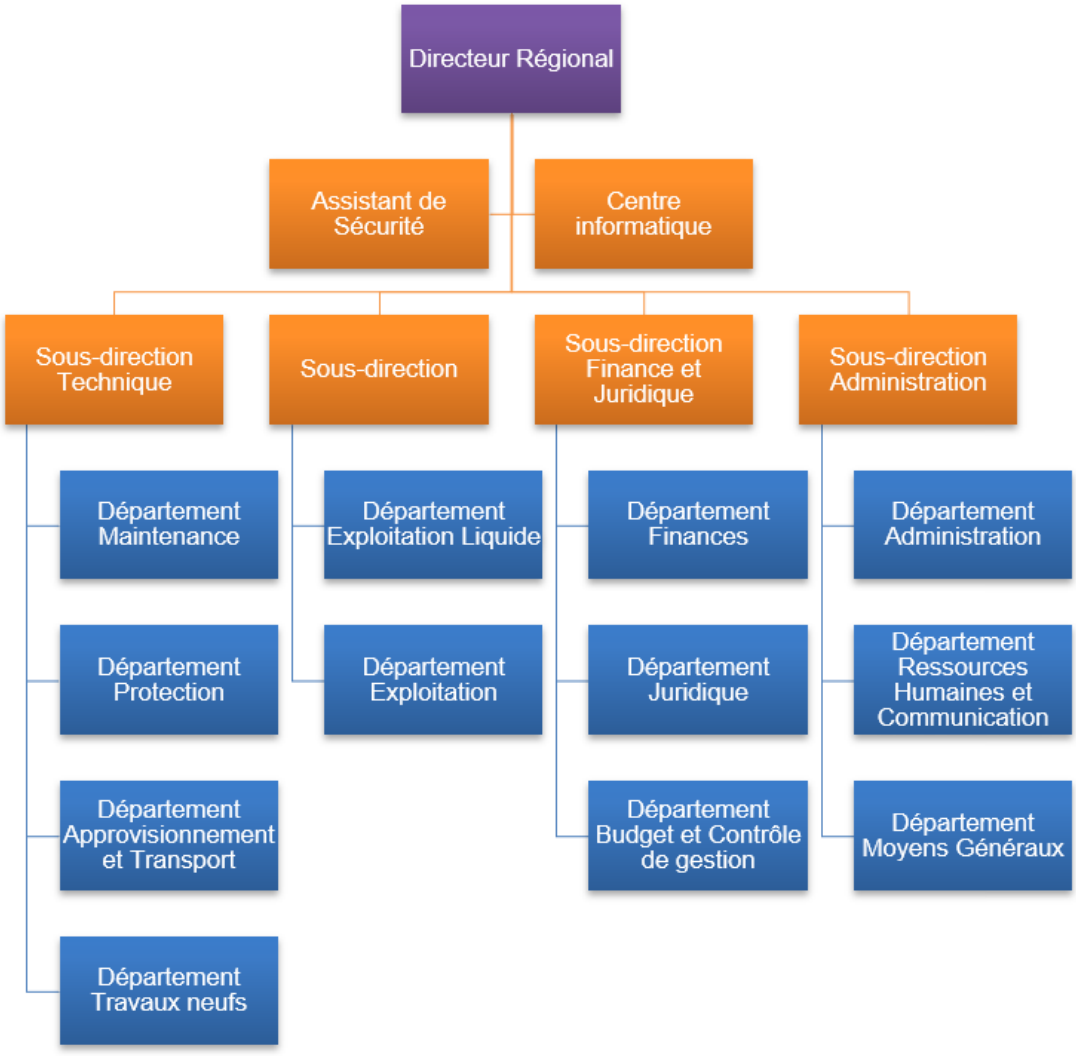


FIGURE 3.1 – Organigramme de Sonatrach

3.4 Problématique : Défaillance des oléoducs

Les pipelines sont l'épine dorsale de l'industrie pétrolière. En USA, depuis 1986, les accidents de pipeline ont déversé en moyenne 76 000 barils par an ou plus de 3 millions de gallons. Cela équivaut à 200 barils par jour. Le pétrole est de loin la substance la plus couramment déversée, suivi du gaz naturel et de l'essence [87]. En Algérie, dans les 10 dernières années, il y a eu deux fuites de gaz et une explosion d'oléoduc [86]. La plupart des grands pipelines sont en acier avec des diamètres qui varient de 8 à 47 pouces, tandis que les pipelines de distribution sont principalement en plastique avec de petits diamètres allant jusqu'à seulement 2 pouces [81]. Dans notre étude nous nous intéressons aux oléoducs qui sont en acier au carbone, qui est

un acier dont le principal composant d'alliage est le carbone. La nuance d'acier varie de la nuance A à la nuance X 80. Habituellement, l'acier de qualité supérieure est utilisé pour les pipelines à haute pression et les pipelines offshore. De plus, les nuances d'acier élevées sont fortement affectées par la présence d'impuretés, en particulier H2S. Généralement, le pétrole et le gaz sont mélangés à certaines impuretés lorsqu'ils sont extraits du champ. Ces impuretés augmentent le risque de corrosion interne. Les impuretés les plus courantes sont [81] :

H2S (gaz corrosif) : H2S forme de l'acide sulfurique en présence d'eau, qui provoque alors des piqûres, un laminage et de la corrosion.

CO2 : Lorsqu'il est exposé à l'eau, le CO2 forme de l'acide carbonique, un acide hautement corrosif.

Chlorures :les chlorures sont des substances hautement corrosives

L'Association européenne des compagnies pétrolières pour les questions d'environnement, de santé et de sécurité dans le raffinage et la distribution, CONCAWE, énumère plusieurs types de défaillances pour les oléoducs. La CONCAWE, de l'anglais COnservation of Clean Air and Water in Europe, a été créée en 1963 par un groupe de sociétés pétrolières de premier plan pour mener des recherches sur les questions environnementales liées à l'industrie pétrolière [81].

3.4.1 Types de défaillances

La CONCAWE publie des rapports, collecte et analyse les accidents de pipelines en Europe. La section suivante montre les principaux types de défaillance de pipeline selon la CONCAWE.

Panne mécanique

Les défaillances mécaniques incluent toutes les défaillances dues à une mauvaise construction ou à l'utilisation de matériaux de mauvaise qualité. Les dommages mécaniques peuvent entraîner une défaillance immédiate, une défaillance retardée ou aucune défaillance, selon la gravité des dommages. Actuellement, le moyen le plus courant de détecter les dommages mécaniques consiste à effectuer une inspection en ligne (ILI), telle que l'échographie ou le flux magnétique [71].

Corrosion

La corrosion se forme en raison de la tendance des métaux manufacturés à revenir à leur forme minérale d'origine ; ce processus est généralement très lent. La corrosion provoque une perte de métal de la paroi du pipeline qui pourrait entraîner une panne. La corrosion est considérée comme la deuxième cause la plus fréquente de défaillance du pipeline après l'intervention d'un tiers. Pour évaluer le potentiel de modification de la corrosion, le type de corrosion doit être clairement identifié. Il existe trois principaux types de corrosion, présentés ci-dessous [66].

Corrosion Externe : La corrosion externe pourrait être une corrosion atmosphérique pour les composants de pipeline hors sol exposés à l'atmosphère. Il s'agit d'un mécanisme de défaillance rare en raison de la lenteur du mécanisme atmosphérique. La corrosion externe peut également se produire en raison de la corrosion souterraine dans les pipelines enterrés. La corrosion souterraine est plus agressive que la corrosion atmosphérique en raison du mécanisme compliqué qui sous-tend cette corrosion. La corrosion souterraine peut être minimisée en utilisant une protection cathodique et un revêtement de pipeline [66]

Corrosion interne : Ce type de corrosion attaque la surface intérieure d'un pipeline. Elle est moins sévère que la corrosion souterraine mais plus dangereuse que la corrosion atmosphérique. Elle est généralement fonction du produit transporté par le pipeline [66].

Corrosion fissurante sous contrainte : La corrosion par fissuration sous contrainte est un type de corrosion induite par l'influence combinée de la contrainte de traction et de l'environnement corrosif [66].

Activité de tiers et interférence externe

La défaillance d'un tiers est le résultat de tout dommage causé par des personnes qui ne sont pas associées à un pipeline. Cela inclut les accidents non détectés et peut entraîner une défaillance à tout moment ultérieur [66]. Les statistiques sur les pipelines montrent que les activités de tiers sont la principale cause de défaillance des pipelines. 20 à 40 % de toutes les défaillances de pipelines sont causées par des dommages causés par des tiers [17]. Malgré cette réalité, les dommages causés par des tiers sont le facteur le moins pris en compte dans l'évaluation des risques liés aux pipelines [66]. De nombreux facteurs peuvent affecter la survenue de dommages causés par des tiers, tels que le type d'utilisation des terres, l'emplacement du pipeline, l'instabilité politique et son accessibilité.

Défaillance opérationnelle

La défaillance opérationnelle résulte de perturbations opérationnelles : le dysfonctionnement ou l'inadéquation d'un ou plusieurs systèmes de protection ou l'erreur des opérateurs [66]. La défaillance opérationnelle est considérée comme l'une des causes les plus rares de défaillance des pipelines, bien qu'elle puisse avoir des conséquences catastrophiques. 80% des défaillances opérationnelles sont causées par des erreurs humaines. Ce type de défaillance pourrait être considérablement réduit en exécutant régulièrement des programmes de sécurité et en offrant une formation approfondie ainsi que des tests de dépistage de drogue aux exploitants de pipelines. Des dispositifs de sécurité et une surveillance de la pression à jour pourraient également réduire le risque de cette défaillance [17].

Risques naturels

Les aléas naturels provoquent rarement une défaillance des pipelines, mais ils doivent quand même être pris en compte dans l'évaluation des défaillances en raison de leurs implications sur la sécurité publique. Les risques naturels comprennent les inondations, les mouvements de terrain, l'activité volcanique et les tremblements de terre, qui peuvent tous gravement endommager un pipeline et l'environnement. Dans la plupart des cas, des études géotechniques et hydrotechniques sont effectuées avant la construction du pipeline.

3.5 Contexte de notre étude

Lors de notre stage effectué au sein du centre informatique, on s'est intéressé à la possibilité d'intégrer les techniques d'apprentissage automatique dans les activités liées au pétrole et gaz. Actuellement, le centre informatique de la RTC est dans une phase d'étude et d'essai d'adoption des méthodes ML. En tant que stagiaires et étudiantes dans l'université d'Abderrahmane Mira de Bejaia, l'objectif de notre étude est de développer un modèle d'apprentissage automatique pour l'identification du type de défaillance qui menace un oléoduc en ayant un historique d'accidents, pour ce faire, il nous a été demandé de :

- Explorer un dataset listant les accidents dus aux pannes d'oléoducs
- Développer un modèle de prédiction de défaillances
- Exploiter les composants d'une plateforme de gestion du cycle de vie d'un modèle ML
- Étudier la capacité de déploiement en utilisant la plateforme ci-dessus
- Implémenter le modèle en utilisant un langage de programmation, soit Python.

Pour ce faire, nous envisageons d'utiliser MLflow qui nous permettra de former, de réutiliser et de déployer des modèles avec n'importe quelle bibliothèque et de les regrouper en étapes reproductibles que d'autres datascientists peuvent utiliser comme une «boîte noire», sans même avoir à savoir quelles bibliothèques on utilise.

3.6 État de l'art sur les méthodes d'apprentissage automatique pour la détection des défaillances des oléoducs

Les oléoducs et gazoducs transportent chaque jour des millions de dollars de marchandises dans le monde entier. Même s'ils constituent le moyen le plus sûr de transporter des produits pétroliers, les oléoducs connaissent encore parfois des défaillances, générant des dommages environnementaux dangereux et irréparables. Ces statistiques soulignent l'importance d'adopter la prévision des pannes et la planification de la maintenance pour établir des stratégies de prévention et d'intervention en

temps opportun. Des efforts considérables ont été déployés au cours de la dernière décennie pour évaluer l'état des oléoducs. Dans cette section [80] ont conçu une approche floue pour la prédiction de 5 types de défaillances. Le modèle a été en mesure de prédire de manière satisfaisante les défaillances de pipeline dues à des risques mécaniques, opérationnels, de corrosion, de tiers et naturels avec un pourcentage de validité moyen de 83 %.

Les deux modèles sont incapables de toujours prédire avec précision le type de défaillance «risque naturel» par rapport à d'autres cas réels. Cela est dû au fait que les cas réels de défaillance par «risque naturel» soient rares. En conséquence, les deux modèles n'ont pas été suffisamment entraînés pour prédire ce type de défaillance. En outre, la défaillance des pipelines due à des «risques naturels» est en fait principalement associée à la sismicité de la zone, mais une telle variable n'est pas présente dans les données réelles. Des études complémentaires sont donc nécessaires pour prendre en compte cette limitation [81].

De nombreux modèles ont été développés au cours de la dernière décennie pour prévoir les défaillances et les conditions des pipelines. Cependant, ceux-ci sont basés généralement sur un type particulier de données (les données issues du CONCAWE) et ne peuvent être généralisés pour n'importe quelles données. Le problème de faible détection figure aussi dans la classification multi-classes. Ceci est du au fait que quelques défaillances ne peuvent pas être prédites avec des données historiques.

3.7 Conclusion

Dans ce troisième chapitre, nous avons abordé quelques notions de l'industrie pétrolière déconcertées lors de notre stage au sein de l'entreprise nationale des hydrocarbures Sonatrach. À l'issue de celui-ci nous avons pu cerner notre problématique et contexte d'étude et ainsi synthétiser quelques travaux dans le cadre d'un état de l'art sur les méthodes d'apprentissage automatique pour la détection des défaillances des oléoducs.

Dans le chapitre suivant, nous allons présenter et implémenter des modèles d'apprentissage automatique qui traitera notre problématique .

Chapitre 4

Réalisation

4.1 Introduction

Dans ce chapitre, nous allons présenter les outils et les bibliothèques utilisés lors de l'implémention, ainsi que le processus de prétraitement de notre jeu de données. Nous allons présenter aussi le framework utilisé et son importance lors de la réalisation d'un projet ML.

4.2 Outils et bibliothèques utilisés

4.2.1 Python

Python est un langage de programmation avec une structure de données simple, efficace et idéal pour les scripts et le développement rapide d'applications dans de nombreux domaines sur la plupart des plateformes. L'interpréteur Python et la bibliothèque standard sont disponibles gratuitement sur toutes les principales plates-formes du site Python [73], et peuvent être librement distribués. Le même site contient aussi des distributions et des pointeurs vers de nombreux modules, programmes et outils Python.L'interpréteur Python est facilement étendu avec de nouvelles fonctions et types de données implémentés en C ou C++ [91].

4.2.2 Azure Databricks

Azure Databricks fournit les dernières versions d'Apache Spark et vous permet une intégration transparente avec les bibliothèques open source. Lancez des clusters et construisez rapidement dans un environnement Apache Spark entièrement géré avec l'échelle mondiale et la disponibilité d'Azure [23].

Cluster Azure Databricks

Un cluster Azure Databricks est un ensemble de ressources de calcul et de configurations sur lesquelles vous exécutez des charges de travail d'ingénierie de données, de science des données et d'analyse de données, telles que des pipelines ETL de production, des analyses de streaming, des analyses ad hoc et l'apprentissage automatique [24]. L'environnement d'exécution utilisé est 10.0 ML (Scala 2.12, Spark 3.3.0) car il dispose des dernières versions de chaque framework ou bibliothèque d'apprentissage automatique dont nous pourrions avoir besoin.

Apache Spark

Apache Spark est un framework de traitement de données qui peut effectuer rapidement des tâches de traitement sur de très grands ensembles de données et peut également répartir les tâches de traitement de données sur plusieurs ordinateurs, seul ou en tandem avec d'autres outils informatiques distribués [74].

4.2.3 MLflow

MLflow est une plate-forme open source qui permet de structurer le processus de développement ML tout en laissant aux utilisateurs un maximum de flexibilité À la base, les projets MLflow ne sont qu'une convention pour organiser et décrire votre code afin de permettre à d'autres data scientists (ou outils automatisés) de l'exécuter [95].

Plus précisément, MLflow offre trois composants, qui peuvent être utilisés ensemble ou séparément :

MLflow Tracking : est une API qui permet de suivre les exécutions d'expériences y compris le code utilisé, les paramètres, les entrées données, métriques et fichiers de sortie arbitraires afin de les enregistrer et de comparer les paramètres et les résultats [1]. Ces exécutions peuvent ensuite être interrogées via une API ou une interface utilisateur.

MLflow Project : un format simple pour empaqueter du code dans des projets réutilisables. Chaque projet définie son environnement (les bibliothèques logicielles requises...ect), le code à exécuter et les paramètres qui peuvent être utilisés pour appeler le projet de manière programmatique dans un flux de travail à plusieurs étapes ou dans des outils automatisés [95].

MLflow Models : un format commun pour le packaging des modèles (à la fois le code et les données nécessaires) qui peut fonctionner avec divers outils de déploiement [95].

4.3 Présentation de l'ensemble de données

Notre étude repose principalement sur les données historiques présentées dans le rapport CONCAWE [17], qui affiche tous les accidents survenus dans le réseau de canalisations européen depuis 1971 jusqu'à 2020. Le rapport cite la cause de chaque accident, qui est la sortie du modèle, ainsi que certains attributs de l'oléoduc, qui sont les entrées du modèle. Dans cette section, nous donnons une brève description de l'ensemble de données qui est un résumé des informations présentées dans ce rapport.

Concawe a collecté 50 ans de données sur les déversements sur les oléoducs transnationaux européens. Ce rapport couvre la performance de ces pipelines en 2020 et une perspective historique complète depuis 1971. La performance sur l'ensemble des 50 années est analysée de différentes manières, y compris les volumes bruts et nets de déversement, et les causes de déversement regroupées en cinq catégories principales : défaillance mécanique, opérationnelle, corrosion, risque naturel et tiers. Dans ce qui suit, nous mettons en évidence la signification de chaque caractéristique figurant dans le dataset.

- Spillage ID : identifiant de l'accident
- Year : année de l'occurrance de l'accident
- Pipe dia o : diamètre de l'oléoduc
- Service : le type de fluent
- Fatalities : le nombre de morts
- Injuries : le nombre de blessés
- Gross spilled volume : ou volume brut déversé, désigne la quantité totale estimée, exprimée en m^3, d'hydrocarbures rejetés par le réseau de canalisations à la suite de l'incident.
- Recovered oil : ou pétrole récupéré, désigne la quantité estimée, exprimée en m3, récupérés lors de l'opération de nettoyage, soit sous forme d'huile, soit dans le cadre du sol contaminé enlevé.
- Net loss : ou perte nette, désigne la différence entre le volume brut déversé et le pétrole récupéré.
- Leak first detected by : le service qui a détecté l'incident en premier lieu.
- Facility : le type d'équipement (souterrain, au dessus du sol, station de pompage).
- Facility part : la partie du pipeline.
- Age years : l'âge du pipeline
- Land use : type du terrain où se trouve l'oléoduc
- Cause category : la catégorie de la défaillance
- Cause reason : la raison de la défaillance

- Impact water bodies : enregistre si les déversements ont eu des conséquences sur le captage d'eau potable
- Impact contaminated land area (m^2) : zone contaminée par la fuite

Type d'équipement	
1	souterrain
2	hors sol
3	station de pompage

Service	
1	pétrole brut
2	produit blanc
3	mazout (chaud)
4	pétrole brut ou produit
5	Lubrifiants (chaud)

Panne détectée par	
1	R/W surveillance par le personnel du pipeline
2	opérateur P/L de suivi de routine
3	système de détection automatique
4	test de pression
5	partie tierce
6	inspection interne

Partie installation	
1	plié
2	découpé
3	tuyau
4	soupape
5	pompe
6	piège à cochon
7	petit alésage
6	inconnue

Type de terrain	
1	résidentiel à haute densité
2	résidentiel à faible densité
3	agricole
4	industriel ou commercial
5	collines boisées
6	Dénudé
7	plan d'eau

TABLE 4.1 – Clé liant les valeurs des différentes variables et leurs signification réelle [17]

Dans cet ensemble de données, il existe 5 variables (ou caractéristiques) de type

nominal qui ont été transformées en variables numériques. La figure **??** donne la codification des valeurs de ces variables. Les variables "Category" et "Reason" sont des variables alphabétiques. La signification de chaque lettre est donnée dans la table 4.2. Les autres variables sont de type numérique.

Cause primaire		Cause secondaire		Raison	
A	Mécanique	Ab	Design et matériel	1	Design incorrect
				2	Matériel défectueux
				3	Spécification de matériau incorrecte
				4	Age
		Aa	Construction	5	Soudure défectueuse
				6	Dégats de construction
				7	Installation incorrecte
B	Opérationnel	Ba	Système	8	Équipement
				9	Systèmes d'instrumentation et de contrôle
		Bb	Humain	10	Non dépressurisé ou ou vidangé
				11	Opération incorrecte
				12	Maintenance ou construction incorrecte
				13	Procédure incorrecte
C	Corrosion	Ca	Externe	14	Défaut de revêtement
				15	Défaillance de la protection cathodique
		Cb	Interne	16	Échec de l'inhibiteur
		Cc	Fissuration par corrosion corrosion		
D	Naturel	Da	Mouvement du sol	20	Glissement de terrain
				21	Affaissement
				22	Tremblement de terre
				23	Inondation
		Db	Autre		
E	Tiers	Ea	Accidentel	17	Construction
				18	Agricole
				19	Infrastructure souterraine
		Ec	Incidemment		
		Eb	Intentionnel	24	Activité terroriste
				25	Vandalisme
				26	Vol (y compris tentative)

TABLE 4.2 – Clé liant les valeurs des variables catégorie et raisons et leurs signification réelle [17]

4.4 Pré-traitement des données

4.4.1 Nettoyage

Le processus de nettoyage qu'on a adopté est basé sur l'étude [2]. Les variables : accidents mortels 'fatalities,',blessures 'injuries',volume des déversements ' spillage volume', méthode de détection des fuites ' leak detection method' et et partie de l'installation 'facility part' ont été exclues du modèle. Cela est dû au fait que ces variables ne peuvent pas être connues avant que la défaillance ne se produise réellement alors que le modèle est censé prédire le type de défaillance avant qu'elle ne se produise. Donc notre modèle est constitué de cinq principaux prédicateurs de types de défaillance des oléoducs :(1) type de produit transporté par l'oléoduc 'service',(2) emplacement de l'oléoduc 'facility',(3) âge de l'oléoduc 'Age',l'utilisation des terres 'land use', et (5) le diamètre du pipeline 'Diameter'.En outre, les données des incidents n'étaient pas toutes complètes ; certains incidents contenaient des valeurs de variables manquantes, Par conséquent, ces incidents ont été exclus du modèle, ce qui donne un total de 292 incidents à utiliser dans le modèle.

```
dff = pandas_df.drop(['Cause Category', 'Cause Reason', 'Injuries', 'Spillage ID',
                'Year', 'Contaminated land area (m2)', 'Fatalities',
                'Impact Water Bodies', 'Spillag volume (m3)_Net loss',
                'Facility part', 'Spillag volume (m3)_Gross', 'Age Years'], axis = 'columns')

dff = dff[dff.Category != 'B']
dff = dff[dff.Category != 'D']
dff = dff.dropna()
```

FIGURE 4.1 – Nettoyage des données

4.4.2 Fractionnement et normalisation des données

En ce qui concerne le fractionnement de données, celles-ci ont été divisées en deux sous-ensembles : trois quart pour la phase d'entraînement, et le reste pour le test. Pour la phase d'entraînement : nous avons utilisé le Gmean et la technique de validation croisée afin d'évaluer notre modèle. Cette technique permet d'utiliser l'ensemble des données pour l'entraînement et pour la validation. Au début, les données sont découpées en k parties (10 parties dans notre cas) à peu près égales. Tour à tour, chacune des k parties est utilisée comme jeu de test. Le reste (autrement dit, l'union des k-1 autres parties) est utilisé pour l'entraînement. Au final, chaque point a servi 1 fois dans un jeu de test, (k-1) fois dans un jeu d'entraînement.

La figure 4.2 explique le processus de la crosse validation 5 folds pour notre cas) chaque point appartient à un des cinq jeux de test (en blanc) et aux quatre autres jeux d'entraînements (en orange).

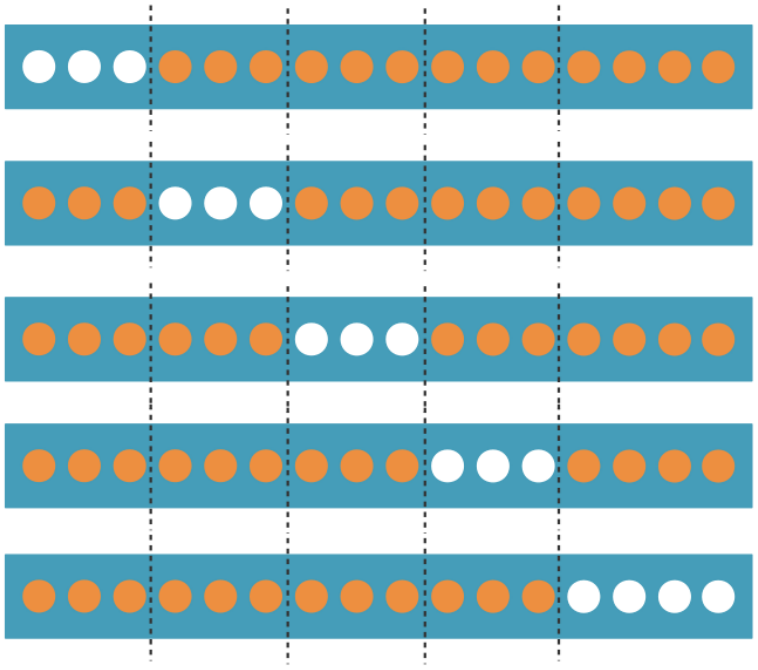


FIGURE 4.2 – Une cross-validation à 5 folds
[7]

Afin d'effectuer la validation croisée dans le langage python, la fonction KFold() [83] est généralement utilisée, toutefois, celle-ci est plus adaptée à des volumes de données plus importants. De ce fait, nous optons pour l'utilisation de la fonction StratifiedKFold() [84] mieux adaptée à des volumes de données réduits.

Pour la phase de test : le reste des données est utilisé pour calculer le résultat des expériences.

Pour la normalisation de nos données, nous avons utilisé la standardisation (Z-score) détaillée dans le chapitre 1. Pour cela, nous avons utilité la fonction *Standard-Scaler* de la bibliothèque sklearn [16].

```
from sklearn.model_selection import train_test_split
import sklearn.metrics as sm
from sklearn import preprocessing as sp
import matplotlib.pyplot as plt

xTrain, Xt, yTrain, Yt = train_test_split(X, Y, test_size= (1/4), shuffle = True, stratify= Y)

sc = sp.StandardScaler(with_mean= True, with_std= True )
sc = sc.fit(xTrain)
xTrain = sc.transform(xTrain)
Xt = sc.transform(Xt)
```

FIGURE 4.3 – Implémentation de StandardScaler

4.4.3 Suréchantillonage

Le sous-échantillonnage équilibre le jeu de données en augmentant la taille de la classe minoritaire. Pour cela ,nous avons utilisé la fonction Smote [44] de la bibliothèque imblearn .cette fonction permet de générer de nouveaux individus minoritaires qui ressemblent aux autres, sans être strictement identiques.

```
from imblearn.over_sampling import SMOTE

def data_over_sampling(X, Y):

    # summarize class distribution
    counter = Counter(Y)
    # transform the dataset
    oversample = SMOTE()
    X, Y = oversample.fit_resample(X, Y)
    # summarize the new class distribution
    counter = Counter(Y)
    print(counter)

    return X, Y

xTrain, yTrain = data_over_sampling(xTrain, yTrain)
```

▾ (2) MLflow runs

Logged 2 runs to an experiment in MLflow. Learn more

```
Counter({2: 143, 0: 143, 1: 143})
```

Command took 5.80 seconds -- by lydia.souici@se.univ-bejaia.dz at 08/09/2022 11:24:19 on My Cluster

FIGURE 4.4 – Implémentation du Suréchantillonage (smote)

4.5 Sélection et évaluation

Dans cette section, nous définissons les différentes métriques de performance que nous avons envisagé d'utiliser. Ensuite, nous utilisons ces métriques pour comparer entre les différents algorithmes.

4.5.1 Métriques de performances

Il existe plusieurs métriques de performances utilisées pour évaluer la performances des classifieurs. Nous avons envisagé d'utiliser la moyenne géométrique. Cette mesure tente de maximiser la précision sur chacune des classes tout en gardant ces précisions équilibrées. Afin de définir la moyenne géométrique, nous commençons par définir la matrice de confusion.

Matrice de confusion

Une matrice de confusion est un résumé des résultats de prédiction sur un problème de classification. Le nombre de prédictions correctes et incorrectes est résumé avec des valeurs de comptage [38], comme le montre la figure 4.5.

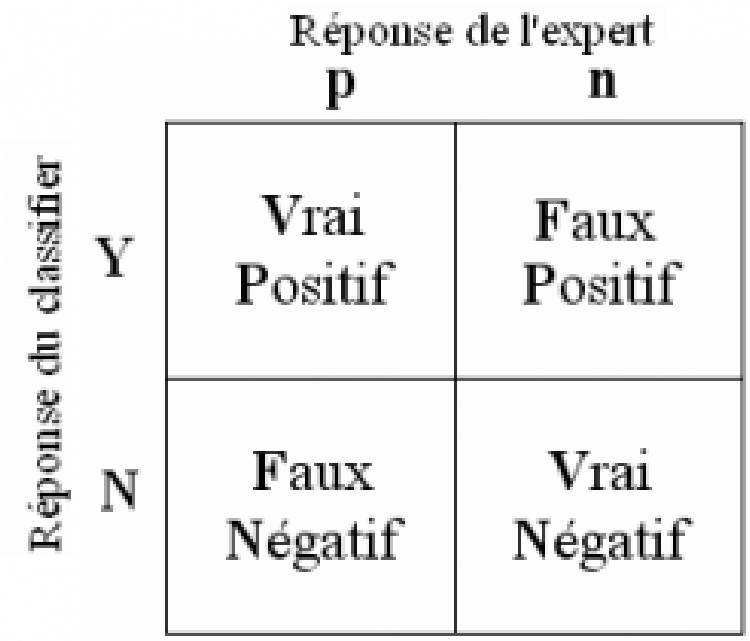


FIGURE 4.5 – Matrice de confusion pour classification binaire [38]

tel que,

- Vrai positif (TP) : Il fait référence au nombre de prédictions où le classifieur prédit correctement la classe positive comme positive.
- Vrai négatif (TN) : Il fait référence au nombre de prédictions où le classificateur prédit correctement la classe négative comme négative.
- Faux positif (FP) : Il fait référence au nombre de prédictions où le classificateur prédit à tort la classe négative comme positive.
- Faux négatif (FN) : Il fait référence au nombre de prédictions où le classificateur prédit à tort la classe positive comme négative.

Le même principe est adopté lors de la classification multi-classes, au lieu des deux valeurs positive et négative, les valeurs correspondent aux différentes classes sont considérées. Pour les métriques qui seront définies dans ce qui suit, la loi du un contre tous est adopté dans le cas de la classification multi-classes.

Accuracy

L'accuracy est la fraction des prédictions que le modèle a eu correctement. La formule de l'accuracy est la suivante [46] :

$$Accuracy = \frac{TP}{TP + FP} \tag{4.1}$$

Recall

Le Recall nous indique la fraction de tous les échantillons positifs qui ont été correctement prédits comme "positive" par le classificateur [46]. Il est également connu sous le nom de True Positive Rate (TPR), Sensitivity et Probability of detection.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

F-score

La mesure F ou le score F équilibré (F1-score) est la moyenne harmonique de la précision (Accuracy) et du Recall. Sa formule est la suivante [46],

$$F = \frac{2}{Precision^{-1} + Recall^{-1}} = 2.\frac{Recall.Precision}{Recall + Precision} = \frac{2.TP}{2.TP + FP + FN} \tag{4.3}$$

Moyenne géométrique

Cette mesure tente de maximiser la précision sur chacune des classes tout en gardant ces précisions équilibrées. Sa formule est de la forme suivante [52] :

$$Gmean = \sqrt[n]{x_1 x_2 ... x_n} \tag{4.4}$$

tel que x_i fait référence à la valeur de Recall de chaque classe.

4.5.2 Comparaison des différents algorithmes

Pour l'algorithme SVM, nous avons appliqué le Grid Search [46] pour le réglage d'hyperparamètres 4.13. Pour les algorithmes KNN et RF, des plots seront générés en fonction des paramètres de modèle et les meilleures métriques seront ainsi choisies. Dans le tableau 4.3, on remarque que la moyenne géométrique des différents algorithmes augmente après suréchantillonage. Ceci indique que le suréchantillonage a amélioré la précision de chaque classe tout en gardant ces précisions équilibrées. La prédiction est donc meilleure, même si l'accuracy n'a pas considérablement augmenté comme le montre le tableau 4.4. On remarque aussi que l'algorithme SVM avec les paramètres $kernel =' rbf'$, $gamma = 1$ et $C = 100$ a donné la meilleur valeur d'accuracy.

Algorithme	G-Mean avant SMOTE	G-Mean après SMOTE
KNN	0.50	0.67
SVM	0.47	0.61
RF	0.57	**0.69**
DT	0.54	0.67
Gaussian NB	0.45	0.53

TABLE 4.3 – Résultats de la validation croisée selon la moyenne géométrique

Algorithme	Précision avant SMOTE	Précision après SMOTE
KNN	0.56	0.58
SVM	0.65	**0.65**
RF	0.60	0.64
DT	0.58	0.60
Gaussian NB	0.63	0.64
RN	0.57	0.64

TABLE 4.4 – Comparaison des performances des différents algorithmes avant et après le suréchantillonage sur les données de test avec accuracy

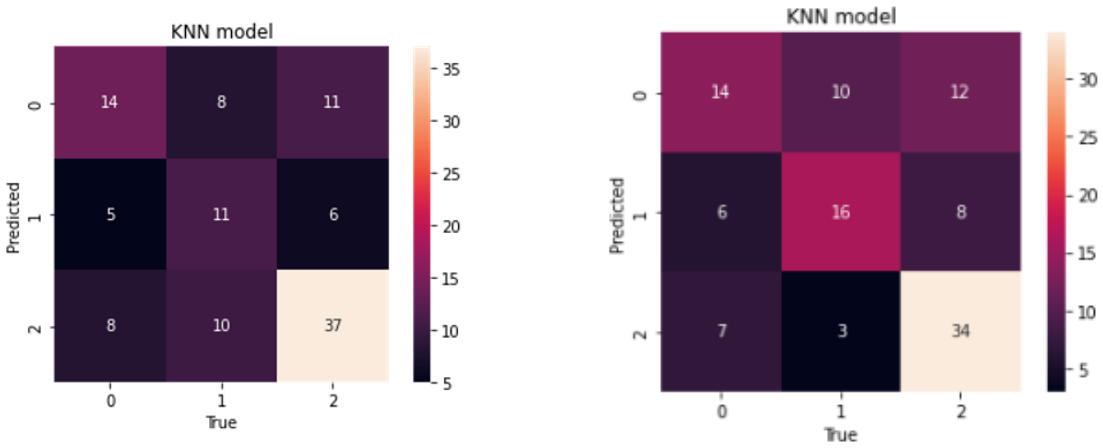
Classe	Recall	Precision
Third party	0.87	0.69
Corrosion	0.34	0.77
Mechanical failure	0.56	0.52

TABLE 4.5 – Precision et Recall des différentes classes selon l'algorithme SVM

Classe	Recall	Precision
Third party	0.76	0.76
Corrosion	0.48	0.61
Mechanical failure	0.59	0.49

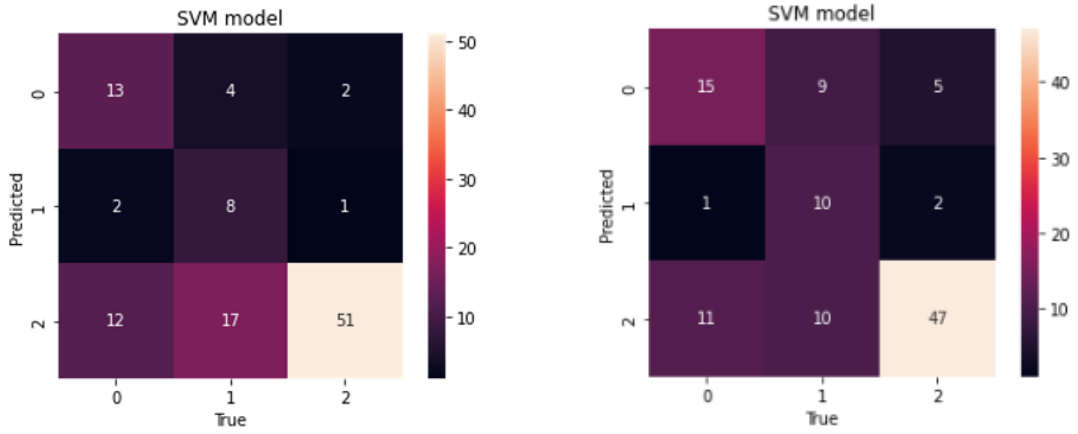
TABLE 4.6 – Precision et Recall des différentes classes selon l'algorithme RF

Les figures 4.6, 4.7, 4.8, 4.9 et 4.10 montrent clairement l'effet du suréchantillonage sur les différents algorithmes.



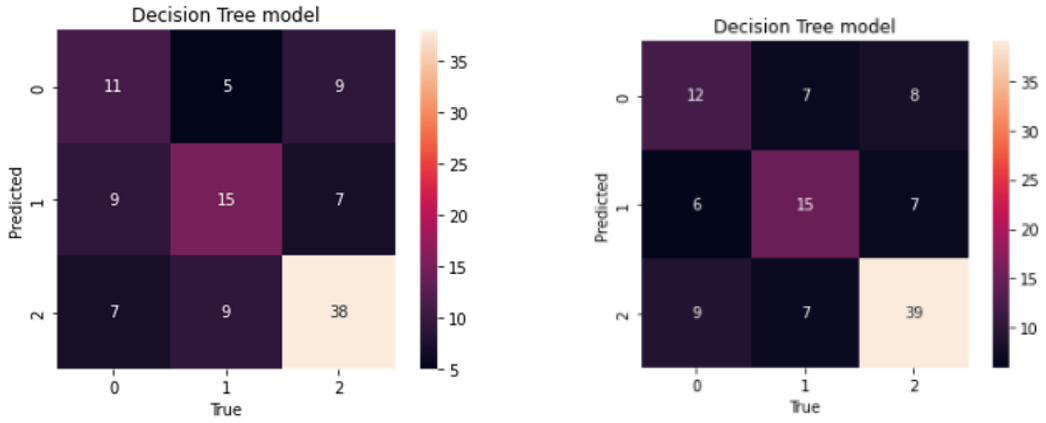
(a) avant (b) après

FIGURE 4.6 – Comparaison des matrices de confusion avant et après suréchantillonage de l'algorithme KNN avec K= 3



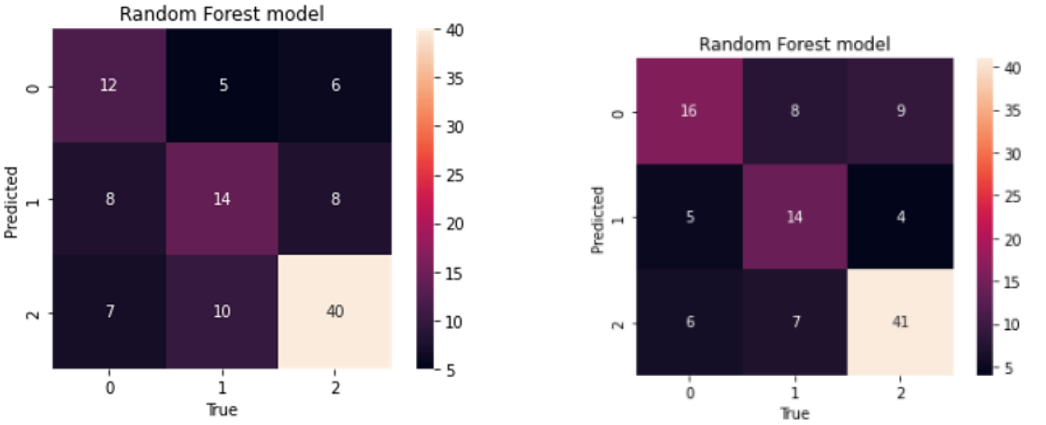
(a) avant (b) après

FIGURE 4.7 – Comparaison des matrices de confusion avant et après suréchantillonage de l'algorithme SVM avec les paramètres kernel = rbf, gamma = 0.1, C = 1



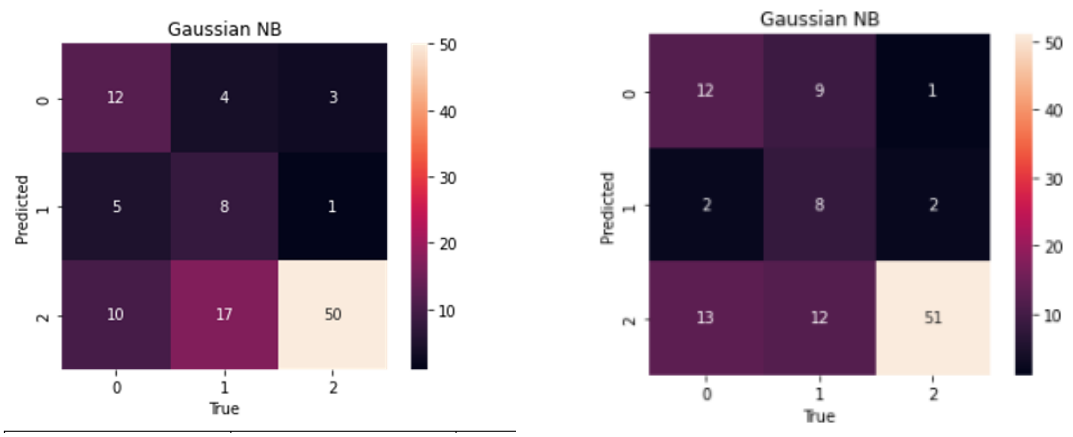
(a) avant (b) après

FIGURE 4.8 – Comparaison des matrices de confusion avant et après suréchantillonage de l'algorithme Decision Tree



(a) avant (b) après

FIGURE 4.9 – Comparaison des matrices de confusion avant et après suréchantillonage de l'algorithme RF avec le nombre d'arbres générés égale à 100



(a) avant (b) après

FIGURE 4.10 – Comparaison des matrices de confusion avant et après suréchantillonage de l'algorithme Naive Bayes

Les résultats de la classification ont montré un taux de *Recall* de 87% et une précision de 69% de l'algorithme SVM, un taux de *Recall* de 76% et une précision de 76% de l'algorithme RF pour la classe *Third party* comme le montre les tableaux 4.6 et 4.5. Celle-ci a produit les meilleures résultats de classification. Quant aux autres classes, les résultats sont moins satisfaisant ce qui a réduit l'*Accuracy*.

4.5.3 Tracking avec MLflow

Dans le but d'exposer la façon dont nous avons utilisé MLflow, nous montrons comment nous avons comparé entre quelques exécutions. Mlflow nous offre une interface qui stocke toutes les exécutions et leurs informations comme le montre la figure 4.14. Parmi tant d'exécutions, mlflow a enregistré les 5 meilleures comme le montre la figure 4.14. En selectionnant des exécutions spécifiques on peut comparer entre elles (figure 4.11). Mlflow nous offre aussi toutes les informations supplémentaires concernant les métriques et les paramètres (figure 4.12).

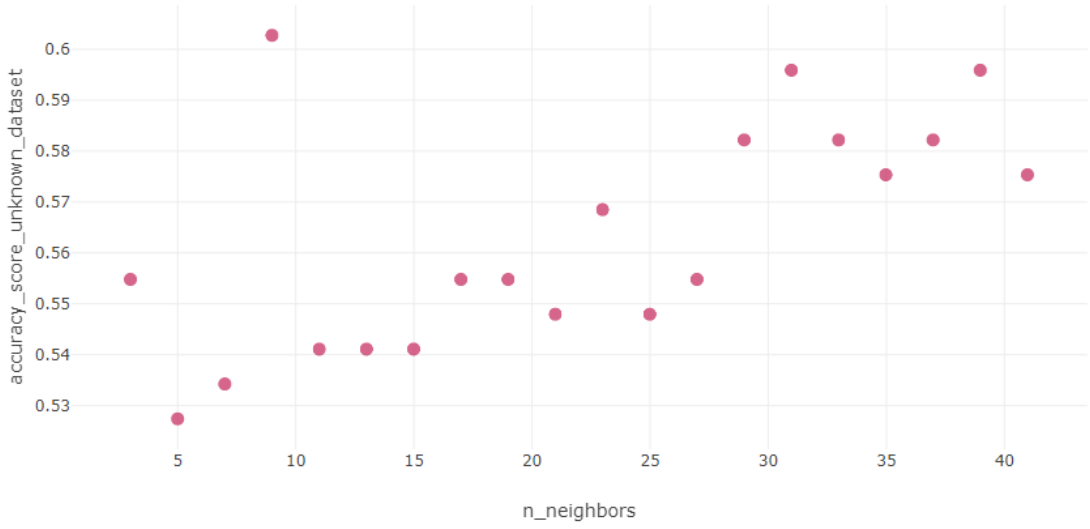


FIGURE 4.11 – Accuracy de l'algorithme KNN en fonction du nombre de voisins sur les données de test

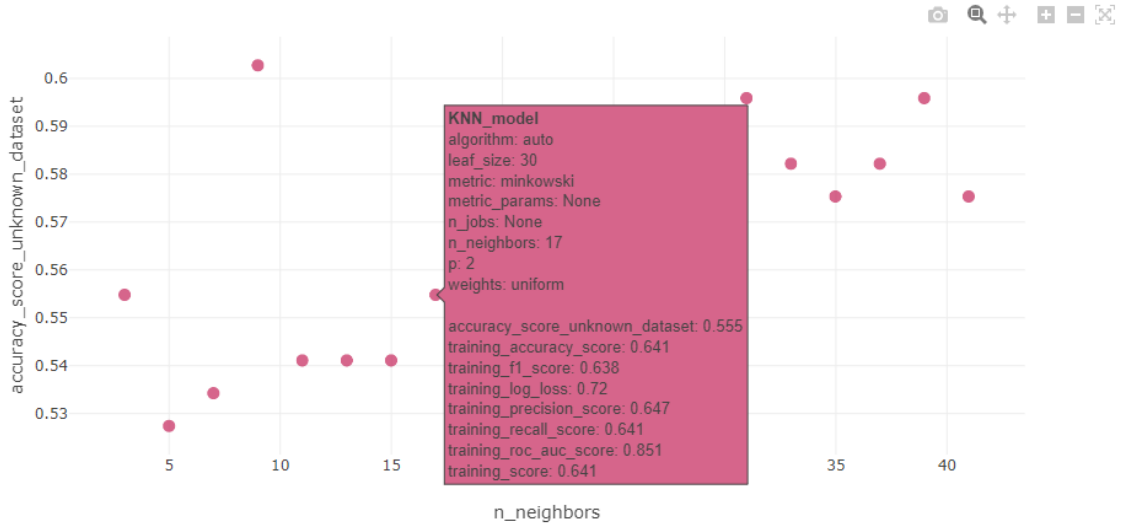
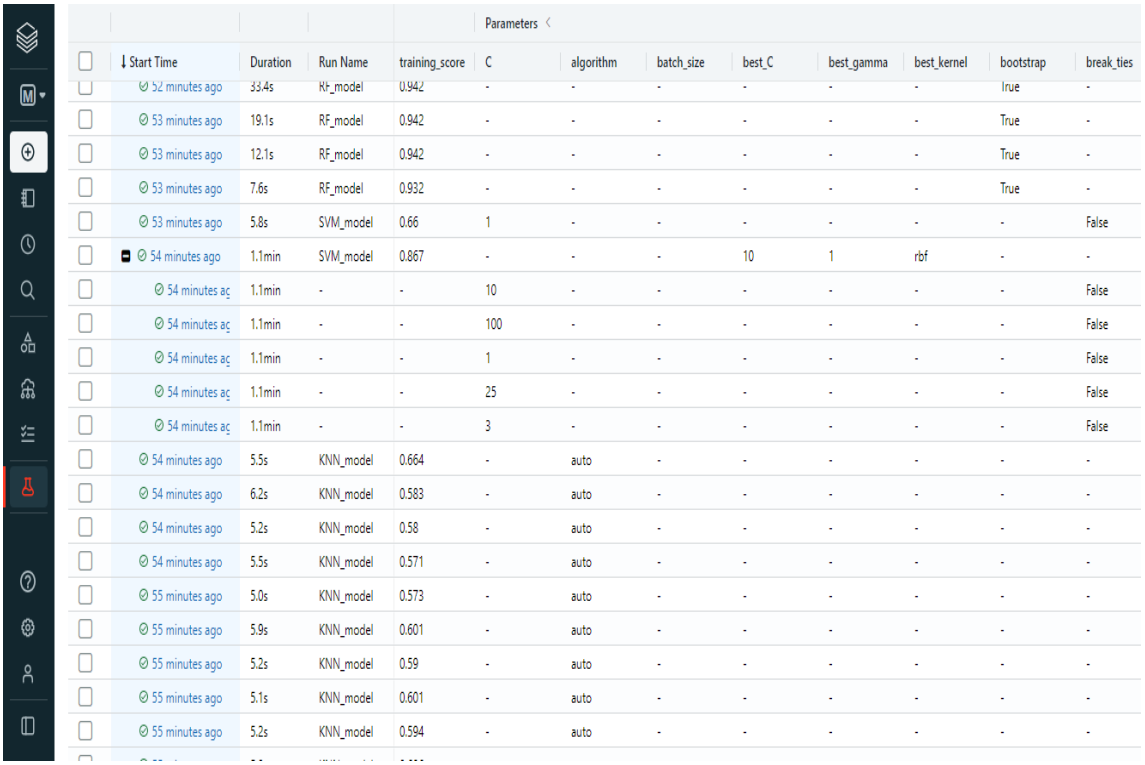


FIGURE 4.12 – Accuracy de l'algorithme KNN en fonction du nombre de voisins sur les données de test (fiche descriptive)

```python
from sklearn import svm
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [0.1, 1, 3, 10, 25, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel': ['rbf', 'poly', 'sigmoid']}
with mlflow.start_run(run_name = "SVM_model") as run:
    mlflow.sklearn.autolog()
    grid = GridSearchCV(svm.SVC(),param_grid, refit=True, verbose=2)
    grid.fit(xTrain,yTrain)
    print(grid.best_estimator_)

```

FIGURE 4.13 – Grid Search avec différentes valeurs pour les paramètres C, gamma et kernel



				Parameters							
Start Time	Duration	Run Name	training_score	C	algorithm	batch_size	best_C	best_gamma	best_kernel	bootstrap	break_ties
52 minutes ago	33.4s	RF_model	0.942	-	-	-	-	-	-	True	-
53 minutes ago	19.1s	RF_model	0.942	-	-	-	-	-	-	True	-
53 minutes ago	12.1s	RF_model	0.942	-	-	-	-	-	-	True	-
53 minutes ago	7.6s	RF_model	0.932	-	-	-	-	-	-	True	-
53 minutes ago	5.8s	SVM_model	0.66	1	-	-	-	-	-	-	False
54 minutes ago	1.1min	SVM_model	0.867	-	-	-	10	1	rbf	-	-
54 minutes ag	1.1min	-	-	10	-	-	-	-	-	-	False
54 minutes ag	1.1min	-	-	100	-	-	-	-	-	-	False
54 minutes ag	1.1min	-	-	1	-	-	-	-	-	-	False
54 minutes ag	1.1min	-	-	25	-	-	-	-	-	-	False
54 minutes ag	1.1min	-	-	3	-	-	-	-	-	-	False
54 minutes ago	5.5s	KNN_model	0.664	-	auto	-	-	-	-	-	-
54 minutes ago	6.2s	KNN_model	0.583	-	auto	-	-	-	-	-	-
54 minutes ago	5.2s	KNN_model	0.58	-	auto	-	-	-	-	-	-
54 minutes ago	5.5s	KNN_model	0.571	-	auto	-	-	-	-	-	-
55 minutes ago	5.0s	KNN_model	0.573	-	auto	-	-	-	-	-	-
55 minutes ago	5.9s	KNN_model	0.601	-	auto	-	-	-	-	-	-
55 minutes ago	5.2s	KNN_model	0.59	-	auto	-	-	-	-	-	-
55 minutes ago	5.1s	KNN_model	0.601	-	auto	-	-	-	-	-	-
55 minutes ago	5.2s	KNN_model	0.594	-	auto	-	-	-	-	-	-

FIGURE 4.14 – Historique des exécutions

4.6 Conclusion

À travers ce chapitre, nous avons présenté les outils utilisés dans l'implémentation. Nous avons aussi décrit la manière dont nous avons amélioré la classification ; le sur-échantillonage SMOTE. De plus, nous avons converti notre modèle d'un éditeur de code vers MLflow et on a conclu qu'il est très facile de convertir un modèle d'apprentissage automatique vers MLflow. Les résultats de la classification des deux

classes *panne mécanique* et *corrosion* ne sont pas assez satisfaisants contrairement à la classe *third party* qui a atteint un taux de Recall de 87%. Ceci est dû au fait qu'il n'y ait pas beaucoup d'échantillons appartenant aux deux autres classes et aussi parce que les occurrences de ces accidents ne suivent pas un modèle qui pourrait être appris par ces algorithmes.

Conclusion

L'objectif de ce mémoire a été de montrer une application de l'apprentissage automatique en entreprise, plus précisément dans l'industrie pétrolière. L'utilisation de MLflow nous a permis de tracker les performances du modèle en fonction de plusieurs métriques. Les données ont été divisées selon le type de défaillance en 3 classes : corrosion, panne mécanique et activité tierce. Plusieurs algorithmes ont été implémentés en utilisant les bibliothèques python, également Sklearn. Les résultats obtenus montrent une bonne classification de l'activité tierce. De plus, le suréchantillonage a amélioré la classification multiclasses. Le meilleur résultat par rapport à l'accuracy est obtenu par le SVM avec 65% et par rapport à la moyenne géométrique, par le Random Forest avec 69%. Malgré l'efficacité des algorithmes cités précédemment, les deux classes : corrosion et panne mécanique n'ont pas eu un taux de détection élevé. Ceci est du au fait que les données historiques ne sont pas efficaces dans la classification de ces deux classes. Toutefois, l'amélioration de la classification pourrait consister à utiliser une autre approche de suréchantillonage plus performante, ou la combinaison du suréchantillonage et sous-échantillonage dans le cas où une plus grande quantité de données serait disponible.

Bibliographie

[1] Databricks 2022. Guide MLflow | databrick sur AWS, 2022. https://docs.databricks.com/applications/mlflow/index.html, [accessed :2022-08-30].

[2] Bassem Abdrabou. *Failure prediction model for oil pipelines*. PhD thesis, Concordia University, 2012.

[3] Khan Academy. Diagramme tiges-feuilles, 2019. https://fr.khanacademy.org/math/be-4eme-secondaire2/x213a6fc6f6c9e122:statistiques-1/x213a6fc6f6c9e122:graphiques-statistiques/a/stem-and-leaf-plots-review, [accessed : 30-05-2022].

[4] Marcel Anee. Les méthodes de normalisation avec scikit-learn, 2021. https://www.alliage-ad.com/tutoriels-python/les-methodes-de-normalisation/ [accessed :13-06-2022].

[5] Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the machine learning lifecycle : Desiderata, methods, and challenges. *ACM Computing Surveys (CSUR)*, 54(5) :1–39, 2021.

[6] Chloé-Agathe Azencott. *Introduction au machine learning*. Dunod, 2019.

[7] Chloé-Agathe Azencott. Mettez en place un cadre de validation croisée, 2021. https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308241-mettez-en-place-un-cadre-de-validation-croisee,[accessed :15-09-2022].

[8] Lotfi Baghli. *Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques*. PhD thesis, Université Henri Poincaré-Nancy I, 1999.

[9] Karthik R Bandi and Sargur N Srihari. Writer demographic classification using bagging and boosting. In *Proc. 12th Int. Graphonomics Society Conference*, pages 133–137, 2005.

[10] Dhaya Sindhu Battina. An intelligent devops platform research and design based on machine learning. *training*, 6(3), 2019.

[11] Giuseppe Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[12] Petra Brosch, Gerti Kappel, Philip Langer, Martina Seidl, Konrad Wieland, and Manuel Wimmer. An introduction to model versioning. In *International school on formal methods for the design of computer, communication and software systems*, pages 336–398. Springer, 2012. `http://www.sti.uniurb.it/events/sfm12mde/slides/kappel.pdf`.

[13] Jason Brownlee. *Imbalanced classification with python : Better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery, 2020.

[14] Jason Brownlee. *Data preparation for machine learning*. Machine Learning Mastery, 2022.

[15] Bruce G Buchanan. A (very) brief history of artificial intelligence. *Ai Magazine*, 26(4) :53–53, 2005.

[16] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. Api design for machine learning software : experiences from the scikit-learn project. *arXiv preprint arXiv :1309.0238*, 2013.

[17] M Cech, P Davis, W Guijt, A Haskamp, and I Huidobro Barrio. Performance of european cross-country oil pipelines. *Concawe Environmental Science for European Refining Report*, 22(6), 2020.

[18] Tomas Cerny, Michael J Donahoo, and Michal Trnka. Contextual understanding of microservice architecture : current and future directions. *ACM SIGAPP Applied Computing Review*, 17(4) :29–45, 2018.

[19] Ananya Chattopadhyay, Sushruta Mishra, and Alfonso González-Briones. Integration of machine learning and iot in healthcare domain. In *Hybrid artificial intelligence and IoT in healthcare*, pages 223–244. Springer, 2021.

[20] Serena H Chen and Carmel A Pollino. Good practice in bayesian network modelling. *Environmental Modelling & Software*, 37 :134–145, 2012.

[21] Ankit Choudhary. Decoding the black box : An important introduction to interpretable machine learning models in python, 2019. `https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-lear-\\ning-models-python/` [accessed :15-06-2022].

[22] connaissances des énergies. Pétrole : formation, exploration et production, enjeux et chiffres clés, 2014. `https://www.connaissancedesenergies.org/fiche-pedagogique/petrole`, [accessed :2022-09-04].

[23] Azure databricks. Azure databricks | microsoft azure, . https://azure.microsoft.com/fr-fr/products/databricks/,[accessed :15-09-2022].

[24] Azure databricks. Clusters - azure databricks, . https://learn.microsoft.com/en-us/azure/databricks/clusters/,[accessed :16-09-2022].

[25] DataRobot. What a machine learning pipeline is and why it's important, 2020. http:/www.adresse.com/, [accessed : 31-05-2022].

[26] Datatron. What is model drift, 2021. https://datatron.com/what-is-model-drift/#:~:text=Concept\%20drift\%20is\%20a\%20type,s)\%20change(s) [accessed :10-07-2022].

[27] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10) :78–87, 2012.

[28] Gérard Dreyfus, JM Martinez, M Samuelides, MB Gordon, F Badran, S Thiria, and L Hérault. *Réseaux de neurones*, volume 39. Eyrolles Paris, 2002.

[29] S. Durga, Rishabh Nag, and Esther Daniel. Survey on machine learning and deep learning algorithms used in internet of things (iot) healthcare. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1018–1022, 2019. doi : 10.1109/ICCMC.2019.8819806.

[30] Sean R Eddy. What is dynamic programming ? *Nature biotechnology*, 22(7) : 909–910, 2004.

[31] IBM Cloud Education. Analyse exploratoire des données, 2020. https://www.ibm.com/fr-fr/cloud/learn/exploratory-data-analysis, [accessed : 23-05-2022].

[32] IBM Cloud Education. Apprentissage automatique, 2020. URL https://www.ibm.com/fr-fr/cloud/learn/machine-learning. https://www.ibm.com/fr-fr/cloud/learn/machine-learning, [accessed : 27-05-2022].

[33] IBM Cloud Education. What is etl (extract, transform, load ?, 2020. https://www.ibm.com/cloud/learn/etl [accessed :10-07-2022].

[34] IBM Cloud Education. What is a data warehouse ?, 2020. https://www.ibm.com/cloud/learn/data-warehouse [accessed :10-07-2022].

[35] Floris MA Erich, Chintan Amrit, and Maya Daneva. A qualitative study of devops usage in practice. *Journal of software : Evolution and Process*, 29(6) : e1885, 2017.

[36] Bence Ferdinandy, Linda Gerencsér, Luca Corrieri, Paula Perez, Dóra Újváry, Gábor Csizmadia, and Ádám Miklósi. Challenges of machine learning model validation using correlated behaviour data : evaluation of cross-validation strategies and accuracy measures. *PloS one*, 15(7) :e0236092, 2020. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0236092.

[37] KROHNE Messtechnik GmbH. Exploration et production dans l'industrie pétrolière & gazière, 2021. https://krohne.com/fr/industries/industrie-petroliere-gaziere/exploration-production-industrie-petroliere-gaziere,[accessed :2022-09-04].

[38] Gavin Hackeling. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.

[39] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning : data mining, inference, and prediction*, volume 2. Springer, 2009.

[40] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[41] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8) :832–844, 1998.

[42] Judith Hurwitz and Daniel Kirsch. Machine learning for dummies. *IBM Limited Edition*, 75, 2018.

[43] IBISWorld. IBISWorld - industry market research, reports, and statistics, 2022. https://www.ibisworld.com/default.aspx, [accessed : 23-08-2022].

[44] The imbalanced-learn developers. SMOTE — version 0.9.1, 2014. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html, [accessed :10-09-2022].

[45] Energy Institute. Oil and gas, 2017. https://www.energyinst.org/exploring-energy/topic/oil-and-gas,[accessed :2022-09-05].

[46] Nathalie Japkowicz and Mohak Shah. *Evaluating learning algorithms : a classification perspective*. Cambridge University Press, 2011.

[47] Neroda Justin, Escaravage Steve, and Peters Aaron. *Entreprise AIOps*. O'Reilly, 1 edition, 2021.

[48] KamilTaylan. Comprendre les entreprises pétrolières et les services de raffinage - KamilTaylan.blog, 2017. https://fr.kamiltaylan.blog/difference-between-oil-services-and-refiners/,[accessed :2022-09-04].

[49] Kamran Karimi and Howard J Hamilton. Generation and interpretation of temporal decision rules. *arXiv preprint arXiv :1004.3334*, 2010.

[50] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize ! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

[51] Sotiris Kotsiantis and P Pintelas. Combining bagging and boosting. *International Journal of Computational Intelligence*, 1(4) :324–333, 2004.

[52] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets : one-sided selection. In *Icml*, volume 97, page 179. Citeseer, 1997.

[53] Bart Larivière and Dirk Van den Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert systems with applications*, 29(2) :472–484, 2005.

[54] Philippe Leray. Réseaux bayésiens : apprentissage et modélisation de systèmes complexes. *habilitation à diriger les recherches, Université de Rouen*, 2006.

[55] Srikanth Machiraju. Why data drift detection is important and how do you automate it in 5 simple steps, 2021. https://towardsdatascience.com/why-data-drift-detection-is-important-and-how-do-you-automate-it-\\in-5-simple-steps-96d611095d93, [accessed :10-07-2022].

[56] S Madeh Piryonesi and Tamer E El-Diraby. Using machine learning to examine impact of type of performance indicator on flexible pavement deterioration modeling. *Journal of Infrastructure Systems*, 27(2) :04021005, 2021.

[57] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9 :381–386, 2020.

[58] Bruce G Marcot and Trent D Penman. Advances in bayesian network modelling : Integration of modelling technologies. *Environmental modelling & software*, 111 : 386–393, 2019.

[59] Krikor Maroukian and Stephen R Gulliver. Leading devops practice and principle adoption. *arXiv preprint arXiv :2008.10515*, 2020.

[60] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253) :68–78, 1951. https://www.tandfonline.com/doi/abs/10.1080/01621459.1951.10500769.

[61] William F McColl. Scalable computing. *Computer Science Today*, pages 46–61, 1995.

[62] Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artificial intelligence*, 267 :1–38, 2019.

[63] Raymond J Mooney. Integrating abduction and induction in machine learning. *Abduction and Induction*, pages 181–191, 2000.

[64] Gordon Moore. Moore's law. *Electronics Magazine*, 38(8) :114, 1965.

[65] Mohammed Msaaf and Fouad Belmajdoub. L'application des réseaux de neurone de type «feedforward» dans le diagnostic statique. In *Xème Conférence Internationale : Conception et Production Intégrées*, 2015.

[66] W Kent Muhlbauer. *Pipeline risk management manual : ideas, techniques, and resources*. Elsevier, 2004.

[67] Mehalli Nassim. *Mise en oeuvre de l'apprentissage automatique pour l'évaluation des positions échiquéennes.* PhD thesis, Université Mouloud Mammeri, 2019.

[68] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D Lawrence. Challenges in deploying machine learning : a survey of case studies. *ACM Computing Surveys (CSUR)*, 2020.

[69] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10) :1345–1359, 2009.

[70] Yogendra Narayan Pandey, Ayush Rastogi, Sribharath Kainkaryam, Srimoyee Bhattacharya, and Luigi Saputelli. *Machine Learning in the Oil and Gas Industry*. Springer, 2020.

[71] PD Panetta, AA Diaz, RA Pappas, TT Taylor, RB Francini, and KI Johnson. Mechanical damage characterization in pipelines. *Pacific Northwest National Laboratory, US Dept. of Energy, Richland, WA*, 2001.

[72] S Madeh Piryonesi and Tamer E El-Diraby. Role of data analytics in infrastructure asset management : Overcoming data size and quality problems. *Journal of Transportation Engineering, Part B : Pavements*, 146(2) :04020022, 2020.

[73] python. Welcome to python.org, 2020. `https://www.python.org/`, [accessed : 29-08-2022].

[74] Quantmetry. Présentation d'apache spark et avantages de ce framework. `https://www.quantmetry.com/glossaire/spark/`„[accessed :17-09-2022].

[75] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3) :221–234, 1987.

[76] Emmanuel Raj. *Engineering MLOps : Rapidly build, test, and manage production-ready machine learning life cycles at scale*. Packt Publishing Ltd, 2021. ISBN 978-1-80056-632-3.

[77] RedHat. What is a CI/CD pipeline ?, 2022-06-02. `https://www.redhat.com/en/topics/devops/what-cicd-pipeline`.

[78] Kabita Sahoo, Abhaya Kumar Samal, Jitendra Pramanik, and Subhendu Kumar Pani. Exploratory data analysis using python. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(12) :4727–4735, 2019.

[79] Arthur L Samuel. Machine learning. *The Technology Review*, 62(1) :42–45, 1959.

[80] Ahmed Senouci, Mohamed S El-Abbasy, and Tarek Zayed. Fuzzy-based model for predicting failure of oil pipelines. *Journal of Infrastructure Systems*, 20(4) : 04014018, 2014.

[81] Ahmed Senouci, Mohamed Elabbasy, Emad Elwakil, Bassem Abdrabou, and Tarek Zayed. A model for predicting failure of oil pipelines. *Structure and Infrastructure Engineering*, 10(3) :375–387, 2014.

[82] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. Continuous integration, delivery and deployment : a systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5 :3909–3943, 2017.

[83] sklearn. sklearn.model_selection.KFold, 2020. `https://scikit-learn/stable/modules/generated/sklearn.model_selection.KFold.html`,[accessed :15-09-2022].

[84] sklearn. sklearn.model_selection.StratifiedKFold, 2020. `https://scikit-learn/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html`,[accessed :15-09-2022].

[85] Julian Soh and Priyanshi Singh. Machine learning operations. In *Data Science Solutions on Azure*, pages 259–279. Springer, 2020.

[86] Sonatrach, 2022. `https://sonatrach.com/`.

[87] Richard Stover. America's dangerous pipelines, 2021. `https://www.biologicaldiversity.org/campaigns/americas_dangerous_pipelines/#:~:text=Since\%201986\%20pipeline\%20accidents\%20have,by\%20natural\%20gas\%20and\%20gasoline.` [accessed :01-08-2022].

[88] Georgios Symeonidis, Evangelos Nerantzis, Apostolos Kazakis, and George A. Papakostas. Mlops - definitions, tools and challenges. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0453–0460, 2022. doi : 10.1109/CCWC54503.2022.9720902.

[89] TechTarget. Que signifie apprentissage par transfert ? - definition IT de whatis.fr, 2019. https://www.lemagit.fr/definition/Apprentissage-par-transfert [accessed :04-06-2022].

[90] ThePressFree. Opérations pétrolières et gazières en amont et en aval, 2022. https://thepressfree.com/operations-petrolieres-et-gazieres-en-amont-et-en-aval/,[accessed :2022-09-04].

[91] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*, volume 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[92] AI Wiki. Artifacts, 2021. https://machine-learning.paperspace.com/wiki/artifacts [accessed :10-07-2022].

[93] wikipedia. Hydrocarbure, 2022. https://fr.wikipedia.org/w/index.php?title=Hydrocarbure&oldid=196516853,[accessed :2022-09-05].

[94] Bee Wah Yap, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon Fong, Zuraida Khairudin, and Nik Nik Abdullah. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, pages 13–22. Springer, 2014.

[95] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41(4) :39–45, 2018.

[96] Aniketh Reddy Zihao Ding and Aparna Joshi. 5 - reproducibility, 2020. https://blog.ml.cmu.edu/2020/08/31/5-reproducibility/ [accessed :15-06-2022].

[97] Wei Zong, Yang-Wai Chow, and Willy Susilo. Interactive three-dimensional visualization of network intrusion detection data for machine learning. *Future Generation Computer Systems*, 102 :292–306, 2020.