

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université -Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département de Mathématiques et Informatique



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : Informatique
Domaine : Mathématiques et informatique
Filière : Informatique
Spécialité : Réseaux et ingénierie des données

Thème

Extraction des connaissances à partir d'une base de données (application à la détection de fraude dans la consommation d'électricité et du gaz)

Présenté par :

- 1) Melle OUNANE Amina
- 2) Melle MESSABIHI Meriem

Devant le jury composé de :

Dr BENDIABDELLAH Hakim	MCB	UAT.B.B (Ain Temouchent)	Président
Dr SAIDI Mohamed Reda	MCB	UAT.B.B (Ain Temouchent)	Examineur
Dr BOUHALOUAN Djamilia	MCB	UAT.B.B (Ain Temouchent)	Encadreur
Mr BENFODDA Mohamed	Ingénieur	Sonelgaz (Ain Temouchent)	Co-Encadreur

Année Universitaire 2022/2023

DECLARATION

Nous déclarons par la présente que le contenu et l'organisation du travail présenté dans ce manuscrit constituent notre propre travail original, réalisé sous la direction de Mme BOUHALOUAN et ils ne compromettent pas les droits des tiers.

OUNANE Amina
MESSABIHI Meryem
Juin 2023.

« Je vais rendre l'électricité si bon marché que seuls les riches pourront payer le luxe d'utiliser des bougies »

- Thomas Edison -

*« Soyons reconnaissants aux personnes qui nous donnent du bonheur ; elle sont les charmants
jardiniers par qui nos âmes sont fleuries. »*

- Marcel Proust -

REMERCIEMENTS

Nous avons l'insigne honneur de réserver ces quelques lignes pour exprimer notre gratitude et notre reconnaissance envers tous ceux qui ont contribué de manière directe ou indirecte à l'élaboration de ce travail.

En premier lieu, nous souhaitons exprimer notre profonde reconnaissance envers notre encadrante, Mme BOUHALOUANE. Votre expertise, votre patience et votre guidance éclairée ont été des éléments clés qui nous ont permis de donner le meilleur de nous-mêmes dans notre recherche. Vos conseils précieux et votre disponibilité constante ont joué un rôle déterminant dans la réalisation réussie de ce mémoire.

Nous tenons également à adresser nos chaleureux remerciements à notre co-encadrant, Mr BENFODDA, pour son soutien inestimable et sa contribution précieuse à notre travail. Votre expertise pointue et votre engagement envers notre succès ont grandement enrichi notre recherche et renforcé notre motivation.

Nous souhaitons exprimer notre profonde gratitude envers les membres distingués du jury, qui ont généreusement consacré leur temps et leur expertise à l'évaluation de notre travail. Vos commentaires constructifs et vos suggestions nous permettront de perfectionner notre recherche et de progresser en tant qu'étudiantes.

Nos remerciements s'étendent à tous nos enseignants du département MI de l'Université Belhadj Bouchaib d'Ain Temouchent.

Enfin, nous tenons à exprimer notre profonde gratitude envers nos familles, dont le soutien indéfectible, l'amour inconditionnel et les encouragements constants ont été une source d'inspiration tout au long de nos années d'études. Votre présence bienveillante et votre confiance en nos capacités ont été les moteurs de notre réussite.

Nous espérons sincèrement que notre travail répondra à vos attentes et reflétera notre engagement, notre passion et notre détermination dans la réalisation de ce mémoire.

Avec une reconnaissance profonde et un respect sincère.

Résumé

Notre travail consiste à combiner les domaines de l'extraction de connaissances et de l'intelligence artificielle afin de développer un système sophistiqué pour détecter les fraudes liées à la consommation d'électricité et de gaz. Pour cela, nous avons exploité la base de données en ligne de STEG et mis en œuvre deux classificateurs de pointe, XGBoost et LightGBM, qui sont parmi les meilleurs algorithmes d'apprentissage automatique pour résoudre ce type de problèmes. Après avoir réalisé une évaluation comparative, nous avons sélectionné le modèle le plus performant, LightGBM, en tenant compte de plusieurs mesures démontrant sa supériorité. Grâce à ce modèle, nous avons obtenu un taux de précision de 95,00% et un score de 88,71%, ce qui nous a valu une excellente position, la 19ème place sur 295 participants, lors du challenge Zindi. Ces résultats témoignent de l'efficacité de notre approche novatrice et de notre engagement à relever les défis liés à la détection des fraudes énergétiques.

Mot clés : Extraction de connaissances, intelligence artificielle, apprentissage automatique détection de fraudes, consommation d'électricité et de gaz, Base de données révérencielle STEG, classificateurs XGBoost et LightGBM, challenge Zindi.

Abstract

Our work involves combining the fields of knowledge extraction and artificial intelligence to develop a sophisticated system for detecting fraud related to electricity and gas consumption. To achieve this, we leveraged the online database of STEG and implemented two state-of-the-art classifiers, XGBoost and LightGBM, which are among the top machine learning algorithms for solving such problems. After conducting a comparative evaluation, we selected the most performant model, LightGBM, taking into account several metrics demonstrating its superiority. Using this model, we achieved a precision rate of 95.00% and a score of 88.71%, resulting in an excellent 19th position out of 295 participants in the Zindi challenge. These results showcase the effectiveness of our innovative approach and our dedication to tackling challenges in energy fraud detection.

Keywords: Knowledge extraction, artificial intelligence, machine learning fraud detection, electricity and gas consumption, Rev.STEG database, XGBoost and LightGBM classifiers, Zindi challenge.

ملخص:

في هذا المشروع كانت مهمتنا هي الجمع والمزج بين مجالات استخراج المعرفة والذكاء الاصطناعي لتطوير نظام متطور للكشف عن الاحتيال المتعلق باستهلاك الكهرباء والغاز. للقيام بذلك، استخدمنا قاعدة بيانات STEG عبر الإنترنت ونفذنا اثنين من أحدث المصنفات، XGBoost وLightGBM، وهما من بين أفضل خوارزميات التعلم الآلي لحل مثل هذه المشكلات. بعد تقييم مقارن، اخترنا النموذج الأكثر نجاحًا، LightGBM، مع الأخذ في الاعتبار العديد من التدابير التي تظهر تفوقه. بفضل هذا النموذج، حصلنا على معدل دقة 95,00% ودرجة 87,71%، مما منحنا مركزًا ممتازًا، المركز التاسع عشر من بين 295 مشاركًا، خلال تحدي Zindi تُظهر هذه النتائج فعالية نهجنا المبتكر والتزامنا بالتصدي لتحديات الكشف عن الاحتيال في مجال الطاقة.

الكلمات الرئيسية: استخراج المعرفة، والذكاء الاصطناعي، والكشف عن الاحتيال في التعلم الآلي، واستهلاك الكهرباء والغاز، وقاعدة بيانات Rev.STEG، ومصنفات XGBoost وLightGBM، وتحدي Zindi

TABLE DE MATIERE

III.3.3 L'apprentissage semi-supervisé.....	13
III.3.4 Apprentissage par renforcement.....	13
III.4 Evaluation des performances des algorithmes d'apprentissage	13
III.4.1 Evaluation du fractionnement Train-Test	13
III.4.2 Evaluation du modèle avec la validation croisée	14
III.4.3 La matrice de confusion	14
III.5 Quelques familles d'apprentissage.....	15
III.5.1 Algorithme de classification (classification)	15
III.5.2 Algorithme de régression (regression)	17
III.5.3 Algorithme de catégorisation (clustering)	18
IV PRESENTATION DU PROCESSUS DE L'EXTRACTION DE CONNAISSANCES	19
IV.1 Les étapes d'un processus d'ECD.....	19
IV.1.1 Nettoyage et intégration des données	20
IV.1.2 Le pré-traitement des données.....	20
IV.1.3 Fouille de données (Data Mining)	20
IV.1.4 Evaluation et présentation	20
V FOUILLE DE DONNEES	21
V.1 Quelques définitions	21
V.2 Historique	21
V.3 Tâches du Data Mining	22
V.3.1 Classification	22
V.3.2 Estimation	22
V.3.3 La prédiction.....	22
V.3.4 Règles d'association.....	23
V.3.5 La segmentation.....	23
V.3.6 Description	23
V.4 Méthodes du Data Mining.....	23
V.5 Domaines d'application	24
V.5.1 Le secteur bancaire	24
V.5.2 La détection de fraude	25
V.5.3 Le secteur des assurances	25

TABLE DE MATIERE

V.5.4 La médecine	25
VI APPRENTISSAGE AUTOMATIQUE DANS LA DETECTION DE FRAUDES	25
VII CONCLUSION	26
CHAPITRE II : APPLICATIONS DES TECHNIQUES D'APPRENTISSAGE APERÇU DE L'ETAT DE L'ART	27
I INTRODUCTION.....	28
II QUELQUES TRAVAUX CONNEXES	28
II.1 Apprentissage en profondeur	28
II.2 Apprentissage supervisé	29
III SYNTHÈSE.....	32
IV CONCLUSION	33
CHAPITRE III : LA FAMILLE DES ALGORITHMES GARDIEN BOOSTING	34
1 INTRODUCTION	34
II HISTORIQUE.....	34
III QU'EST-CE QUE LE GRADIENT BOOSTING ?	34
IV LE BOOSTING EN MACHINE LEARNING	35
V PRINCIPAUX TYPES D'ALGORITHMES DE BOOSTING.....	35
VI COMPARAISON DES CARACTERISTIQUE DES MODELES	37
VII FONCTIONNEMENT DES ALGORITHMES DE BOOSTING	37
VIII AVANTAGES DU BOOSTING	38
IX DOMAINE D'APPLICATION D'ALGORITHMES DE GRADIENT BOOSTING	38
X CONCLUSION	39
CHAPITRE IV: PROPOSITION DE MODELES D'APPRENTISSAGE POUR LA DETECTION DE FRAUDES D'ENERGIE [ELECTRICITE – GAZ].....	40
I INTRODUCTION.....	40
II CONCEPTION DES MODELES	40
II.1 Spécification fonctionnelle du suivi des consommations	40
II.2 Description des modèles	41

TABLE DE MATIERE

II.3 Comparaison des deux modèles	43
II.4 Métrique d'évaluation	43
II.5 Jeux de données pour les modèles de détection et classification	44
II.5.1 Structuration de la base de données	44
II.5.2 Définition des structures de données	46
II.6 Partitionnement des données	47
III APPLICATION DU PROCESSUS D'ECD DANS NOTRE APPROCHE	47
IV EXPERIMENTATION ET RESULTATS	47
IV.1 Architecture globale	47
IV.2 Application des modèles (XGBoost- LightGBM).....	48
IV.3 Comparaison des modèles et Synthèse.....	56
IV.4 Généralisation du modèles [LightGBM]	59
V VALIDATION DES RESULTATS PAR ZINDI.....	63
VI CONCLUSION	66
CONCLUSION GENERALE.....	67
ANNEXES	
Annexe 1	68
Annexe 2	72
BIBLIOGRAPHIE	80

FIGURE 1	Contexte général des consommations frauduleuses en Algérie (Découverte de plus de 704 manipulations sur compteur dans la région de l'ouest.....	2
FIGURE 2	Résumé des contributions de notre étude.....	3
FIGURE I.1	La nature interdisciplinaire de la fouille de données	6
FIGURE I.2	Les 3V du Big Data	7
FIGURE I.3	Le lien entre Machine Learning Deep Learning et intelligence artificielle	8
FIGURE I.4	Représentation graphique d'un système d'apprentissage automatique avec ses entrées et ses sorties	9
FIGURE I.5	Les différents types d'apprentissage et des exemples d'utilisation	11
FIGURE I.6	Les différentes étapes du processus d'ECD	19
FIGURE III.1	Les différentes étapes du processus d'ECD	38
FIGURE IV.1	Diagramme de classe général du fonctionnement du suivi des consommations	39
FIGURE IV.2	La différence entre lightgbm et xgboost	42
FIGURE IV.3	Compétition Zindi détection de fraude dans la consommation d'électricité et du gaz	43
FIGURE IV.4	Aperçu du fichier SampleSubmission.csv	45
FIGURE IV.5	Architecture globale du système	47
FIGURE IV.6	Importation des bibliothèques.....	48
FIGURE IV.7	Accès à Google drive	48
FIGURE IV.8	Chargement des données.....	48
FIGURE IV.9	Afficher les cinq premières et cinq dernières lignes du fichier client...	49
FIGURE IV.10	Afficher le détail du fichier importé des clients	49
FIGURE IV.11	Afficher les cinq premières et cinq dernières lignes du fichier des factures	50
FIGURE IV.12	Afficher le nombre des factures	50

LISTE DES FIGURES

FIGURE IV.13	Afficher les cinq premières lignes du fichier "test_client"	50
FIGURE IV.14	Afficher les détails du fichier "test_client"	50
FIGURE IV.15	Afficher les cinq premières lignes du fichier de soumission (sub) sur Zindi	51
FIGURE IV.16	Jointure des données client_train et invoice_train	51
FIGURE IV.17	Affichage du résultat dans 'data'	51
FIGURE IV.18.a	Distribution des clients (frauduleux (=1), non-frauduleux (=0)) par un diagramme en cercle	52
FIGURE IV.18.b	Répartition des clients frauduleux par région représentée par un histogramme.....	52
FIGURE IV.19.a	Le nombre des clients (frauduleux (=1), non-frauduleux (=0)) par « Région »	52
FIGURE IV.19.b	Le nombre des clients (frauduleux (=1), non-frauduleux (=0)) par « Distcrit »	52
FIGURE IV.20.a	Le nombre des clients (frauduleux (=1), non-frauduleux (=0)) par « catégorie du client »	53
FIGURE IV.20.b	Le nombre des clients (frauduleux (=1), non-frauduleux (=0)) par « type de compteurs »	53
FIGURE IV.21	Convertir la date de facture en entier	53
FIGURE IV.22	Convertir la date de création en entier	53
FIGURE IV.23	Convertir le type des compteurs (ELEC, GAZ) en entier (0,1)	54
FIGURE IV.24	Définir la fonction d'agrégation 'aggs'	54
FIGURE IV.25	Appliquer l'agrégation.....	54
FIGURE IV.26	Afficher résultat d'agrégation	54
FIGURE IV.27	Affichage des champs de agg_train	55
FIGURE IV.28	Jointure à gauche entre client_train et invoice_train et client_test et invoice_test	55
FIGURE IV.29	Conversion du champ client_id sur train et test en entier	55
FIGURE IV.30	Définir la validation croisée avec k=5	56
FIGURE IV.31	Les paramètres à tester pour le modèle lightgbm	57
FIGURE IV.32	Les paramètres à tester pour le modèle xgboost	57
FIGURE IV.33	Définir le modèle LGBClassifier	58
FIGURE IV.34	Définir le modèle XGBClassifier.....	58

LISTE DES FIGURES

FIGURE IV.35	Entraîner le modèle lightgbm.....	58
FIGURE IV.36	Entraîner le modèle xgboost	58
FIGURE IV.37.a	Prédiction du modèle lightgbm.....	59
FIGURE IV.37.b	Prédiction du modèle xgboost.....	59
FIGURE IV.38.a	Calculer la Matrice de confusion de lightgbm (cm)	59
FIGURE IV.38.b	Calculer la Matrice de confusion de xgboost (CM)	59
FIGURE IV.39.a	Matrice de confusion de lightgbm	59
FIGURE IV.39.b	Matrice de confusion de xgboost	59
FIGURE IV.40.a	Taux de précision pour lightgbm	60
FIGURE IV.40.b	Taux de précision pour xgboost.....	60
FIGURE IV.41.a	La courbe ROC de lightgbm	60
FIGURE IV.41.b	La courbe ROC de xgboost.....	60
FIGURE IV.42	Entraîner le modèle avec la totalité des données	62
FIGURE IV.43	Tester le modèle sur des nouveaux exemples de Zindi.....	62
FIGURE IV.44	Prédire la probabilité de fraude de chaque abonné	63
FIGURE IV.45	Soumission.....	63
FIGURE IV.46	Nos profils sur Zindi	63
FIGURE IV.47	Fraud Detection in Electricity and Gas Consumption Challenge in Zindi	64
FIGURE IV.48	Sélectionner le fichier de soumission	64
FIGURE IV.49	Score de soumission	65
FIGURE IV.50	Le classement final sur Zindi	65

TABLEAU I.1	Matrice de confusion	14
TABLEAU II.1	Une comparaison des algorithmes d'apprentissage	33
TABLEAU III.1	Comparaison synthétique entre XGBoost et LightGBM	37
TABLEAU IV.1	Description de la structure « Client »	45
TABLEAU IV.2	Description de la structure « Facturation »	45
TABLEAU IV.3	Comparaison entre Lightgbm et Xgboost d'après les résultats obtenus	61

ALGORITHME IV.1	Modèle de classification de fraude avec LightGBM utilisant la validation croisée	40
ALGORITHME IV.2	Principe LightGBM	40
ALGORITHME IV.3	Modèle de classification de fraude avec XGBoost utilisant la validation croisée	41
ALGORITHME IV.4	Principe XGBoost	41



INTRODUCTION GENERALE

« *Je crois que le génie particulier des femmes tient à l'électricité de leurs mouvements, à l'intuition de leur rôle et à la spiritualité de leur tendance* »

- Margaret Fuller -

1. Contexte du mémoire

L'*exploration de données*, connue aussi sous l'expression de *fouille de données*, *forage de données*, *prospection de données*, *data mining*, ou encore *extraction de connaissance à partir de données*, a pour objet l'extraction d'un savoir ou d'une connaissance à partir de grandes quantités de données, par des méthodes automatiques ou semi-automatiques.

Aujourd'hui, toutes les entreprises disposent de volumes plus ou moins grands de données qui caractérisent leur savoir-faire, leurs processus de fabrication, leurs données clients... Ce module vise donc à créer de la valeur ajoutée à partir de données et à permettre la construction de connaissances. Comme exemple, la prise de décision critique qui s'avère très importante dans de nombreux domaines et secteurs, s'inscrit dans ce contexte, ainsi que la détection des fraudes qui est un défi de taille dans différents domaines (réseaux, commerce, paiement,...). Cependant, les transactions frauduleuses peuvent rapidement se transformer en des pertes financières importantes dans l'absence des outils efficaces et systèmes intelligents pour y faire face et mettre ces activités à l'écart avec des sanctions.

Dans de tels contextes, les techniques d'Intelligence Artificielle peuvent apporter une aide précieuse, car elles peuvent traiter une masse importante de données, ensuite proposer pour un nouveau cas une décision fondée sur une analyse de tous les cas précédents traités par apprentissage.

De ce fait, notre travail intitulé « Extraction des connaissances à partir d'une base de Données (Application à la Détection de Fraude dans la Consommation d'Electricité et du Gaz) » s'inscrit dans le contexte de la détection et la classification de la consommation [normale/anormale] d'électricité et du gaz, afin d'analyser une masse de données recueillie et d'en extraire des connaissances utiles.



Figure 1 – Contexte général des consommations frauduleuses en Algérie (*Découverte de plus de 704 manipulations sur compteur dans la région de l'ouest. Source : HORIZONZ- Quotidien national, Avril, 2023*).

2. Problématique

La fraude ou le vol d'électricité et de gaz est un problème important pour les fournisseurs d'énergie, car il peut causer des pertes financières importantes. Les clients qui modifient ou perturbent leurs installations de comptage peuvent réduire considérablement leur consommation d'énergie, ce qui se traduit par des factures d'énergie plus faibles. Cependant, cela a un impact négatif sur les fournisseurs d'énergie, qui perdent des revenus et doivent faire face à des coûts supplémentaires pour réparer les dommages causés aux installations de comptage. En outre, la fraude peut également entraîner des perturbations dans la distribution d'électricité et de gaz, ce qui peut affecter les autres clients qui dépendent du réseau public pour leur approvisionnement en énergie. Il est donc important de détecter et de prévenir la fraude dans la consommation d'électricité et de gaz afin de garantir un approvisionnement fiable et de maintenir des coûts raisonnables pour les clients et les fournisseurs d'énergie.

Toutefois, lorsqu'il est impossible d'empêcher la fraude, celle-ci doit être détectée dès que possible, et des mesures nécessaires doivent être prises à son encontre. De là, nous signalons la nécessité des systèmes automatisés de détection des fraudes, surtout compte tenu de l'énorme trafic de données de consommation de cette énergie, et il n'est pas possible pour un humains de vérifier manuellement les consommations si elles sont frauduleuses ou non.

La construction d'un système de détection de fraude n'est pas aussi simple qu'il y paraît. Car le spécialiste (concepteur/gestionnaire) doit déterminer quelle stratégie d'apprentissage utiliser (p. ex. apprentissage supervisé ou apprentissage non supervisé) et quels algorithmes utiliser (par exemple, régression logistique, arbres de décision, etc.).

L'idéal est d'avoir des modèles forts qui se généraliseront pour faire des prédictions précises sur les données futures. Cependant, cela, n'empêche pas l'apparition de quelques problèmes lors du déroulement de ce processus. On peut avoir des modèles qui peuvent être sans intérêt, d'autres peuvent être faux, des modèles qui dépendent de ce que nous appelons – coïncidences accidentelles – ou faux positifs, des données réelles imparfaites ou certaines parties qui seront tronquées, d'autres manquantes. Et delà, tout ce qu'est découvert sera inexact, il y aura des exceptions à chaque règle et des cas non couverts par aucune règle. Ainsi, les algorithmes doivent être suffisamment robustes pour faire face aux imperfections des données et d'en extraire des régularités les plus utiles qu'elles soient.

3. Contribution - Positionnement du sujet vis-à-vis de la stratégie du Sonelgaz

Notre travail comporte un volet applicatif très important notamment vis-à-vis de la stratégie de l'entreprise Sonelgaz qui doit souvent associer une analyse exploratoire à une analyse prédictive (apprentissage supervisé) afin de découvrir des schémas de fraudes inconnus en exploitant au mieux des traces d'usage et une expertise métier.

Dans ce contexte, notre contribution se concentre principalement sur l'analyse de la consommation de gaz et d'électricité afin de développer des méthodes de détection, de classification et de prédiction. Nous extrayons des connaissances à partir d'une base de données et les appliquons pour détecter les fraudes en utilisant des algorithmes prédictifs qui agissent sur les données provenant des compteurs. Notre objectif est également de reconnaître les clients impliqués dans des activités frauduleuses. Nous avons mis en œuvre et testé deux techniques d'apprentissage : LightGBM et XGBoost.

La figure 1 résume les contributions principales de notre étude :

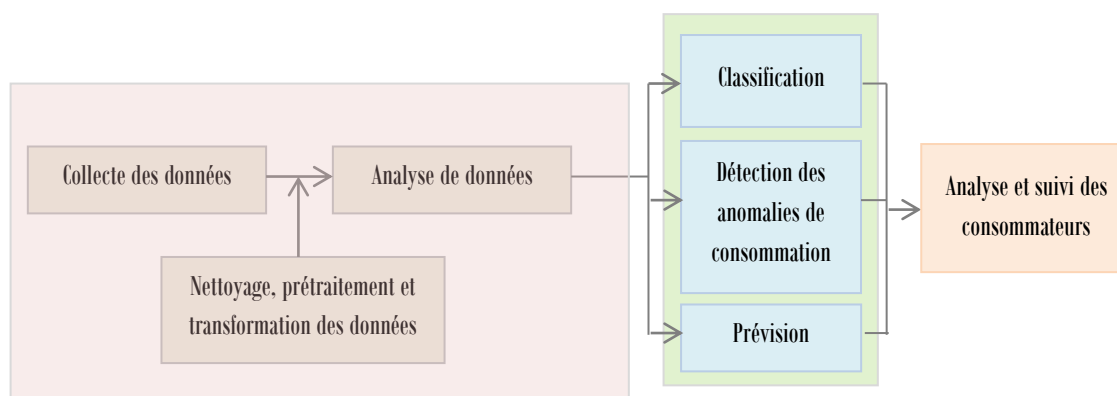


Figure 2 – Résumé des contributions de notre étude

4. Description du contenu du mémoire

- Le **chapitre 1 « Extraction de connaissances et machine learning »** introduit la notion de l'intelligence artificielle et comment elle occupe une place primordiale dans la vie humaine, nous avons parlé du data mining comme un volet qui fait cohabiter plusieurs disciplines. Nous proposons toutefois dans cette étude, les algorithmes d'apprentissages automatiques utilisés dans le cadre de l'analyse, la détection, la classification et la prédiction des problèmes de fraudes notamment dans les le réseau d'électricité et du gaz.
- Dans le **chapitre 2 « Quelques applications de techniques d'apprentissage : Aperçu d'un état de l'art »**, nous présentons un état de l'art sur certaines application des algorithmes du machine learning proposés dans la littérature.

- Le **chapitre 3 « La famille des algorithmes du gradient boosting »**, contient une description des algorithmes d'apprentissage faisant partie de la famille du gradient boosting, en particulier les deux modèles que nous avons mis en œuvre, le LightGbm et le XGBoost.
- Le **quatrième chapitre « Proposition de modèles d'apprentissage pour la détection et classification de fraudes d'énergie [Electricité – Gaz] »**, expose notre contribution en mettant en œuvre ces deux modèles d'apprentissage avec des expérimentations et tests qui nous ont permis d'élaborer un choix fondé qui a porté sur le LightGbm.
- Finalement, nous présentons quelques conclusions que nous avons tiré tout au long de cette étude, nous proposons également quelques points d'amélioration du présent travail.

EXTRACTION DE
CONNAISSANCES ET
MACHINE LEARNING

I

III.4.2	Evaluation du modèle avec la validation croisée	14
III.4.3	La matrice de confusion	14
III.5	Quelques familles d'apprentissage.....	15
III.5.1	Algorithme de classification (classification)	15
III.5.2	Algorithme de régression (regression)	17
III.5.3	Algorithme de catégorisation (clustering)	18
IV	PRESENTATION DU PROCESSUS DE L'EXTRACTION DE CONNAISSANCES	19
IV.1	Les étapes d'un processus d'ECD	19
IV.1.1	Nettoyage et intégration des données	20
IV.1.2	Le pré-traitement des données.....	20
IV.1.3	Fouille de données (Data Mining)	20
IV.1.4	Evaluation et présentation	20
V	FOUILLE DE DONNEES	21
V.1	Quelques définitions	21
V.2	Historique.....	21
V.3	Tâches du Data Mining	22
V.3.1	Classification	22
V.3.2	Estimation	22
V.3.3	La prédiction.....	22
V.3.4	Règles d'association.....	23
V.3.5	La segmentation.....	23
V.3.6	Description	23
V.4	Méthodes du Data Mining	23
V.5	Domaines d'application	24
V.5.1	Le secteur bancaire	24
V.5.2	La détection de fraude	25
V.5.3	Le secteur des assurances	25
V.5.4	La médecine	25
VI	APPRENTISSAGE AUTOMATIQUE DANS LA DETECTION DE FRAUDES	25
VII	CONCLUSION	26

I. INTRODUCTION

En tant que domaine fortement axé sur les applications, la fouille de données (data mining en anglais) a intégré de nombreuses techniques issues d'autres domaines tels que les statistiques, l'**apprentissage automatique**, la reconnaissance de formes, les **bases de données** et les entrepôts de données (data warehouse), la recherche d'informations, l'**extraction de connaissances**, la visualisation, l'algorithmique, le calcul de haute performance et de nombreux domaines d'application (voir Figure. I). La nature interdisciplinaire de la recherche et du développement en matière de fouille de données contribue de manière significative au succès de cette dernière et de ses nombreuses applications.

L'extraction de connaissances à partir des bases de données (ECD) est une discipline récente, à l'intersection des domaines des bases de données, de l'intelligence artificielle, de la statistique, des interfaces homme/ machine et de la visualisation. A partir de données collectées par des experts, il s'agit de proposer des connaissances nouvelles qui enrichissent les interprétations du champ d'application, tout en fournissant des méthodes automatiques qui exploitent cette information.

La fouille de données est au cœur du processus d'ECD. Il s'agit à ce niveau de trouver des pépites de connaissances à partir des données. Tout le travail consiste à appliquer des méthodes intelligentes dans le but d'extraire cette connaissance. Il est possible de définir la qualité d'un modèle en fonction de critères comme les performances obtenus, la fiabilité, la compréhensibilité, la rapidité de construction et d'utilisation et enfin l'évolutivité. Tout le problème de la fouille de données réside dans le choix de la méthode adéquate à un problème donné. Il est possible de combiner plusieurs méthodes pour essayer d'obtenir une solution optimale globale. Nous détaillons le principe de la fouille de données dans la section **II**.

Nous notons qu'il y a une interdépendance très forte entre les domaines sus-cité et nous pouvons parfois intégrer certaines technologies ou outils dans d'autres et vice versa, cela dépend des critères de classification.

Selon notre contexte du travail, nous illustrons dans les sections suivantes les domaines étroitement liés à la fouille de données, nous commençons par présenter un aperçu général sur l'Intelligence Artificielle (IA), ses principaux usages, l'Apprentissage Automatique (Machine learning), dans une autre section, nous décrivons le processus d'ECD et notamment l'étape de fouille de données proprement dite (data mining).., les méthodes choisies et les différentes disciplines qui influencent fortement le développement de celles-ci.

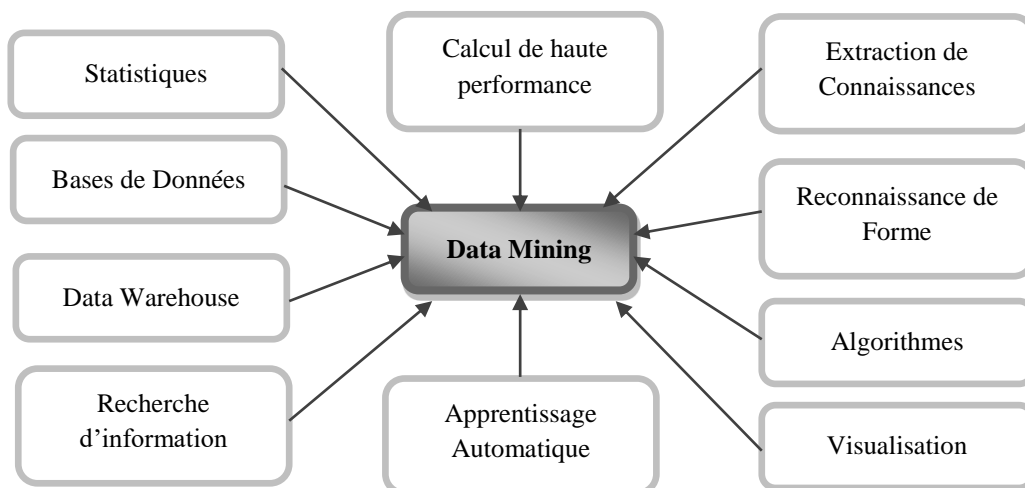


Figure I.1 – La nature interdisciplinaire de la fouille de données

II. L'INTELLIGENCE ARTIFICIELLE

II.1 Définition

L'intelligence artificielle peut se définir comme « l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence » selon le Larousse.

Le but de l'Intelligence Artificielle (IA) est de concevoir des systèmes capables de reproduire le comportement de l'humain dans ses activités de raisonnement. Par exemple, la capacité à interagir avec l'homme, à traiter de grandes quantités de données ou encore à apprendre progressivement et donc à s'améliorer de manière continue. C'est donc un vaste sujet, en perpétuelle évolution !

Soit des ordinateurs ou des programmes **avec des puissances de calcul capables de performances habituellement associées à l'intelligence humaine, et amplifiées par la technologie**, d'où :

- La capacité de raisonner,
- La capacité de traiter de grandes quantités de données,
- La faculté de discerner des patterns et des modèles indétectables par un humain,
- L'aptitude à comprendre et analyser ces modèles,
- Les capacités à interagir avec l'homme,
- La faculté d'apprendre progressivement,
- Et d'améliorer continuellement ses performances.

« **L'intelligence artificielle** » couvre donc un vaste sujet, en perpétuelle mutation. Et aux progrès fulgurants depuis **1950, année fondatrice de l'IA [Micr, 22]**.

II.2 L'intelligence artificielle et les Big data

Littéralement, le terme Big data signifie Mégadonnées, grosses données ou encore données massives. Il désigne un ensemble très volumineux de données. Selon le Gartner, ce concept **regroupe une famille d'outils** qui répondent à une triple problématique dite règle des 3V. Il s'agit notamment d'un **Volume** de données considérable à traiter, une grande **Variété** d'informations (venant de diverses sources, non-structurées, organisées, Open...), et un certain niveau de **Vélocité** à atteindre, autrement dit de fréquence de création, collecte et partage de ces données [Loï, 22].

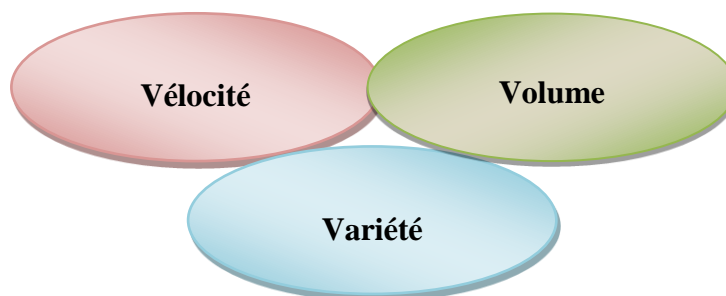


Figure I.2 – Les 3V du Big Data

Nous constatons également que le **Big Data** et l'**IA** sont **intimement liés**, du fait que si nous cherchons un poste dans le secteur de l'intelligence artificielle, nous devons obligatoirement avoir de solides connaissances en gestion de données massives. Ces deux technologies, sont si proches que l'on peut presque parler de **Big Data Intelligence**. L'intelligence artificielle est aujourd'hui présente dans toutes les industries, et la prise de décision est souvent déléguée à des **machines intelligentes**. Sans la data, l'IA ne pourrait exister. L'**automatisation** de ces prises de décisions accentue la convergence entre l'IA et le Big Data, qui ont besoin l'un de l'autre pour avancer et évoluer [YON, 03].

Nous nous sommes servis dans notre travail du Big data fournies par ZINDI comme étant un référentiel d'apprentissage, les détails de cette base énorme ainsi que son utilisation sont présentés dans le dernier chapitre.

II.3 Principaux usages de l'IA

II.3.1 Machine learning (apprentissage automatique)

Nous verrons en détail ce concept dans la section III.

II.3.2 Deep learning (apprentissage profond)

Le Deep Learning ou apprentissage profond : c'est une technique de machine learning reposant sur le modèle des réseaux neurones: des dizaines voire des centaines de couches de neurones sont empilées pour apporter une plus grande complexité à l'établissement des règles.

II.3.3 La robotique

C'est un domaine de l'ingénierie consacré à la conception et à la fabrication de robots. Ces robots servent souvent à exécuter des tâches que les humains ont du mal à réaliser ou à réaliser à la pareille.

La figure ci-dessous montre le lien étroit entre ; Machine learning, Deep learning et IA.

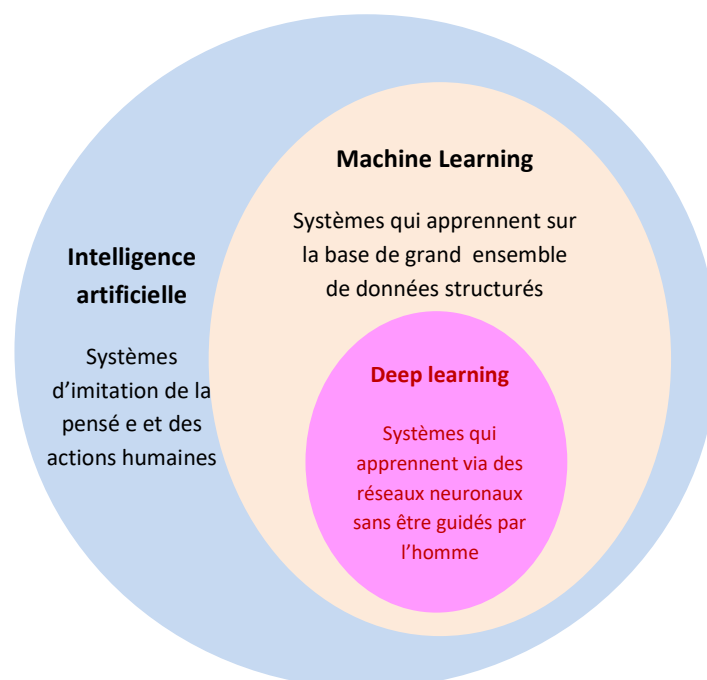


Figure I.3 – Le lien entre Machine Learning Deep Learning et intelligence artificielle.

III. L'APPRENTISSAGE AUTOMATIQUE

III.1 Définition

L'apprentissage automatique (en anglais machine learning) ou apprentissage statistique est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données. Il permet aux ordinateurs d'apprendre sans avoir été programmés explicitement cet effet. Concrètement, il s'agit d'une science moderne permettant de découvrir des patterns et d'effectuer des prédictions à partir de données en se basant sur des statistiques, sur du forage de données, sur la reconnaissance de patterns et sur les analyses prédictives.

Le machine learning est une méthode d'analyse des données qui automatise la création de modèles analytiques. C'est la branche de l'IA qui repose sur l'idée que les systèmes peuvent apprendre des données, identifier des tendances et prendre des décisions avec un minimum d'intervention humaine.

D'une manière plus concrète, examinons la figure 4, elle prend en exemple la caractérisation d'une image en entrée, l'apprentissage sert donc à construire un modèle mathématique afin d'apprendre les différentes variables de celui-ci à partir des données prétraitées.

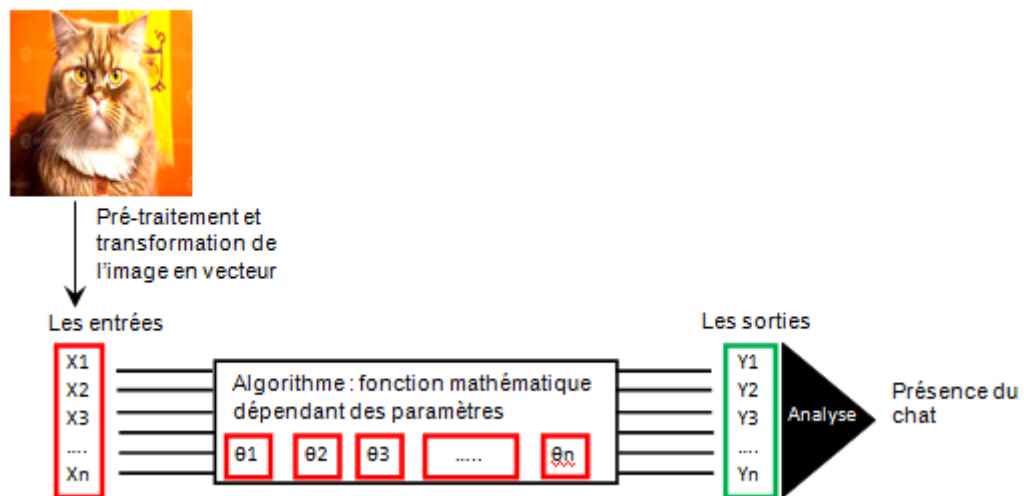


Figure I.4 – Représentation graphique d'un système d'apprentissage automatique avec ses entrées et ses sorties.

III.2 Les pré-requis pour un apprentissage réussi

Comment la machine apprend-t-elle ?

Grâce aux nouvelles technologies informatiques, le machine learning a énormément progressé. Les chercheurs s'intéressant à l'intelligence artificielle voulaient en effet savoir si les ordinateurs étaient capables d'apprendre des données. La dimension itérative du machine learning est importante car les modèles s'adaptent d'eux-mêmes lorsqu'ils sont exposés à de nouvelles données. Ils apprennent de calculs précédents afin de produire des décisions et résultats fiables et reproductibles. La science n'est donc pas nouvelle, mais elle connaît un nouvel élan [Wil & All, 18].

La disponibilité d'une grande masse de données de diverses natures constitue un facteur crucial et un des paliers les plus importants sur lesquels repose l'apprentissage afin de garantir sa qualité. Cependant, il existe d'autres facteurs garantissant ainsi la qualité de tels systèmes, tels que ; le choix de l'algorithme d'apprentissage, méthodes utilisées...nous les verrons plus loin dans le dernier § de cette section.

Nous voulons juste mettre l'accent dans ce paragraphe sur le **lien intrinsèque entre les Big data et machine learning**. En effet, les Big Data sont essentiels au Machine Learning pour apprendre et se développer car les ordinateurs ont toutefois besoin de données à analyser et sur lesquelles s'entraîner. Parallèlement, l'apprentissage automatique est la technologie qui permet d'exploiter pleinement le potentiel du Big Data.

Nous récapitulons les points essentiels pour permettre à la machine d'apprendre :

- Des données qui sont des exemples, relativement abondantes et compilées dans un tableau appelé Dataset à partir desquels la machine va apprendre.
- Des méthodes d'apprentissage fortement inspirées de la façon dont l'être humain apprend les choses. Les méthodes d'apprentissage sont fonction de la nature du problème étudié.
- Un algorithme d'apprentissage qui est la procédure que l'on fait tourner sur les données pour obtenir un modèle, en l'occurrence prédictif. Il existe plusieurs familles d'algorithmes à utiliser selon la nature du problème étudié et la solution la mieux adaptée. Dans le présent chapitre, nous commencerons par introduire les concepts fondamentaux du Machine Learning : définition ; types et les algorithmes les plus utilisés.

III.3 Types d'apprentissage

De nombreuses approches d'apprentissage existent, nous présentons dans la figure I.5 celles les plus répandues.

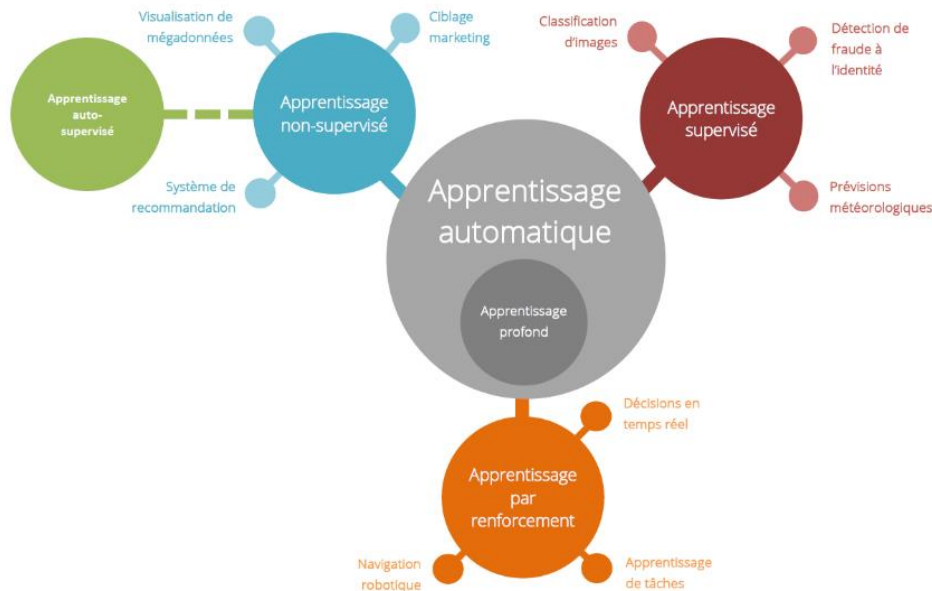


Figure I.5 – Les différents types d'apprentissage et des exemples d'utilisation [Ceo, 19].

III.3.1 L'apprentissage supervisé

L'apprentissage supervisé est une approche de la machine learning qui se définit par l'utilisation d'ensembles de données étiquetées. Ces ensembles de données sont conçus pour former les algorithmes afin qu'ils classent les données ou prédisent les résultats avec précision. En utilisant des entrées et des sorties étiquetées, le modèle peut mesurer sa précision et apprendre au fil du temps.

Avec l'apprentissage supervisé on peut développer des modèles pour résoudre 2 types de problèmes :

- Les problèmes de Régression,
- Les problèmes de classification [Mobi, 21].

III.3.1.1 Classification

Les problèmes de *classification* utilisent un algorithme pour affecter avec précision des données de test à des catégories spécifiques. Les algorithmes d'apprentissage supervisé peuvent être utilisés pour classer les spams dans un dossier distinct de sa boîte de réception par exemple : les classifieurs linéaires, les machines à vecteurs de support, les arbres de décision et les forêts d'arbres décisionnels sont tous des types courants d'algorithmes de classification [Mobi, 21].

III.3.1.2 Régression

La régression est un autre type de méthode d'apprentissage supervisé qui utilise un algorithme pour comprendre la relation entre les variables dépendantes et indépendantes. Les modèles de régression sont utiles pour prédire des valeurs numériques sur la base de différents points de données.

Les algorithmes de régression sont par exemple : la régression linéaire, la régression logistique et la régression polynomiale [Mobi, 21].

III.3.1.3 Fonctionnement de l'apprentissage supervisé

Avec l'apprentissage supervisé, la machine peut apprendre à faire une certaine tâche en étudiant des exemples de cette tâche. Par exemple, elle peut apprendre à reconnaître une photo d'une personne après qu'on lui ait montré des millions de photos de personnes. Ou bien, elle peut apprendre à traduire le français en chinois après avoir vu des millions d'exemples de traduction français-chinois.

D'une manière générale, la machine peut apprendre une **relation** $f: x \rightarrow y$ qui relie x à y en ayant analysé des millions d'exemples d'association $x \rightarrow y$ [GUI, 19].

III.3.1.3.1 Principales étapes du fonctionnement de l'apprentissage supervisé

1. Importer un Dataset (x, y) qui contient nos exemples ;
2. Développer un modèle aux paramètres aléatoires ;
3. Développer une **Fonction Coût** qui mesure les erreurs entre le modèle et le Dataset ;
4. Développer un **Algorithme d'apprentissage** pour trouver les **paramètres** du modèle qui **minimisent** la **Fonction Coût** [GUIL, 19].

A. Le Dataset : exemples de ce qu'il faut apprendre

La première étape d'un algorithme d'apprentissage supervisé consiste donc à importer un Dataset qui contient les exemples que la machine doit étudier.

Ce Dataset inclut toujours 2 types de variables :

- Une variable objectif (**target**) y .
- Une ou plusieurs variables caractéristiques (**features**) x .

B. Modèle

Un modèle est une **représentation** simplifiée de la réalité, que l'on peut utiliser pour **prédire** ce qui se passerait dans certaines conditions. Ça peut être un dessin, une équation physique, une fonction mathématique, une courbe...etc., n'importe quelle représentation [GUIL, 19].

- **La Fonction coût : mesure de la performance**

Pour que la machine trouve le meilleur modèle, il faut déjà qu'elle puisse **mesurer la performance** d'un modèle donné.

- **Un algorithme d'apprentissage**

Il s'agit ici d'utiliser un **Algorithme d'Apprentissage** pour trouver le modèle qui minimise la Fonction Coût.

Cette démarche fonctionne aussi bien pour les problèmes de régression tels que, la prédiction du cours de la bourse, prédire le temps de trajet d'un taxi, etc., que pour les problèmes de classification comme par exemple la détection d'une cellule cancéreuse, tri des emails spam, etc. et même pour les classifications en Deep Learning comme la vision par ordinateur, la reconnaissance vocale, etc. [GUIL, 19].

III.3.2 Apprentissage non supervisé

Il n'y a pas d'étiquetage ou de résultats corrects. La tâche consiste à découvrir la structure des données : par exemple, regrouper des éléments similaires pour former des «grappes», ou réduire les données à un petit nombre de «dimensions» importantes. La visualisation des données peut aussi être considérée comme un apprentissage non supervisé [Min, 23].

III.3.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé se situe entre l'apprentissage supervisé et l'apprentissage non supervisé, et il se distingue par l'intégration des données non labellisées dans l'ensemble d'apprentissage pour entraîner un (ou plusieurs) modèle(s). En d'autres termes, pendant l'entraînement, les données labellisées disponibles sont utilisées, mais les ajustements successifs du modèle sont également fortement influencés par les informations extraites des données non labellisées [AQU, 22].

III.3.4 Apprentissage par renforcement

Dans ce type d'apprentissage, l'algorithme apprend par un mécanisme de rétroaction et des expériences passées. Il est toujours souhaitable que chaque étape de l'algorithme soit effectuée pour atteindre un objectif. Ainsi, chaque fois que la prochaine étape doit être franchie, il reçoit les commentaires de l'étape précédente, ainsi que l'apprentissage de l'expérience pour prédire quelle pourrait être la meilleure étape suivante. Ce processus est également appelé un processus d'essai et d'erreur pour atteindre l'objectif.

III.4 Evaluation des performances des algorithmes d'apprentissage

III.4.1 Evaluation du fractionnement Train-Test

C'est une technique permettant d'évaluer les performances d'un algorithme d'apprentissage automatique qui peut être utilisé pour des problèmes de classification ou de régression et peut être utilisé pour tout algorithme d'apprentissage supervisé. La procédure consiste à prendre un ensemble de données et à le diviser en deux sous-ensembles. Le premier sous-ensemble est utilisé pour ajuster le modèle et est appelé ensemble de données d'apprentissage. Le deuxième sous-ensemble n'est pas utilisé pour entraîner le modèle ; à la place, l'élément d'entrée de l'ensemble de données est fourni au modèle, puis des prédictions sont faites et comparées aux valeurs attendues. Ce deuxième ensemble de données est appelé ensemble de données de test. L'objectif est d'estimer les performances du modèle d'apprentissage automatique sur de nouvelles données : données non utilisées pour entraîner le modèle.

Ceci est le plus souvent exprimé sous forme de pourcentage entre 0 et 1 pour le train ou les jeux de données de test. Par exemple, un ensemble de formation d'une taille de 0,67 (67%) signifie que le pourcentage restant de 0,33 (33%) est attribué à l'ensemble de test. Il n'y a pas de pourcentage de partage optimal. Nous devons choisir un pourcentage partagé qui répond aux objectifs de notre projet avec des considérations qui incluent:

- Coût de calcul de la formation du modèle.
- Coût de calcul lors de l'évaluation du modèle.
- Représentativité de l'ensemble de formation.
- Représentativité de l'ensemble de test. Néanmoins, les pourcentages fractionnés courants comprennent: {Train: 80%, Test: 20% Train: 67%, Test: 33% Train: 50%, Test: 50% }.

III.4.2 Evaluation du modèle avec la validation croisée

La validation croisée k-fold est une méthode répandue pour évaluer les performances des modèles de machine learning. Elle implique de diviser l'ensemble de données en k parties de taille égale, puis d'entraîner le modèle sur k-1 parties tout en le testant sur la partie restante. Cette opération est répétée k fois, en utilisant chaque partie comme ensemble de test une fois. Les résultats obtenus sont ensuite moyennés afin d'obtenir une estimation plus précise de la performance du modèle. La validation croisée k-fold est considérée comme plus fiable que la simple division entraînement/test, car elle utilise plusieurs ensembles de données différents pour évaluer le modèle [Team, 21b].

III.4.3 La matrice de confusion

Dans la plus part des cas d'application, il est très important de connaître la nature des erreurs commises : *Quelle classe est considérée comme quelle classe par le modèle ?*

Par exemple dans un modèle appris pour la détection de fraude, considérer un échantillon {Non-Fraudeur} est beaucoup plus grave de considérer un échantillon {Fraudeur} alors qu'il ne l'est pas. Dans le cas de classification binaire, le résultat de test d'un modèle peut être une possibilité parmi quatre :

- $f(x_i) = 1$ et $y_i = 1$ correcte positive
- $f(x_i) = 0$ et $y_i = 1$ fausse positive
- $f(x_i) = 0$ et $y_i = 0$ correcte négative
- $f(x_i) = 1$ et $y_i = 0$ fausse négative

La matrice de confusion est une matrice qui rassemble en lignes les observations (y) et en colonnes les prédictions f(x). Les éléments de la matrice représentent le nombre d'exemples correspondants à chaque cas [DJE, 14].

Prédictions f(x) \ Observations (y)	Non-Fraudeur (0)	Fraudeur (1)
Non-Fraudeur (0)	VN (Vrai Négatif)	FP (Faux Positif)
Fraudeur (1)	FN (Faux Négatif)	VP (Vrai positif)

Tableau I.1 – Matrice de confusion

Un modèle sans erreurs aura ses résultats rassemblés sur la diagonale de sa matrice de confusion (CP et CN). Dans le cas multi classes la matrice sera plus large avec les classes possibles au lieu des deux classes +1 et -1. La précision P du modèle peut être calculée à partir de la matrice de confusion comme suit :

$$P = \frac{VP+VN}{VP+FN+FP+VN} \quad (\text{III.1})$$

Deux autre mesures sont utilisées : la sensibilité « Sv » et la spécificité « Sp ». La première représente le rapport entre les observations positives correctement prédites et le nombre des observations positives, et la deuxième représente le rapport entre les observations négatives correctement prédites et le nombre total des observations négatives.

$$\left\{ \begin{array}{l} S_v = \frac{VP}{VP+FN} \\ S_p = \frac{VN}{VN+FP} \end{array} \right. \quad (\text{III.2})$$

L'outil d'évaluation graphique le plus utilisé dans la littérature est la Courbe ROC (receiver operating characteristic) et le calcul de l'AUC (Area Under Curve). Une courbe ROC est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs (TVP) en fonction du taux de faux positifs (TFP). Où :

- Le TVP est l'équivalent au Rappel. Il est donc défini comme suit :

$$\text{TVP} = \frac{VP}{VP+FN} \quad (\text{III.3})$$

- Le TFP est défini comme suit :

$$\text{TFP} = \frac{FP}{FP+VN} \quad (\text{III.4})$$

III.5 Quelques familles d'algorithmes d'apprentissage

De nombreuses méthodes d'apprentissage automatique sont utilisées que ce soit pour faire de la classification ou de la régression. Nous détaillons quelques-unes dans les sections suivantes.

III.5.1 Algorithme de classification (classification)

a). Naïve Bayes//réseaux bayésiens naïfs

La classification naïve bayésienne est un type de classification basée sur le théorème de Bayes, caractérisée par une forte indépendance (appelée naïveté) des hypothèses. Elle utilise un classifieur bayésien naïf, qui est un type de classifieur linéaire. On peut également qualifier le modèle probabiliste sous-jacent de "modèle à caractéristiques statistiquement indépendantes". Les classifieurs bayésiens naïfs peuvent être efficacement entraînés dans un contexte d'apprentissage supervisé, selon la nature de chaque modèle probabiliste.

Les réseaux bayésiens naïfs, tels que le naïve Bayes, sont une forme simplifiée de ces réseaux, qui se révèle particulièrement efficace pour les tâches d'apprentissage et d'inférence. Parmi les différentes méthodes proposées pour améliorer le réseau bayésien naïf, on trouve le TANB (Tree Augmented Naive Bayes), qui utilise une structure naïve. L'algorithme naïf de Bayes est également utilisé pour la classification. De plus, il est possible de travailler avec le modèle bayésien naïf sans se préoccuper des probabilités bayésiennes ou d'utiliser des méthodes bayésiennes. [Wiki, 22]

b). Arbres de décision

L'arbre de décision est un type d'algorithme d'apprentissage automatique utilisé pour la classification et la régression. Il est utilisé pour prendre des décisions en suivant une série de règles logiques basées sur les caractéristiques d'un ensemble de données [BEL, 22].

c). Forêt aléatoire (Random Forest)

Random Forest est un algorithme d'apprentissage automatique extrêmement populaire, largement utilisé pour les tâches de classification et de régression. Il s'agit d'une méthode d'apprentissage ensembliste qui combine plusieurs arbres de décision afin d'obtenir des prédictions plus précises.

Dans une forêt aléatoire, chaque arbre de décision est entraîné sur un sous-ensemble de données et un sous-ensemble de caractéristiques. La prédiction finale est obtenue en prenant la moyenne des prédictions de tous les arbres individuels. Cette approche contribue à réduire le surajustement et à améliorer les performances globales du modèle [HEL, 23].

d). SVM (Machine à vecteurs de support)

La machine à vecteurs de support (SVM) est un type d'algorithme d'apprentissage automatique utilisé pour la classification et la régression. Elle est utilisée pour trouver la meilleure ligne ou courbe qui sépare les données en différentes classes [Kha, 05].

e). KNN

L'algorithme des k plus proches voisins (KNN) est une méthode de classification des données permettant d'estimer la probabilité qu'un point de données devienne membre d'un groupe ou d'un autre en se basant sur le groupe auquel appartiennent les points de données les plus proches.

L'algorithme des k plus proches voisins est un type d'algorithme d'apprentissage automatique supervisé utilisé pour résoudre des problèmes de classification et de régression. Cependant, il est principalement utilisé pour les problèmes de classification. [JOB, 21]

f). Boosting

Le boosting est une méthode de machine learning visant à réduire les erreurs dans l'analyse prédictive des données. En utilisant des données étiquetées, les scientifiques des données entraînent des modèles de machine learning pour effectuer des prédictions sur des données non étiquetées. Cependant, un modèle unique peut commettre des erreurs en fonction de la qualité des données d'entraînement. Par exemple, un modèle d'identification de chats qui n'a été entraîné qu'avec des images de chats blancs peut se tromper lorsqu'il s'agit d'identifier un chat noir. Le boosting résout ce problème en entraînant successivement plusieurs modèles afin d'améliorer la précision globale du système [AWS, 22].

g). Les réseaux de neurones

Les réseaux de neurones sont des modèles d'apprentissage automatique qui sont inspirés du fonctionnement du cerveau humain. Ils sont composés de plusieurs couches de "neurones" artificiels qui sont connectés les uns aux autres pour former un réseau. Les réseaux de neurones sont utilisés pour la classification, la régression, la reconnaissance d'images, la reconnaissance vocale, et bien plus encore [Che, 23].

h). Modèle de Markov caché

Un modèle de Markov caché, également connu sous le nom de HMM (Hidden Markov Model), est un processus à double stochastique dans lequel l'une des composantes est une chaîne de Markov non observable. Ce processus peut être observé à travers un autre ensemble de processus qui génère une séquence d'observations.

De manière plus simple, il s'agit d'un modèle qui décrit les états d'un processus markovien en utilisant les probabilités de transition et les probabilités d'observation associées à ces états [KIM, 20].

i). **Modèle de mélange gaussien**

Un modèle de mélange de gaussiennes est une approche qui suppose que les données ont été générées selon le processus suivant : pour chaque échantillon de données, nous choisissons un entier k parmi un ensemble de K valeurs possibles, chaque valeur ayant une certaine probabilité associée. Ensuite, nous générons l'échantillon x_n à partir d'une loi de probabilité spécifique correspondant à la valeur k choisie. En d'autres termes, les données d'entrée sont des échantillons provenant de l'une des K différentes lois gaussiennes, chaque loi ayant ses propres moyennes et covariances distinctes [Pip, 21a].

III.5.2 Algorithmes de régression (regression)

a). **Régression linéaire**

La régression linéaire est une technique statistique utilisée pour trouver la relation entre deux variables continues. Elle est utilisée pour prédire la valeur d'une variable à partir de la valeur d'une autre variable en utilisant une formule mathématique qui représente la relation entre les deux variables [Spi, 23].

b). **Régression polynomiale**

La régression polynomiale est une extension d'un modèle de régression linéaire classique. La régression polynomiale modélise la relation non linéaire entre une variable prédictrice et une variable de résultat en utilisant un polynôme de degré N de la variable prédictrice [SPA, 23].

c). **Lasso régression**

La régression LASSO est une méthode utilisée pour surmonter les limitations de la régression linéaire dans les situations où la dimension des données est élevée. Elle vise à remédier aux problèmes d'estimation instable et de prévision peu fiable associés à la régression linéaire [Pit, 21b].

d). **Régression logistique**

La régression logistique est un modèle statistique qui étudie les relations entre un ensemble de variables qualitatives X_i et une variable qualitative Y . Il s'agit d'un modèle linéaire généralisé qui utilise une fonction logistique comme fonction de liaison.

Un modèle de régression logistique permet également de prédire la probabilité qu'un événement se produise (valeur de 1) ou non (valeur de 0) en optimisant les coefficients de régression. Ce résultat varie toujours entre 0 et 1. Lorsque la valeur prédite dépasse un seuil, l'événement est susceptible de se produire, tandis que lorsqu'elle est inférieure au même seuil, il ne l'est pas [RED, 22].

e). **Régression multivariée**

La régression multivariée est une technique statistique utilisée pour mesurer les relations linéaires entre plusieurs variables indépendantes et dépendantes dans un ensemble de données. Elle permet de prédire le comportement d'une variable de réponse en fonction de variables prédictives correspondantes. Cette méthode est souvent utilisée comme algorithme supervisé en apprentissage automatique pour prédire le comportement de multiples variables indépendantes et dépendantes dans un modèle [VOX, 23].

f). Algorithme de régression multiple

La régression linéaire multiple est une extension de la régression linéaire simple qui utilise plusieurs variables explicatives (indépendantes) pour prédire une variable de résultat (expliquée). La variable de résultat est toujours continue, tandis que les variables explicatives peuvent être continues ou catégorielles. L'objectif est similaire à celui de la régression linéaire simple : réaliser des prédictions.

Par exemple, une régression linéaire multiple peut être utilisée pour prédire le niveau de vente d'un produit en fonction des caractéristiques des acheteurs telles que l'âge, le niveau de salaire, l'adresse, etc [CDA, 22].

III.5.3 Algorithme de catégorisation (clustering)

a). L'algorithme K-Means

Est parfaitement indiqué pour faire un tel regroupement. Cet algorithme d'apprentissage automatique non supervisé permet à partir d'un ensemble de données et de K groupes, de segmenter les différents éléments en ce même nombre de groupes. Il effectue ce regroupement en minimisant la distance euclidienne entre le centre du cluster et un objet donné [Dab, 22].

b). Fuzzy C-means Algorithm

La méthode de regroupement C-means (C-means clustering), ou regroupement flou C-means (fuzzy c-means clustering), est une technique de regroupement doux en apprentissage automatique dans laquelle chaque point de données est séparé en différents groupes, puis se voit attribuer un score de probabilité d'appartenance à ce groupe. Le regroupement flou C-means donne de meilleurs résultats pour les ensembles de données superposés par rapport au regroupement k-means [KUM, 22].

c). Le clustering hiérarchique ou encore la décomposition en valeurs singulières (SVD)

Le clustering hiérarchique est un algorithme de regroupement qui crée une structure en forme d'arbre appelée dendrogramme. Il existe deux types de clustering hiérarchique : la classification descendante hiérarchique et la classification ascendante hiérarchique. Dans la classification descendante, tous les points sont initialement regroupés ensemble, puis séparés en clusters jusqu'à ce qu'il y ait un cluster pour chaque point. Dans la classification ascendante, chaque point est initialement considéré comme un groupe séparé, puis des paires de clusters sont combinées en fonction de leurs similarités pour former un grand groupe contenant toutes les observations.

Tout comme la méthode des K-moyennes, les mesures de distances sont utilisées pour évaluer la similarité entre les points [ALL, 22].

d). Algorithme Espérance-maximisation

L'algorithme espérance-maximisation (EM) est un algorithme itératif couramment utilisé dans divers domaines tels que l'apprentissage automatique et l'imagerie médicale. Il permet d'estimer les paramètres optimaux d'un modèle probabiliste en présence de variables latentes non observables. L'algorithme EM consiste en deux étapes, l'étape d'espérance et l'étape de maximisation, qui sont répétées jusqu'à atteindre la convergence. Il est utilisé pour la classification, le clustering et la segmentation des données, permettant ainsi de découvrir des structures cachées et d'améliorer la compréhension des données [Wiki, 23].

IV. PRESENTATION DU PROCESSUS DE L'EXTRACTION DE CONNAISSANCE (ECD)

Le processus d'extraction des connaissances dans les bases de données (ECD). Présenté sur la **figure I.6** désigne l'ensemble des opérations qui permettent d'exploiter avec facilité et rapidité des données stockées massivement. Il s'agit d'un processus non trivial, consistant à identifier dans les données des schémas nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables [FAY & All, 96a].

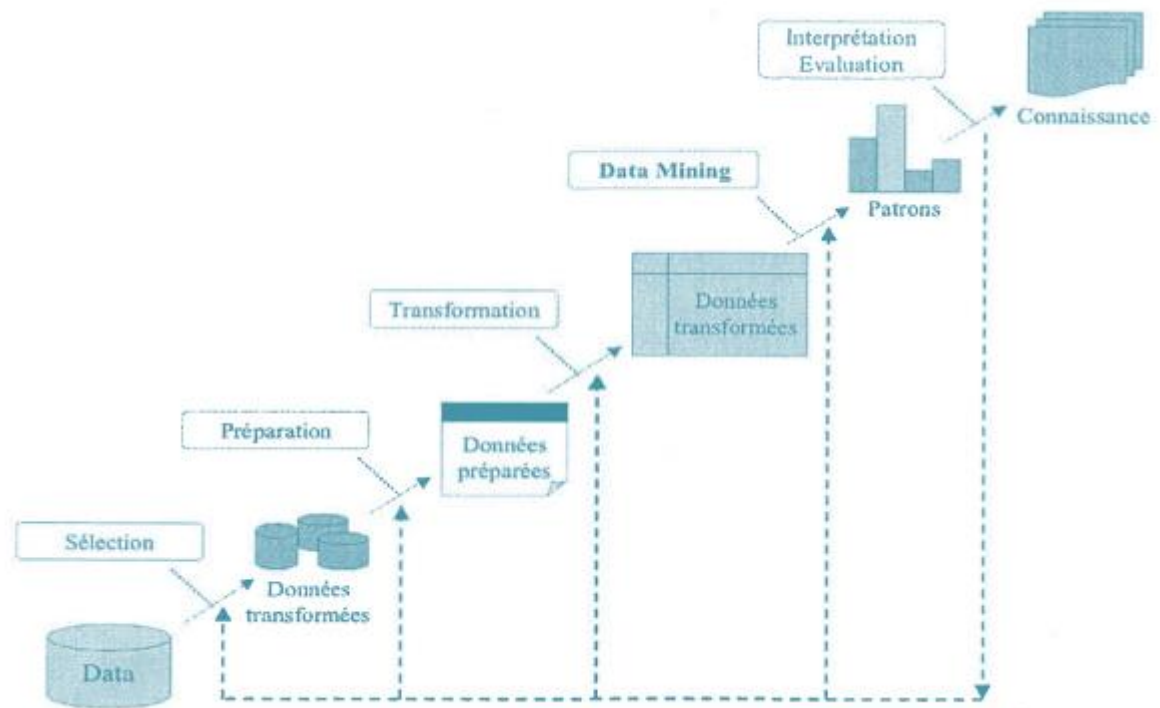


Figure I.6 – Les différentes étapes du processus d'ECD [Jol, 03].

Le processus d'ECD peut avoir deux objectifs, soit vérifier les hypothèses d'un utilisateur, soit découvrir de nouveaux motifs. Un motif, ou schéma, est une expression dans un langage spécifique qui décrit un sous-ensemble de données ou un modèle applicable à ce sous-ensemble [FAY & All, 96b].

IV.1 Les étapes d'un processus d'ECD

Ce processus comporte quatre étapes principales :

- Nettoyage et intégration des données,
- La préparation des données,
- La fouille de données (data mining),
- L'interprétation.

IV.1.1 Nettoyage et intégration des données

Le nettoyage des données consiste à traiter ces données bruitées, soit en les supprimant, soit en les modifiant de manière à tirer le meilleur profit. L'intégration est la combinaison des données provenant de plusieurs sources (base de données, sources externes, etc.). Le but de ces deux opérations est de générer des entrepôts de données et/ou des magasins de données spécialisés contenant les données traitées pour faciliter leurs exploitations futurs.

IV.1.2 Le prétraitement des données

Il peut arriver parfois que les bases de données contiennent à ce niveau un certain nombre de données incomplètes et/ou bruitées. Ces données erronées, manquantes ou inconsistantes doivent être traitées si cela n'a pas été fait précédemment. Dans le cas contraire, durant l'étape précédente, les données sont stockées dans un entrepôt. Cette étape permet de sélectionner et transformer des données de manière à les rendre exploitables par un outil de fouille de données. Cette seconde étape du processus d'ECD permet d'affiner les données. Si l'entrepôt de données est bien construit, le prétraitement de données peut permettre d'améliorer les résultats lors de l'interrogation dans la phase de fouille de données.

IV.1.3 Fouille de données (Data Mining)

Nous avons déjà introduit ce concept dans la section I et nous n'allons pas le détailler dans cette section car il en fera le sujet d'une autre plus loin dans ce rapport (*section V*).

IV.1.4 Evaluation et présentation

Cette phase mesure l'intérêt des motifs extraits, et de la présentation des résultats à l'utilisateur grâce à différentes techniques de visualisation. Cette étape est dépendante de la tâche de fouille de données employée. En effet, bien que l'interaction avec l'expert soit importante quelle que soit cette tâche, les techniques ne sont pas les mêmes. Ce n'est qu'à partir de la phase de présentation que l'on peut employer le terme de connaissance à condition que ces motifs soient validés par les experts du domaine. Il y a principalement deux techniques de validation qui sont la technique de *validation statistique* qui à utiliser des méthodes de base de statistique descriptive. L'objectif est d'obtenir des informations qui permettront de juger le résultat obtenu, ou d'estimer la qualité ou les biais des données d'apprentissage. Cette validation peut être obtenue par : (i). Le calcul des moyennes et variantes des attributs – (ii). Si possible, le calcul de la corrélation entre certains champs – (iii). Ou la détermination de la classe majoritaire dans le cas de la classification. Quant à *la validation par expertise*, elle est réalisée par un expert du domaine qui jugera la pertinence des résultats produits. Par exemple, pour la recherche des règles d'association, c'est l'expert du domaine qui jugera la pertinence des règles. Pour certains domaines d'application (le diagnostic médical, par exemple), le modèle présenté doit être compréhensible. Une première validation doit être effectuée par un expert qui juge la compréhensibilité du modèle. Cette validation peut être, éventuellement, accompagnée par une technique statistique.

Grâce aux techniques d'extraction de connaissances, les bases de données volumineuses sont devenues des sources riches et fiables pour la génération et la validation de connaissances.

V. FOUILLE DE DONNEES

Les concepts de fouille de données et l'extraction de connaissances à partir de données sont parfois confondus et considérés comme synonymes. Mais, formellement on considère la fouille de données comme une étape centrale du processus d'extraction de connaissances des bases de données (ECBD ou KDD pour Knowledge Discovery in Databases en anglais) [Lie, 07].

V.1 Quelques définitions

Le data mining, ou fouille de données, est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de bases de données informatiques (souvent grandes), de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de données [KAN, 03].

D'après [Had, 02], la définition la plus communément admise de Data Mining est celle de [FAY & All, 98] : « Le Data Mining est un processus non trivial qui consiste à identifier, dans des données, des schémas nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables ».

En bref, le data mining est l'art d'extraire des informations (ou mêmes des connaissances) à partir des données [TUF, 02].

V.2 Historique

L'expression "data mining" est apparue vers le début des années 1960 et avait, à cette époque, un sens péjoratif. En effet, les ordinateurs étaient de plus en plus utilisés pour toutes sortes de calculs qu'il n'était pas envisageable d'effectuer manuellement jusque-là. Certains chercheurs ont commencé à traiter sans a priori statistique les tableaux de données relatifs à des enquêtes ou des expériences dont ils disposaient. Comme ils constataient que les résultats obtenus, loin d'être aberrants, étaient tout au contraire prometteurs, ils furent incités à systématiser cette approche opportuniste. Les statisticiens officiels considéraient toutefois cette démarche comme peu scientifique et utilisèrent alors les termes "data mining" ou "data fishing" pour les critiquer. Cette attitude opportuniste face aux données coïncida avec la diffusion dans le grand public de l'analyse de données dont les promoteurs, comme Jean-Paul Benzecri [Zig & All, 00], ont également dû subir dans les premiers temps les critiques venant des membres de la communauté des statisticiens.

Le succès de cette démarche empirique ne s'est pas démenti malgré tout. L'analyse des données s'est développée et son intérêt grandissait en même temps que la taille des bases de données. Vers la fin des années 1980, des chercheurs en base de données, tel que Rakesh Agrawal [Swa & All, 93], ont commencé à travailler sur l'exploitation du contenu des bases de données volumineuses comme par exemple celles des tickets de caisses de grandes surfaces, convaincus de pouvoir valoriser ces masses

de données dormantes. Ils utilisèrent l'expression "database mining" mais, celle-ci étant déjà déposée par une entreprise (Database mining workstation), ce fut "data mining" qui s'imposa.

En mars 1989, Shapiro Piatetski [PIA, 91] proposa le terme "knowledge discovery" à l'occasion d'un atelier sur la découverte des connaissances dans les bases de données. Actuellement, les termes data mining et knowledge discovery in data bases (KDD, ou ECD en français) sont utilisés plus ou moins indifféremment. Nous emploierons par conséquent l'expression "data mining", celle-ci étant la plus fréquemment employée dans la littérature. La communauté de "data mining" a initié sa première conférence en 1995 à la suite de nombreux ateliers (workshops) sur le KDD entre 1989 et 1994. La première revue du domaine "Datamining and knowledge discovery journal" publiée par "Kluwers" a été lancée en 1997.

V.3 Tâches du Data mining

De nombreuses tâches peuvent être associées au Data Mining, parmi elles nous pouvons citer:

V.3.1 Classification

Pour faire une classification il suffit d'étudier les caractéristiques d'un nouvel objet afin de lui attribuer une classe prédéfinie. Les objets à classer sont généralement des enregistrements d'une base de données, cette classification va permettre de mettre à jour chaque enregistrement en déterminant un champ de classe. La tâche de classification consiste à définir une classe assez précise et un ensemble d'exemples classés auparavant. Le but est la création d'un modèle qui peut être appliqué aux données non classifiées afin de les classifiées [BER & Ail, 00]. Nous citons quelques exemples de tâche de classification

- Diagnostiquer la possibilité de l'existence d'une maladie.
- Déterminer si l'utilisation d'une carte de crédit est frauduleuse [LAR, 05].

V.3.2 Estimation

A partir des caractéristiques d'un objet, on peut arriver à faire l'estimation d'un champ en ce qui concerne sa valeur. Cette opération d'estimation peut se faire dans un but essentiel de classification. Il faut pour cela lui donner une classe particulière à un intervalle de valeurs d'un champ estimé. Par exemple : la classification peut concerner des événements discrets (le patient a été ou non hospitalisé). L'estimation peut alors se baser à des variables contenues à savoir la durée d'hospitalisation [BEL, 11]. Voici quelques exemples :

- Estimer le nombre d'enfants dans une famille [BER & Ail, 04].
- Estimer le montant d'argent qu'une famille de quatre membres choisis aléatoirement dépensera pour la rentrée scolaire [LAR, 05].

V.3.3 La prédiction

La classification et l'estimation tous deux ressemblent à la prédiction. Cependant, cela se fait dans une échelle temporelle différente comme nous l'avons vu dans les tâches précédentes, tout s'appuie sur le passé et présent. Il y a seul le résultat qui appartient dans un futur à préciser.

Parmi les techniques les plus appropriées à la prédiction sont [LOH, 11] :

- L'analyse du panier de la ménagère (ou règles d'association) ;

- Le raisonnement basé sur la mémoire ;
- Les arbres de décision ;
- Les réseaux de neurones

Nous citons ci-dessous quelques exemples de tâche prédiction :

- Prédire au vu de leurs actions passées les départs de clients.
- Prévoir le champion de la coupe du monde en football en se basant sur la comparaison des statistiques des équipes [BEL, 11].

V.3.4 Règle d'association

Grouper par similitude est une tâche d'association. Cela va permettre de déterminer d'avance les attributs qui vont ensemble. Le data mining a pour fonction de donner un sens aux données, il faut pour cela en extraire les relations masquées et non triviales à utiliser la base de données. La technique la plus recommandée est celle qui consiste au regroupement par similitudes en faisant l'analyse de panier de la ménagère [BEN, 13].

Voici un exemple de tâche d'association :

- Trouver dans un supermarché quels produits sont achetés ensemble et quels sont ceux qui ne s'achètent jamais ensemble.
- Déterminer la proportion des cas dans lesquels un nouveau médicament peut générer des effets dangereux [LAR, 05].

V.3.5 La segmentation

Il faut démontrer et arriver à trouver les observations qui s'associent sans pour cela ne privilégier aucune variable. On partage une certaine population hétérogène en sous-groupes pour le rendre plus homogènes (clusters). A ce stade, les classes n'ont pas été définies. Ici la technique la plus appropriée à la segmentation est l'analyse des clusters [HOU, 07].

V.3.6 Description

Data mining est des fois utilisé pour décrire ce qu'il y a sur une base de donnée complexe pour expliquer les relations qui existent dans les données pour la bonne compréhension des individus , des produits et des processus existants sur cette base . Une bonne description d'un comportement implique souvent une bonne explication de celui-ci. Ici, la technique la plus appropriée à la description est l'analyse du panier de la ménagère [AGA & All, 05].

V.4 Méthodes du data mining

Pour tout jeu de données et un problème spécifique, il existe plusieurs méthodes que l'on choisira en fonction de :

- La tâche à résoudre ;
- La nature et de la disponibilité des données.
- L'ensemble des connaissances et des compétences disponibles ;
- La finalité du modèle construit ;

- L'environnement social, technique, philosophique de l'entreprise ;
- Etc.

On peut dégager deux grandes catégories de méthodes d'analyse consacrées à la fouille de données [Fio06]. La frontière entre les deux définie par la spécificité des techniques, et marque l'aire proprement dite du « Data Mining ». On distingue donc :

A. Les méthodes classiques

On y retrouve des outils généralistes de l'informatique ou des mathématiques :

- Les requêtes dans les bases de données, simples ou multicritères, dont la représentation est une vue,
 - les requêtes d'analyse croisée, représentées par des tableaux croisés,
 - les différents graphes, graphiques et représentations,
 - les statistiques descriptives,
 - l'analyse de données : analyse en composantes principales,
 - Etc

B. Les méthodes sophistiquées

Elles ont été élaborées pour résoudre des tâches bien définies. Ce sont :

- Les algorithmes de segmentation,
- les règles d'association,
- les algorithmes de recherche du plus proche voisin,
- les arbres de décision,
- les réseaux de neurones,
- les algorithmes génétiques,
- Etc.

V.5 Domaine d'application du Data Mining

Le data mining est une spécialité transverse, elle regroupe un ensemble de théories et d'algorithmes ouverts à tout domaine susceptible de drainer une masse importante de données.

Parmi ces domaines on cite [MEN, 09]:

V.5.1 Le secteur bancaire

- Identifier les clients « fidèles ».
- Identifier les clients qui seront les plus réceptifs aux nouvelles offres de produits.
- Prédire les clients qui sont susceptibles de changer leurs cartes d'affiliation au cours du prochain trimestre.

V.5.2 La détection de fraude

La fouille de données est largement appliquée dans des processus de détection de fraude divers tel que :

- Détection de fraude dans la consommation d'électricité et du gaz.
- Détection de fraude de cartes de crédits.
- Détection de fausses demandes de remboursement médicale.

V.5.3 Le secteur des assurances

- Evaluation du risque d'un bien assuré prenant en compte les caractéristiques du bien et de son propriétaire.
- Formulation des modèles statistiques des risques d'assurance.

V.5.4 La médecine

- Prédiction de présence de maladies.
- Approvisionnement des médicaments les plus fréquemment prescrits [CHA, 10].

VI. APPRENTISSAGE AUTOMATIQUE DANS LA DETECTION DE FRAUDES

La détection et la prévention des données anormales représentent une des tâches primordiales en apprentissage automatique et pour la fouille des données.

Dans le contexte de la consommation d'énergie tels que le gaz et l'électricité, ces données reflètent souvent des comportements inhabituels (**anomalies**) de consommation. Une anomalie peut être définie de manières différentes, par exemple, par le rapport de la consommation de l'électricité dans les jours fériés et les autres, les saisons estivales et autres...etc.

Dans le cadre de notre travail, une consommation est qualifiée d'anormale lorsqu'elle comparée aux données historiques et/ou après l'avoir constatée sur terrain.

[GAU & AIL, 19] et [BEN, 18] ont globalement classé les anomalies en trois types : (i). les anomalies ponctuelles, elles sont considérées lorsqu'une observation individuelle est considérée comme anormale par rapport au reste des données. (ii). Les anomalies collectives, elles sont définies lorsqu'une séquence d'observations est anormale par rapport au reste des données. (iii). Une anomalie contextuelle est constatée lorsqu'une observation est considérée comme normale par rapport à un contexte mais pas dans un autre contexte.

La mise en œuvre des algorithmes d'apprentissages permettent d'analyser et de comprendre le comportement des actions à travers les données récupérées et d'extraire des connaissances utiles relatives aux caractéristiques de ces comportements. Nous pouvons par la suite mettre en application des règles afin de bloquer ou d'autoriser certaines actions des consommateurs, ainsi, nous signalons les cas antérieurs des fraudes, non-fraudes afin d'éviter les faux positifs et améliorer nos règles de précision.

VII. CONCLUSION

La convergence de l'informatique et de la communication a produit une société qui se nourrit sur les informations. Si nous caractérisons les données comme des faits enregistrés, alors l'information est l'ensemble des modèles, ou des attentes, qui sous-tendent les données. Notre travail était de faire émerger ces données à travers des outils bien déterminés.

Nous avons présenté dans ce chapitre une vision multidimensionnelle de la fouille de données. Ses dimensions étant : les données, connaissances, techniques d'extraction et des algorithmes pouvant explorer efficacement ces bases de données et créer des modèles d'apprentissage.

La fouille de données est l'extraction d'éléments implicites, et potentiellement utiles à partir des données. L'idée est de construire des programmes informatiques qui font un filtrage automatique -par le biais de bases de données- et recherchent des régularités ou des modèles. Ces modèles se généraliseront par la suite pour faire des prédictions précises sur les données futures. Cependant, il peut probablement y avoir des problèmes, tels que ; la création de modèles qui peuvent être sans intérêt, d'autres peuvent dépendre de coïncidences accidentelles dans l'ensemble de données utilisé, données réelles imparfaites où certaines parties seront tronquées et d'autres manquantes, et par conséquent, out ce qu'est découvert sera probablement inexact car il y aura des exceptions à chaque règle et des cas non couverts par aucune règle. A partir de ce constat, il faut choisir des algorithmes qui doivent être suffisamment robustes pour faire face à ces imperfections et créer des modèles d'apprentissage utiles.

Un vocabulaire de l'apprentissage automatique (machine learning) a été également décrit dans ce chapitre, cette discipline est relativement récente mais en perpétuelle mutation. Une technologie en plein essor qui a bénéficié d'une attention assez intense de la part des chercheurs.

Dans le cadre de notre travail, nous avons varié deux techniques d'algorithmes qui font partie de la famille d'algorithmes de boosting.

Dans le chapitre qui suit, nous proposons une description détaillée de cette famille d'algorithmes. Ces techniques seront utilisées et appliquées sur des données des consommations issues des points de mesure effectuées par l'organisme où nous avons effectué notre stage.

QUELQUES

CHAPITRE

APPLICATIONS DES
TECHNIQUES

II

D'APPRENTISSAGE
APERÇU D'UN ETAT
DE L'ART

II

Applications des techniques d'apprentissage : Aperçu de l'Etat de l'Art

Sommaire

I INTRODUCTION.....	28
II QUELQUES TRAVAUX CONNEXES	28
II.1 Apprentissage en profondeur	28
II.2 Apprentissage supervisé	29
III SYNTHÈSE.....	32
IV CONCLUSION	33

I. INTRODUCTION

Nous avons consacré cette partie du mémoire pour faire un survol sur les ouvrages antérieurs traitant les problématiques d'extraction de connaissances à partir d'une base de données, utilisant le machine learning.

Il est évident que nous ne pouvons pas présenter une liste exhaustive de tous les travaux effectués qui adoptent ces approches. Seulement, nous allons citer dans ce qui suit, quelques contributions qui ont retenu notre attention classées selon le type d'apprentissage

II. QUELQUES TRAVAUX CONNEXES

D'après nos recherches dans la littérature, nous avons jugé nécessaire de présenter cette liste de travaux connexes relatifs au domaine du machine learning et apprentissage par leurs classes de types

II.1 Apprentissage en profondeur

[MER & Ail, 17] ont réalisé et testé un modèle pour la détection des maladies avec le traitement d'image. Ils ont implémenté le réseau de neurones convolutés.. Ils ont utilisé un benchmark de la base d'images *chest-xray-pneumonia* et à travers les divers tests ils ont obtenu deux graphes : (i) un Graphe de précision et (ii) un Graphe d'erreur.

[BEN, 19] a proposé une nouvelle approche basée sur l'apprentissage en profondeur « Deep Learning » semi-supervisée évalué et testé sur la base de données « Brest-cancer-Wisconsin ». Pour la prédiction du cancer du sein. L'approche proposée DBNALRBM (Deep belief network all layer RBM) a été implémentée dans l'environnement de développement *spyder* avec l'utilisation des bibliothèques *tkinter* et *flask* sous *python*. D'autres maladies ont été prédites par ce modèle telles que, le diabète, les maladies cardiaques et la Covid-19. Ce modèle a donné des résultats prometteurs et très encourageants (diabète 60.43, maladie cardiaque 97.16, covid-19 50.46).

DBRCNN est un système de compression d'images basé sur les réseaux de neurones profonds qui a été mis en place par [KER, 20]. Il est capable de générer des résultats raisonnables à partir des images dégradées (brouillées) d'entrée. Ce modèle a été développé sous *python* avec les bibliothèques *TensorFlow* et *Keras* dans l'environnement *Google Colab* sur le dataset *UTKFace* qui se compose de plus de 20000 images de visages avec des annotations d'âge, de sexe et d'ethnicité. A travers les résultats obtenus, les auteurs ont conclu que cette approche peut être utilisée comme une étape de prétraitement pour différentes tâches de vision par ordinateur dans le cas où les images sont dégradées par la compression à un point tel que les algorithmes de pointe échouent.

[OUK & Ail, 20] ont développé un modèle avec *python* sous l'environnement de développement *anaconda* pour la détection de la température et du rythme respiratoire à distance basée sur l'intelligence artificielle (apprentissage profond) pour la prévention contre la pandémie du COVID-19. Le but de ce projet est de réaliser un système intelligent capable de mesurer à distance la température corporelle et d'analyser le rythme respiratoire en surveillant les changements de température autour de la zone nasale en utilisant les algorithmes de détection de masque et du *Face Landmark*. La base de données utilisée est recueillie auprès des sources (*Kaggle datasets* et *RMFD dataset*), cet ensemble de données se compose de 4095 images appartenant à deux classes : avec masque (2165 images) et sans masque (1930 images). Les résultats obtenus ont été satisfaisants et encourageants.

Un « **Système de prédiction de la consommation d'énergie basé Deep Learning** » est la proposition de [DJA,21], c'est un modèle de prédiction de la consommation d'énergie hybride, qui combine entre les réseaux de neurones à convolution (CNN) et l'un le plus utilisé des méthodes de réseaux de neurone récurrent les réseaux à mémoire longue à court terme (LSTM). Le modèle hybride CNN-LSTM a été évalué et a prouvé son efficacité en utilisant la base de données IHEPC. Le modèle est formé sur Google Colab avec un moteur de calcul Google Python 3 sur le navigateur Google Chrome.

L'auteur a utilisé les données de test pour obtenir des données prédites, puis comparé les résultats avec les données réelles à l'aide des mesures de performances. Pour la validation du modèle, il a utilisé plusieurs mesures de performance, telles que MSE, MAE, RMSE (0.009, 0.096, 0.072).

Le projet intitulé « **Une approche d'optimisation pour une meilleure efficacité d'un modèle d'estimation de temps restant utile du moteur à double flux à base de deep learning** » est réalisé par [HAS & All, 21] et qui a pour objectif l'estimation de RUL en utilisant le deep learning (apprentissage en profondeur) et la recherche de la meilleure combinaison d'hyperparamètres à l'aide d'une méthode d'optimisation aboutissant à une meilleure performance du modèle. Pour atteindre cet objectif, ils ont utilisé LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Unit) comme modèle de deep learning et l'algorithme génétique comme méthode d'optimisation des hyperparamètres. Le modèle proposé a été entraîné, testé et validé sur l'ensemble de données CMAPSS (Commercial Modular Aero-Propulsion System Simulation) de la NASA. Pour ce faire, ils ont utilisé python comme langage de programmation, Anaconda comme environnement de programmation et Spyder comme outil de programmation. Les résultats de cette proposition ont justifié l'efficacité du modèle LSTM, de l'algorithme génétique ainsi que ses paramètres choisis.

II.2 Apprentissage supervisé

Les auteurs [SCH & All, 01], ont mis en place le premier système de détection de malwares basé sur des techniques d'apprentissage automatique. Les auteurs ont étudié différentes informations contenues dans le fichier PE tels que des chaînes de caractères, les APIs, et la séquence d'octets. Ils ont utilisé une méthode de classification basée sur un algorithme Bayésien naïf, et ils ont obtenu une précision globale de 97,11% en utilisant les chaînes de caractères comme attributs.

[BOU & All, 19] ont présenté un système de reconnaissance d'activité en utilisant l'apprentissage supervisé à partir d'un ensemble de données contenant des enregistrements de signaux d'accéléromètre et gyroscope intégrés dans les Smartphones de différents utilisateurs qui effectuent, à différents endroits, plusieurs activités physiques telles que la marche, le jogging, la montée des escaliers, etc. Pour construire le modèle, l'apprentissage en profondeur, les réseaux de neurones récurrents (RNN) et les réseaux de mémoire à court terme (LSTM) ont été utilisés. L'environnement de développement utilisé est Google Collab dont le langage de programmation est Python 3.6. Après l'entraînement, le modèle a été enregistré et exporté vers une application Android pour les prédictions en temps réel, avec une interface utilisateur pour exprimer les résultats à l'aide de l'API de synthèse vocale. Le résultat final consiste en, l'affichage de la séquence de probabilité de chaque activité en temps-réel.

Un travail de détection et classification des émotions des personnes a été proposé par [GUE & All, 19], il consiste à mettre en œuvre une procédure de traitement pour la reconnaissance d'émotion

à partir d'images. Pour la classification et la prédiction, ils ont utilisé le classificateur SVM (polynomiale et linéaire), les classifieurs KNN et RFC qui ont été très performants au niveau de la reconnaissance des états émotionnels à partir des exemples. Pour certains exemples leurs classificateurs ont atteint un taux de précision de 95%. Ils ont choisi la plateforme anaconda et le langage python pour l'implémentation du modèle. La base de données utilisée s'appelle KDEP, cette base de données se compose de 140 images dans chacune des 7 états d'émotions, ce qui fait un total de 980 images.

[SAL & Ali, 19] ont présenté un travail qui consiste à proposer une solution qui permettra de détecter si une machine windows est infectée par diverses familles de logiciels malveillants en fonction des différentes propriétés de cette machine. Ils ont créé leur propre modèle (XGBOOST classifieur) qui aide à résoudre le problème posé. Ils ont utilisé le jeu de données Microsoft Malware Prediction et la plateforme Google colab et le langage python pour l'implémentation du modèle. Les différents entraînements sur le modèle ont été évalués et testés.

Une nouvelle proposition a été développée par [SAM, 19] intitulée « Implémentation et évaluation d'un modèle d'apprentissage automatique pour l'estimation de la valeur marchande de Propriétés immobilières ». L'auteur a défini un modèle utilisant la technique d'apprentissage ensembliste Bagging et cela en utilisant l'algorithme du Random Forest, par la suite, il a construit un second modèle et cela en utilisant une autre technique d'apprentissage ensembliste Boosting. Pour le troisième et dernier modèle, l'auteur opte pour l'utilisation des réseaux de neurones vu l'intérêt grandissant qu'ils portent actuellement à la communauté scientifique.

une entreprise d'annonce immobilière, la collection est hébergée sur Kaggle. Quant aux premier et deuxième modèles, il a trouvé des résultats assez similaires pour les trois mesures de performances, néanmoins le modèle 2 prend énormément de temps pour son entraînement comparativement aux deux autres modèles.

Dans [DJA, 21] les auteurs ont proposé un modèle de prédiction, passant par deux étapes: ingénierie des fonctionnalités et classification. En combinant les techniques (XGBoost) et DT, ils ont proposé un sélecteur de caractéristiques hybride pour minimiser la redondance des fonctionnalités, après, ils utilisent la technique d'élimination des fonctionnalités récursives (RFE) pour réduire les dimensions et améliorer la sélection des fonctionnalités. Finalement, pour prévoir la charge électrique, ils ont appliqué la SVM. L'évaluation expérimentale a été réalisée sur des données quotidiennes sur la charge électrique de la zone de contrôle de l'ISO en Nouvelle-Angleterre (ISO NECA), Les données sont normalisées. Les données sont classées en trois parties : former, tester et valider. Afin d'évaluer et de comparer les modèles, deux mesures d'évaluation ont été utilisées : RMSE, MAPE.

Dans ce travail, [AYB & Ali, 20] ont proposé un modèle basé deep Learning pour la prédiction de la charge électrique en faisant une sélection moyenne. La sélection des caractéristiques réduit la complexité du modèle en fournissant les caractéristiques les plus importantes pour le classificateur. Le Random Forest (RF) et La technologie Extreme Gradient Boosting (XGB) ont été utilisés pour la sélection des caractéristiques et Recursive Feature Elimination (RFE) est utilisée comme méthode d'extraction des caractéristiques. L'activité d'ingénierie de la charge électrique affine les données et les transmet au classificateur. Les technologies Convolutional Neural Network Gated Recurrent Unit (CNN-GRU) et Support Vector Machine (SVM) ont été utilisés pour la classification.

Afin d'améliorer les performances du classifieur, les paramètres de CNN-GRU et SVM ont été ajustés par utilisation respectivement des algorithmes d'optimisation Earth Worm Optimization (EWO) et Grey Wolf Optimization (GWO). L'algorithme d'optimisation trouve les meilleures valeurs optimales pour les techniques d'hyperparamètres. De plus, l'ajustement des paramètres peut fournir la meilleure valeur pour le classificateur, réduisant ainsi le risque de débordement du modèle et aidant à améliorer la précision du modèle. Le rendement de ce modèle a été évalué à l'aide de l'erreur absolue moyenne en pourcentage (MAPE), erreur racine-moyenne-carrée (RMSE), erreur absolue moyenne (MAE), erreur racine-moyenne (MSE), précision, rappel, f-mesure et précision (Accuracy). Les précisions de CNN-GRU-EWO et SVM-GWO sont de 96,33% et 93,99%, respectivement. Les techniques proposées fonctionnent 7% et 3% mieux que CNN et SVM classificateurs.

Dans cette étude « **Une amélioration de la détection d'intrusion par les méthodes de sélection des fonctionnalités à l'aide des arbres de décision** » présenté par [MEZ, 20], le but est de détecter et de réduire l'erreur des systèmes de détection d'intrusion, aussi, de tester l'effet de l'élimination des caractéristiques sans importance et obsolètes des ensembles de données sur le succès de la classification, en utilisant le classifieur arbres de décision. L'implémentation et le développement de l'approche est utilisée dans la classification des attaques. Dans l'ensemble, les mesures de performance tel que le Recall, permet de prendre des bonnes décisions concernant le taux de détection. Pour ce faire, l'auteur a utilisé la plate-forme Anaconda et le langage de programmation python. Les modèles proposés ont ensuite été évalués à l'aide des ensembles de données CICDDoS2019. Le résultat obtenu est de 96%.

[SAI, 21] a présenté une **technique de NLP pour la détection des fausses nouvelles**, ils se sont concentrés sur la classification des fake news, en utilisant des algorithmes de ML et DL. Pour faire réussir cette classification, une phase de pré-traitement du texte utilisant le NLP est effectuée avant de lancer l'apprentissage en profondeur.

L'ensemble de données se compose de 40 000 fausses et vraies nouvelles. L'objectif est de former un modèle pour prédire avec précision si une information particulière est vraie ou fausse. L'expérimentation a montré que l'utilisation des techniques de preprocessing en combinaison avec le modèle LSTM du DL, donne une bonne précision de classification (environ 99%).

« *Bank fraud detection using sequential pattern mining* » est l'application réalisée par [BER, 21] dans le but de connaître et découvrir l'existence de la fraude bancaire à travers une série de comportements formés par les clients dans un certain laps de temps en utilisant les algorithmes d'exploration de motifs séquentiels. Ils ont exécuté BankSim pendant 180 étapes (environ six mois), plusieurs fois et calibré les paramètres afin d'obtenir une distribution suffisamment proche pour être fiable pour les tests. Pour entraîner le modèle, ils ont choisi l'algorithme knn. Le choix de l'implémentation a été fait sur le langage Python et Spyder (anaconda3).

[DJE, 22] a développé un système de détection d'anomalie, adapté à la détection de la fraude documentaire en utilisant une méthode de l'apprentissage profond qui est l'autoencodeur convolutif. Ils ont utilisé une base de données qui est construite dans le LABO LESIA, contenant 512 images documentaires. La base de données est divisée en 2 classes. Comme suit : (i). la Classe 1, contient des images authentiques. Dont 173 images sont utilisées pour l'apprentissage et 22 images sont utilisées pour la validation, et (ii). la Classe 2, contient 317 images falsifiées, qui sont utilisées pour le test.

Le modèle est implémenté dans l'environnement de développement Google Colab qui donne un accès direct à python. Les résultats obtenus sont satisfaisant en termes de précision et d'erreur.

Nous allons présenter dans les sections suivantes un ensemble de travaux de recherche traitant quelques problèmes dans l'optique de la détection, la classification et la prédiction des fraudes de n'importe quelle nature.

De ce fait, divers articles et études ont proposé des méthodes et des algorithmes pour classer, indexer, segmenter et discriminer des données de consommation, telles que l'utilisation des séries temporelles. Ces méthodes incluent l'utilisation de mesures de similarité, d'algorithmes d'apprentissage automatique et de distances telles que la distance euclidienne et la distance dynamique de déformation temporelle. Certains chercheurs ont appliqué ces méthodes pour classer la consommation d'eau et d'électricité en fonction de différents critères tels que le type de résidence et le nombre d'occupants. La détection des anomalies, basée sur ces techniques, est également utilisée dans divers domaines tels que la fraude par carte de crédit, l'assurance, la santé et la cybersécurité.

Clustering des données

Dans le domaine du clustering, il est souvent difficile de connaître le nombre de classes à l'avance. Afin de résoudre ce problème, plusieurs chercheurs, tels que [TIB & All, 01], ont proposé une approche basée sur la statistique des écarts. Cette méthode consiste à comparer la variation intra-cluster pour différentes valeurs de k avec celle attendue dans une distribution de référence nulle des données. Le nombre optimal de clusters est déterminé en maximisant cette statistique d'écart pour chaque cluster. Ainsi, on cherche à assigner un échantillon x à une classe en vérifiant s'il est proche des autres échantillons du même cluster

Détection d'anomalie de consommation

Dans les études sur les anomalies de la consommation d'eau [KAP & All, 10], [LUI & All ,18], [LUI & All ,18], [GUS & All, 19], les deux problèmes les plus fréquemment abordés sont la détection des fuites et l'identification des pics de consommation. À cet égard, plusieurs méthodes ont été proposées. Par exemple, [KAZ & All ,17] a suggéré une approche pour détecter et estimer les fuites en se basant sur les mesures de débit dans les réseaux de distribution d'eau.

La prévision de consommation

La prévision, est un sujet largement abordé [WEI & All, 14], [CHE & All, 14], [CHI & All ,17] et [ARO & All, 16] avec des applications étendues dans divers domaines. Par exemple, dans le domaine de la finance, elle est utilisée pour anticiper les fluctuations des cours de la bourse. En météorologie, elle est appliquée pour prédire la température à court terme. De plus, elle est également utilisée pour estimer le nombre de réservations d'un vol.

III. SYNTHÈSE

Nous avons présenté quelques travaux utilisant différents modèles et algorithmes d'apprentissage automatique les plus souvent utilisés. D'après une analyse et comparaison effectuées des modèles présentés, il convient de dire qu'il n'y a pas un algorithme qui est meilleur que les autres dans toutes les situations, ce qui signifie qu'il n'y a pas un algorithme universel et qu'il faut choisir l'algorithme adapté pour chaque problème/contexte. Et donc, nous avons retenu quelques critères qui guident le

choix de l'algorithme (ou modèle) de l'apprentissage, nous les résumons comme suit : (1). *Problème de tâche*, (2). *S'agit-il d'une tâche de classification ou régression ?*, (3). *Types de caractéristiques sur les données*, (4). *La nature des caractéristiques (catégorielles/numériques)*, (5). *Complexité et temps de l'entraînement*, (6). *Mémoire nécessaire pour l'entraînement*, (7). *Capacité du modèle par rapport à la complexité de la fonction d'apprentissage*, (8). *Fonction linéaire ou non*, (9). *Volume de données d'entraînement*, (10). *Sensibilité de bruit dans les données*, (11). *Performance sur les tâches*, (12). *Justification de la décision prise par le modèle...*et encore d'autres critères peuvent être définis.

Une comparaison des algorithmes d'apprentissage en prenant en compte les critères cités précédemment est présente dans le tableau ci-dessous

	Rég./Class	Catég./Cont.	Complexité Entr.	Capacité	Perform.	Données	Sensibilité aux bruits
Régression linéaire	Régression	Cont.	Simple	Linéaire	Moyenne	Peu	Haut
Arbre de déc. (Forêt)	Class. et Rég.	Cat. (Cont.)	Compl.	Non.Lin	Bonne	Peu	Haut
Bayésien Naïf	Class.	Cat. (Cont.)	Simple	Linéaire	Moyenne	Peu	Bas
SVM	Class. et Rég	Cat. et Cont.	Compl.	Lin./Noyau	Bonne	Peu	Bas
Réseaux de neurones	Class. et Rég	Cat. et Cont.	Compl.	Non.Lin	Bonne	Beaucoup	Bas
Xgboost	Class. et Rég	Cat. et Cont	Simple et Compl	Non.Lin	Très bonne	Beaucoup	Bas
Lightgbm	Class. et Rég	Cat. et Cont	Simple	Non.Lin	Très bonne	Beaucoup	Bas

Tableau II.1– Une comparaison des algorithmes d'apprentissage

IV. CONCLUSION

A travers ce survol, nous avons présenté quelques modèles connexes aux machines learning basés sur l'apprentissage automatique (supervisé, profond). Dans la dernière section, nous avons mis le point – à travers une synthèse – sur les critères qui peuvent guider le choix du modèle de l'apprentissage.

CHAPITRE

III

LA FAMILLE DES
ALGORITHMES DU
GRADIENT BOOSTING

III

La famille des algorithmes gradient boosting

Sommaire

1 INTRODUCTION.....	34
II HISTORIQUE	34
III QU'EST-CE QUE LE GRADIENT BOOSTING ?	34
IV LE BOOSTING EN MACHINE LEARNING	35
V PRINCIPAUX TYPES D'ALGORITHMES DE BOOSTING	35
VI COMPARAISON DES CARACTERISTIQUE DES MODELES.....	37
VII FONCTIONNEMENT DES ALGORITHMES DE BOOSTING.....	37
VIII AVANTAGES DU BOOSTING	38
IX DOMAINE D'APPLICATION D'ALGORITHMES DE GRADIENT BOOSTING	38
X CONCLUSION.....	39

I. INTRODUCTION

Les algorithmes de gradient boosting sont une technique d'apprentissage automatique très populaire. Ils sont utilisés pour résoudre des problèmes complexes tels que la prédiction de tendances du marché, la détection de fraudes et la reconnaissance de formes. Les algorithmes de gradient boosting sont très efficaces car ils combinent plusieurs modèles d'apprentissage automatique plus simples pour obtenir un modèle plus précis. Le processus de gradient boosting consiste à entraîner plusieurs modèles d'apprentissage automatique plus simples, appelés "arbres de décision", et à les combiner pour former un modèle plus puissant. Ce modèle peut ensuite être utilisé pour prédire de nouveaux résultats avec une grande précision. Les algorithmes de gradient boosting sont très utiles pour les entreprises qui cherchent à améliorer leur efficacité, leur rentabilité et leur précision.

Dans ce chapitre, nous explorerons en détail la famille des algorithmes de Gradient Boosting, en examinant leur historique, la définition du gradient boosting, les types d'algorithmes, ainsi que leur fonctionnement et leurs avantages. Nous discuterons également des applications courantes de ces algorithmes dans différents domaines, ainsi que des meilleures pratiques pour les utiliser efficacement.

II. HISTORIQUE

Le gradient boosting trouve ses origines dans les années 1990 lorsque Leo Breiman a introduit le concept de boosting comme algorithme d'optimisation pour une fonction de coût appropriée. En 1995, Adaptive Boosting (AdaBoost) est devenu la première réalisation réussie du boosting.

Ensuite, en 1999, Jerome Friedman a proposé une généralisation du boosting avec l'introduction du Gradient Boosting (Machine), également connu sous le nom de GBM. L'objectif de cette approche était d'améliorer la vitesse de convergence du boosting en intégrant à la fois les informations historiques et les informations de gradient actuelles.

L'implémentation la plus célèbre de l'algorithme de gradient boosting est XGBoost, largement reconnue pour ses performances et ses victoires dans de nombreuses compétitions Kaggle et Zindi. Cependant, bien qu'il fonctionne bien sur diverses tâches, il présente certaines limites en termes d'évolutivité. Pour résoudre ce problème, d'autres compétiteurs ont développé des variations telles que LightGBM, HistGradientBoosting et CatBoost, chacune avec ses propres astuces de mise en œuvre.

Aujourd'hui, le gradient boosting est devenu une technique populaire et puissante dans le domaine de l'apprentissage automatique. Ses différentes implémentations ont été utilisées avec succès dans un large éventail d'applications pratiques [CHO, 22].

III. QU'EST-CE QUE LE GRADIENT BOOSTING ?

Le Gradient Boosting est une technique d'apprentissage automatique qui consiste à construire un modèle prédictif puissant en combinant de manière itérative plusieurs modèles plus simple. Cette approche relève de la famille des méthodes d'ensemble, où plusieurs modèles sont utilisés ensemble pour obtenir de meilleures performances prédictives.

L'idée principale du Gradient Boosting est d'ajuster progressivement les modèles faibles pour corriger les erreurs commises par les modèles précédents. À chaque itération, un nouveau modèle faible est entraîné en se concentrant sur les exemples mal prédits par les modèles précédents. Les prédictions de tous les modèles faibles sont ensuite combinées pour obtenir une prédiction finale.

Le terme "Gradient" dans Gradient Boosting fait référence à l'utilisation du gradient de la fonction de perte pour guider l'ajustement des modèles faibles. Le modèle est ajusté dans la

direction opposée au gradient de la fonction de perte afin de minimiser progressivement la perte et d'améliorer les performances du modèle.

L'algorithme de Gradient Boosting peut être utilisé pour résoudre des problèmes de régression, de classification et de classement. Il a été largement utilisé dans divers domaines de l'apprentissage automatique et des statistiques en raison de sa capacité à capturer des relations complexes, sa résistance au surajustement et sa flexibilité.

Des bibliothèques populaires telles que XGBoost, LightGBM et CatBoost fournissent des implémentations efficaces de l'algorithme de Gradient Boosting, avec des fonctionnalités avancées telles que la gestion des valeurs manquantes, l'optimisation des hyperparamètres et la parallélisation pour des performances accrues [KUR, 20].

IV. LE BOOSTING EN MACHINE LEARNING

Les scientifiques des données utilisent des données étiquetées pour entraîner des logiciels de machine learning (appelés modèles de machine learning) à faire des prédictions sur des données non étiquetées. Un modèle de machine learning unique peut faire des erreurs de prédiction selon la précision du jeu de données d'entraînement. Par exemple, si un modèle d'identification de chats n'a été entraîné que sur des images de chats blancs, il peut occasionnellement faire des erreurs lors de l'identification d'un chat noir. Le boosting s'efforce de résoudre ce problème en entraînant successivement plusieurs modèles afin d'améliorer la précision du système global.

Les algorithmes de boost présentent plusieurs avantages en termes de mise en œuvre et de prétraitement des données.

V. PRINCIPAUX TYPES D'ALGORITHMES DE BOOSTING

Il existe plusieurs types de boosting, cependant, nous allons citer les trois principaux :

- **Boosting adaptatif (AdaBoost)**

C'est l'un des premiers modèles de boosting développés. Il s'adapte et tente de s'autocorriger à chaque itération du processus de boosting. AdaBoost donne initialement le même poids à chaque ensemble de données. Ensuite, il ajuste automatiquement le poids des points de données après chaque arbre de décision. Il donne plus de poids aux éléments mal classés afin de les corriger pour la prochaine tournée. Il répète le processus jusqu'à ce que l'erreur résiduelle tombe sous un seuil acceptable. AdaBoost est un type de boosting approprié pour les problèmes de classification [FER & AIL, 12].

- **Boosting de gradient**

Egalement appelé amplification de gradient, il est similaire à AdaBoost dans la mesure où il s'agit également d'une technique d'entraînement séquentiel. La différence entre AdaBoost et le boosting de gradient est que le logiciel de boosting de gradient optimise la fonction de perte en générant des apprenants de base de manière séquentielle, de sorte que l'apprenant de base actuel soit toujours plus efficace que le précédent. Cette méthode tente de produire des résultats précis au départ au lieu de

corriger les erreurs tout au long du processus, comme AdaBoost. Le boosting de gradient peut aider à résoudre les problèmes de classification et de régression [FER & All, 12].

- **Boosting de gradient extrême (XGBoost)**

Il améliore le boosting de gradient en termes de vitesse de calcul et d'échelle de plusieurs façons. XGBoost utilise plusieurs cœurs sur le CPU afin que l'apprentissage puisse se faire en parallèle pendant l'entraînement. Il s'agit d'un algorithme de boosting qui peut traiter de vastes jeux de données, ce qui le rend attrayant pour les applications de big data. Les principales caractéristiques de XGBoost sont la parallélisation, le calcul distribué, l'optimisation du cache et le traitement hors du cœur apporte plusieurs améliorations par rapport à d'autres implémentations de boosting. Il permet de capturer des interactions d'ordre supérieur, mais les arbres plus profonds peuvent poser des problèmes tels que la diminution du nombre d'observations dans chaque nœud terminal, ce qui peut entraîner une variance plus élevée des poids émis des feuilles. Une régularisation plus forte peut donc être nécessaire lors du boosting avec des arbres plus profonds. De plus, le modèle peut modéliser des interactions là où elles ne sont pas présentes, ce qui peut augmenter inutilement la variance. Cela peut être un domaine où les méthodes de boosting actuelles pourraient être améliorées [FER & All, 12].

- **L'algorithme MART (Multiple Additive Regression Trees)**

Est une variante de l'algorithme de boosting qui utilise des arbres de régression pour prédire des valeurs continues. L'algorithme utilise une fonction de coût exponentielle pour mettre davantage l'accent sur les erreurs de prédiction les plus importantes. L'algorithme MART est similaire à l'algorithme Gradient Boosting, mais utilise une méthode différente pour ajuster le poids des observations [DID, 16].

- **LightGBM (Light Gradient Boosting Machine)**

Est un cadre de boosting de gradient distribué et open-source pour l'apprentissage automatique, initialement développé par Microsoft. Il est basé sur des algorithmes d'arbres de décision et utilisés pour le classement, la classification et d'autres tâches d'apprentissage automatique. Les avantages de LightGBM incluent une vitesse d'entraînement plus rapide, une utilisation de mémoire plus faible, une meilleure précision, une prise en charge de l'apprentissage parallèle, distribué et GPU, et une capacité à gérer des données à grande échelle. LightGBM crée des arbres de décision qui se développent en feuille, ce qui signifie qu'une seule feuille est divisée en fonction du gain. LightGBM utilise une méthode basée sur l'histogramme dans lequel les données sont regroupées en bacs à l'aide d'un histogramme de la distribution [GEE, 23].

- **HistGradientBoostingClassifier**

HistGradientBoostingClassifier est une implémentation alternative de l'algorithme Gradient Boosting qui a été proposée dans la version 0.21.0 de la bibliothèque scikit-learn en tant qu'estimateur expérimental. Depuis la version 1.0.0, cet estimateur est devenu un estimateur stable. Il s'agit d'une implémentation basée sur l'histogramme de Gradient Boosting qui peut traiter les données manquantes grâce à son support intégré pour les valeurs manquantes. HistGradientBoostingClassifier est capable de gérer des ensembles de données hétérogènes et peut être utilisé pour la classification et la régression. Il est également plus rapide que l'implémentation standard de Gradient Boosting, car il utilise des histogrammes pour discrétiser les variables continue. HistGradientBoostingClassifier est basé sur l'algorithme LightGBM de Microsoft et utilise OpenMP pour la parallélisation. Il dispose

également de plusieurs hyperparamètres qui peuvent être ajustés pour améliorer les performances du modèle. HistGradientBoostingClassifier est souvent comparé à d'autres bibliothèques de Gradient Boosting telles que XGBoost, LightGBM et CatBoost [LUV, 22].

VI. COMPARAISON DES CARACTERISTIQUES DES MODELES

Les algorithmes de Gradient Boosting sont des techniques de Machine Learning très populaires pour résoudre des problèmes de classification et de régression. Les deux principaux algorithmes de Gradient Boosting sont XGBoost et LightGBM. XGBoost est connu pour être plus rapide et plus précis sur des ensembles de données plus petits, tandis que LightGBM est plus performant sur des ensembles de données plus grands [VIC, 21]

	XGBoost	LightGBM
Popularité	★ ★ ★	★ ★
Gestion automatique des variables catégorielles	★	★ ★
Rapidité d'entraînement	★	★ ★ ★ ★
Rapidité prédiction	★ ★	★ ★ ★

Tableau III.1– Comparaison synthétique entre XGBoost et LightGBM

VII. FONCTIONNEMENT DES ALGORITHMES DE BOOSTING

Les algorithmes de boosting sont des techniques d'apprentissage automatique qui combinent plusieurs modèles d'apprentissage faibles pour créer un modèle plus fort. Voici comment fonctionnent les algorithmes de boosting :

Construction de modèles faibles : les algorithmes de boosting commencent par construire plusieurs modèles d'apprentissage faibles, qui ont une précision légèrement supérieure à celle d'un modèle aléatoire.

Entraînement itératif : les modèles faibles sont entraînés itérativement en utilisant les données d'entraînement.

À chaque itération le modèle est ajusté pour corriger les erreurs du modèle précédent. Poids des données : les données d'entraînement sont pondérées en fonction de leur à être classées correctement. Les données mal classées ont un poids plus élevé, tandis que les données bien classées ont un poids plus faible.

Agrégation des modèles : les modèles faibles sont agrégés pour former un modèle plus fort. Les prédictions de chaque modèle sont combinées pour donner une prédiction finale.

Réduction de l'erreur : le modèle final est conçu pour réduire l'erreur de prédiction en utilisant des techniques telles que la réduction de la variance et la réduction du biais. [CAY, 22].

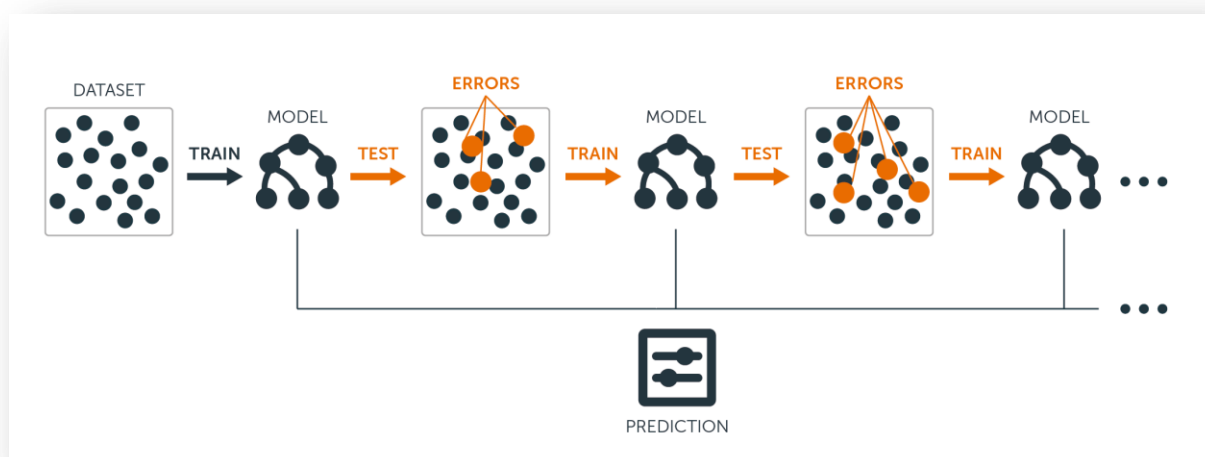


Figure III.1– Fonctionnement des algorithmes de boosting [MM, 19].

VIII. AVANTAGES DU BOOSTING

Dans l'ensemble, les algorithmes de boosting sont relativement faciles à mettre en œuvre et disposent de routines intégrées pour gérer les données manquantes, ce qui en fait un choix populaire pour les tâches d'apprentissage automatique [LAT,23].

Ces algorithmes présentent un certain nombre d'avantages, nous citons :

- Le boosting peut être utilisé avec plusieurs options de réglage d'hyper-paramètres pour améliorer l'ajustement.
- Aucun prétraitement des données n'est nécessaire et les algorithmes ont des routines intégrées pour gérer les données manquantes
- Boosting se concentre davantage sur les prédictions mal classées, ce qui peut aider à réduire les biais

Facilité de mise en œuvre:

- Le boosting est une technique d'apprentissage d'ensemble séquentiel qui n'est pas difficile à apprendre et à appliquer.
- Les algorithmes de boosting ont des routines intégrées pour gérer les données manquantes, ce qui facilite la mise en œuvre.

Dans l'ensemble, les algorithmes de boosting sont relativement faciles à mettre en œuvre et disposent de routines intégrées pour gérer les données manquantes, ce qui en fait un choix populaire pour les tâches d'apprentissage automatique [ZUL, 23].

IX. DOMAINE D'APPLICATION DES ALGORITHMES DE GRADIENT BOOSTING

Le boosting est une technique d'apprentissage automatique qui combine plusieurs modèles d'apprentissage faibles pour créer un modèle plus fort. Les algorithmes de boosting sont utilisés dans une variété de domaines d'application, notamment :

La classification et la régression : les algorithmes de boosting sont souvent utilisés pour la classification et la régression dans des domaines tels que la finance, la publicité en ligne, la reconnaissance de la parole et la vision par ordinateur.

L'apprentissage en ligne : les algorithmes de boosting sont également utilisés pour l'apprentissage en ligne, où les données sont générées en continu et le modèle doit être mis à jour en temps réel.

La reconnaissance de texte : les algorithmes de boosting sont utilisés pour la reconnaissance de texte, où ils peuvent aider à améliorer la précision de la reconnaissance de caractères.

La détection d'anomalies : les algorithmes de boosting peuvent également être utilisés pour la détection d'anomalies, où ils peuvent aider à identifier les points de données qui ne correspondent pas au modèle.

La prédiction de la demande : les algorithmes de boosting peuvent être utilisés pour la prédiction de la demande dans des domaines tels que la vente au détail et la logistique, où ils peuvent aider à prédire les tendances de la demande et à optimiser les stocks [ZHA, 21].

X. CONCLUSION

Ce chapitre a été consacré à la famille des algorithmes de boosting qui est une technique d'apprentissage automatique qui combine plusieurs modèles d'apprentissage faibles pour former un modèle plus fort. Nous avons exposé les variantes d'algorithmes de boosting, tels que, AdaBoost, Gradient Boosting et XGBoost, qui sont largement utilisés. XGBoost est particulièrement connu pour sa performance exceptionnelle dans les compétitions **ZINDI**. Cependant, d'autres implémentations telles que LightGBM, HistGradientBoosting et CatBoost ont été développées pour résoudre les limitations de XGBoost en termes de scalabilité et de gestion des données manquantes.

Nous avons également montré quelques points forts de ces algorithmes de boost, qui sont populaires en raison de leur facilité d'interprétation, de l'absence de prétraitement des données et de leur capacité à traiter les valeurs manquantes. Cependant, ils peuvent être sensibles aux valeurs aberrantes et aux données atypiques qui peuvent influencer les résultats.

Nous avons consacré la dernière section de ce chapitre aux domaines d'utilisation des algorithmes de boosting, allant de la classification et de la régression à la détection d'anomalies, à l'analyse de texte et à l'analyse financière. Ces algorithmes offrent une approche puissante pour améliorer les performances des modèles d'apprentissage automatique en combinant plusieurs modèles plus faibles en un modèle global plus fort.

PROPOSITION

CHAPITRE

DE MODELES

D'APPRENTISSAGE

POUR LA DETECTION

DE FRAUDE

D'ENERGIE

[ELECTRICITE - GAZ]

IV

Proposition de modèles d'apprentissage
pour la détection et la classification
de fraudes d'énergie [Electricité – Gaz]

IV

Sommaire

I INTRODUCTION.....	39
II CONCEPTION DES MODELES	39
II.1 Spécification fonctionnelle du suivi des consommations	39
II.2 Description des modèles	40
II.3 Comparaison des deux modèles	42
II.4 Métrique d'évaluation	43
II.5 Jeux de données pour les modèles de détection et classification.....	43
II.5.1 Structuration de la base de données	43
II.5.2 Définition des structures de données	45
II.6 Partitionnement des données	46
III APPLICATION DU PROCESSUS D'ECD DANS NOTRE APPROCHE.....	46
IV EXPERIMENTATION ET RESULTATS	47
IV.1 Architecture globale	47
IV.2 Application des modèles (XGBoost- LightGBM).....	48
IV.3 Comparaison des modèles et Synthèse.....	59
IV.4 Généralisation du modèles [LightGBM].....	62
V VALIDATION DES RESULTATS PAR ZINDI.....	63
VI CONCLUSION.....	65

I. INTRODUCTION

Nous avons présenté dans le chapitre précédent la famille d'algorithmes de boosting, leurs principes de fonctionnement et d'autres paramètres.

Dans ce chapitre, nous choisirons deux algorithmes de boosting et nous les entraînerons avec des données collectées. Nous montrons l'impact de chaque algorithme et sa précision.

Ce chapitre est divisé en deux volets distincts. Le premier volet aborde la conception de notre système, les pseudos codes des deux modèles LightGBM et XGBoost, ainsi que les nuances qui les différencient. Le second volet présente l'architecture globale de notre système dédié à la détection de la fraude dans la consommation d'électricité et de gaz. Les résultats obtenus sont également présentés, et les discussions qui en découlent sont abordées.

Le détail de l'environnement de développement utilisé, langage de programmation choisi, ainsi que les bibliothèques utilisées seront illustrés en annexe 1 de ce présent mémoire.

II. CONCEPTION DES MODELES

Ce volet du chapitre se concentre sur la conception approfondie de notre système de détection de fraude dans la consommation d'électricité et de gaz. Elle jettera les bases conceptuelles solides sur lesquelles repose notre système, en fournissant une compréhension approfondie de son fonctionnement interne.

II.1 Spécification fonctionnelle du suivi des consommations

La figure suivante représente un processus global, commençant de la collecte des données jusqu'à l'évaluation et la classification des consommations des clients. Nous notons que ce diagramme ne représente pas la description de la structure de la base de données (jeu des données utilisé), mais plutôt le fonctionnement global du suivi des consommations.

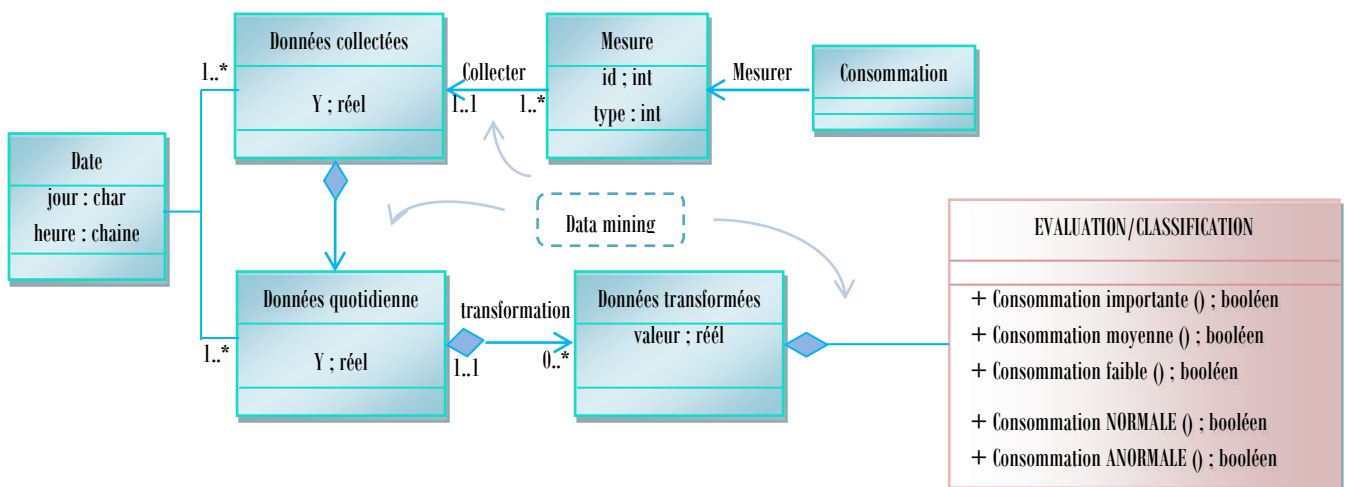


Figure IV.1 – Diagramme de classe général du fonctionnement du suivi des consommations.

II.2 Description des modèles

Notre objectif est de proposer des modèles de classification supervisés capables d'identifier et détecter les utilisations frauduleuses des consommations d'énergie. Deux techniques ont été testées, le LightGBM et le XGBoost, nous décrivons leurs fonctionnements dans ce qui suit :

A. Principe de classification utilisant le modèle LightGbm

Algorithme IV.1 - Modèle de classification de fraude avec LightGBM utilisant la validation croisée

Entrées : Données d'entraînement (X_{train} , y_{train}), données de test (X_{test} , y_{test})
 X (caractéristiques), y (étiquettes), $random_state$ (graine aléatoire) // les entrées de la validation croisée

```

1 : Importation des bibliothèques
2 : Importation des données
3 : validation croisée ()
4 : kf = KFold(n_splits=5, shuffle=True, random_state=random_state) //Initialisation de
    la validation croisée
5 : // Boucle de validation croisée
6 : Pour train_index, test_index dans kf.split(X, y) faire
7 :   // Sélectionner les ensembles d'entraînement et de test
8 :      $X_{train} = X.iloc[train\_index]$ 
9 :      $X_{test} = X.iloc[test\_index]$ 
10:     $y_{train} = y.iloc[train\_index]$ 
11:     $y_{test} = y.iloc[test\_index]$ 
10: // Affichage des informations sur les ensembles d'entraînement et de test
11 : Ecrire( $X_{train}$ ,  $y_{train}$ ,  $X_{test}$ ,  $y_{test}$ )
12 : Fin validation_croisée ()
12 : //Création et entraînement du modèle Lightgbm
13 : model = CréerModèleLightgbm()
14 : EntraînerModèleLightgbm(model,  $X_{train}$ ,  $y_{train}$ )
15 : //Prédictions et évaluation des performances
16 :  $y_{pred} = PrédireFraudeLightgbm(model, X_{test})$ 
17: EvaluerPerformances( $y_{test}$ ,  $y_{pred}$ )

```

Sorties : modèle entraîné (model), score de précision (accuracy).

Fin Algorithme

B. Principe LightGbm

Algorithme IV.2 - Principe LightGBM

Entrées : Nombre maximum d'itérations ($max_iterations$)
 Critère d'arrêt basé sur l'amélioration insuffisante (tolerance)
 Paramètres du modèle ($learning_rate$, max_depth , $n_estimators$, etc.)

```

1 : // Initialisation des poids des instances ( $w$ ) uniformément
2: Pour chaque itération (t) de 1 à  $max\_iterations$ 
3 :   Construction d' un arbre de décision en utilisant les instances pondérées ( $w$ )
4 :   Calcul des gradients et les hessiens pour chaque instance en fonction des prédictions

```

actuelles et des labels réels.

5 : **Recherche de ligne ()** // pour trouver la meilleure valeur de la fonction de perte pour l'arbre de décision.

6 : Mise-à-jour-prédictions () // ajoute l'arbre de décision pondéré à l'ensemble actuel de prédictions.

7 : Mise-à-jour-poids-instances (w) // en fonction des erreurs résiduelles obtenues.

8 : Répéter jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint (par exemple, un nombre maximum d'itérations ou une amélioration insuffisante).

Fin Algorithme

C. Principe de classification utilisant le modèle XGBoost

Algorithme IV.3 - Modèle de classification de fraude avec XGBoost utilisant la validation croisée

Entrées : Données d'entraînement (X_train, y_train), données de test (X_test, y_test)

X (caractéristiques), y (étiquettes), random_state (graine aléatoire) // les entrées de la validation croisée

1 : Importation des bibliothèques

2 : Importation des données

3 : validation croisée ()

4 : kf = KFold(n_splits=5, shuffle=True, random_state=random_state) //Initialisation de la validation croisée

5 : // Boucle de validation croisée

6 : Pour train_index, test_index dans kf.split(X, y) faire

7 : // Sélectionner les ensembles d'entraînement et de test

8 : X_train = X.iloc[train_index]

9 : X_test = X.iloc[test_index]

10 : y_train = y.iloc[train_index]

11 : y_test = y.iloc[test_index]

10 : // Affichage des informations sur les ensembles d'entraînement et de test

11 : Ecrire(X_train, y_train, X_test, y_test)

12 : Fin validation_croisée ()

13 : //Création et entraînement du modèle Lightgbm

14 : model = CréerModèleXgboost()

15 : EntraînerModèleXgboost(clf, X_train, y_train)

15 : //Prédictions et évaluation des performances

16 : y_pred = PrédireFraudeXgboost(clf, X_test)

17 : EvaluerPerformances(y_test, y_pred)

Sorties : modèle entraîné (model), score de précision (accuracy).

Fin Algorithme

D. Principe XGBoost

Algorithme IV.4 - Principe XGBoost

Entrées : Nombre maximum d'itérations (max_iterations), Critère d'arrêt basé sur l'amélioration insuffisante (tolerance), Paramètres du modèle (learning_rate, max_depth, n_estimators, etc)

1 : // Initialisation des poids des instances

2 : Pour chaque itération (t) de 1 à max_iterations

-
- 3 : Construction d'un arbre de décision en utilisant les instances pondérées (w)
 - 4 : Calcul des gradients et des hessiens pour chaque instance en fonction des prédictions actuelles et des labels réels
 - 5 : **Recherche exhaustive** () //pour trouver la meilleure valeur de la fonction de perte pour l'arbre de décision
 - 6 : Mise-à-jour-prédictions () //en ajoutant l'arbre de décision pondéré à l'ensemble actuel de prédictions
 - 7 : Mise-à-jour-poids-instances (w) // en fonction des erreurs résiduelles obtenues, en utilisant une régularisation (par exemple, la réduction du taux d'apprentissage).
 - 8 : Répéter jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint.
-

Fin Algorithme

II.3 Comparaison des deux modèles

Nous allons procéder à une comparaison en tenant en compte l'aspect conceptuel des modèles par rapport à leur composition et fonctionnement, dans la section relative à l'expérimentation, nous exposons les résultats obtenus. Nous avons déjà cité dans les chapitres précédents que **LightGBM** est un modèle de Machine Learning se basant sur le renforcement de gradient et les arbres de décision. Tout comme **XGBoost**, les deux modèles sont très populaires et reconnus par leur haute performance. Mais en quoi sont-ils différents alors ?

- **LightGBM fait croître l'arbre verticalement** grâce à l'algorithme GOSS contrairement à XGBoost qui fait pousser des arbres horizontalement, en d'autres mots LightGBM fait pousser l'arbre par **feuille** tandis que l'autre algorithme se développe par niveau.

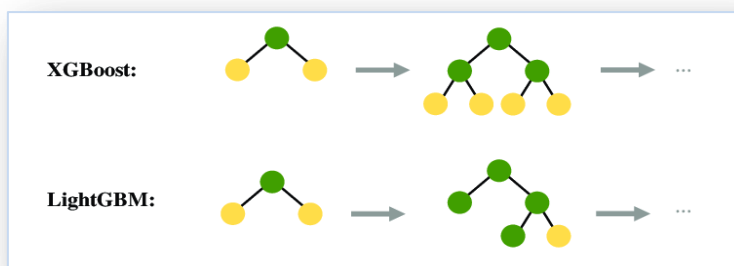


Figure IV.2 – la différence entre lightgbm et xgboost

- Il n'y a pas de vainqueur au niveau de la performance, les deux modèles sont très performants, par contre XGBoost est mieux adapté aux jeux de données de petite taille, on risque **d'avoir un sur-apprentissage** avec de grands volumes de données tandis que LightGBM est son opposé.
- LightGBM est **beaucoup plus rapide** que XGBoost. Grâce à sa méthode de réduction de dimension (EFB), il est capable de gagner en matière de puissance de calcul tout en conservant la même précision de XGBoost.

II.4 Métrique d'évaluation

L'apprentissage supervisé utilise une partie des données pour calculer un modèle de décision qui sera généralisé sur l'ensemble du reste de l'espace. Il est très important d'avoir des mesures permettant de qualifier le comportement du modèle appris sur les données non utilisées lors de l'apprentissage. Ces métriques sont calculées soit sur les exemples d'entraînement eux mêmes ou sur des exemples réservés d'avance pour les tests.

La métrique intuitive utilisée est la précision du modèle appelée aussi le taux de reconnaissance. Elle représente le rapport entre le nombre de donnée correctement classées et le nombre total des données testées.

$$p = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

Généralement, la précision est donnée sous forme de pourcentage ce qui nécessite de multiplier la précision de l'équation précédente par 100.

II.5 Jeux des données pour les modèles de détection et classification

Nous avons besoin d'un ensemble de données d'entraînement et de test pour la détection et la classification de la nature de la consommation d'électricité et du gaz.

II.5.1 Structuration de la base de données

Nous avons téléchargé la base de données (big data) à partir de la plateforme Zindi qui est une plateforme de concours de data sciences dédiée à aider à la résolution des problèmes urgents en rassemblant une communauté de data scientists qui collaborent et sont en compétition pour trouver les meilleures solutions possibles.

La plateforme Zindi coordonne un groupe de plus de 4 000 scientifiques en Afrique qui peuvent s'inscrire à un concours, soumettre leurs ensembles de solutions et gagner le concours - pour un prix en cash <https://zindi.africa/>. Zindi est en collaboration avec la STEG (Société Tunisienne d'Electricité et de Gaz) intitulé « **Détection de fraude dans la consommation d'électricité et du gaz** ».

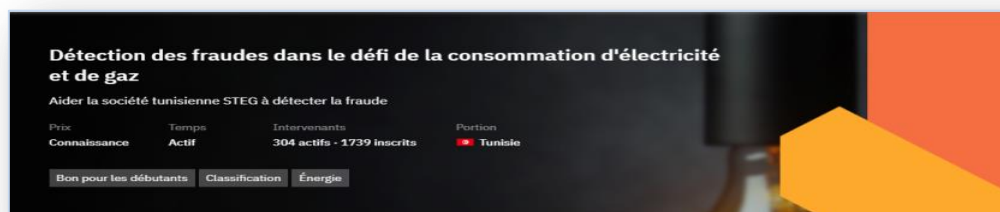


Figure IV.3 – Compétitions Zindi détection de fraude dans la consommation d'électricité et du gaz

A titre d'information, la Société Tunisienne de l'Electricité et du Gaz (STEG) est une entreprise publique et non administrative, elle est chargée d'acheminer l'électricité et le gaz sur l'ensemble de la

Tunisie. L'entreprise a subi des pertes énormes de l'ordre de 200 millions de dinars tunisiens en raison de manipulations frauduleuses des compteurs par les consommateurs.

A partir de l'historique de facturation du client, le challenge vise à détecter et reconnaître les clients impliqués dans des activités frauduleuses. La solution augmentera les revenus de l'entreprise et réduira les pertes causées par ces activités frauduleuses.

Les données fournies par la STEG sont composées de deux fichiers. Le premier est composé de données client et le second contient l'historique de facturation.

Nous trouvons les éléments suivants : (1). train.zip et (2). test.zip et (3). SampleSubmission.csv. Dans chaque fichier .zip, il existe un fichier client et facture.

- Le dossier train

Le train contient la cible. Il s'agit de l'ensemble de données que nous utilisons pour entraîner notre modèle.

- Client_train.csv : dans ce fichier il se trouve les Informations client dans la partie apprentissage contenant un total de **135.493,00** clients.

- Invoice_train.csv : Les différentes factures des clients appartenant à la base d'apprentissage avec un total de **4.476.749,00** lignes de données.

- Un autre dossier (Test) contenant aussi deux Fichiers

Test ressemble à Train.csv mais sans les colonnes liées à la cible. Il s'agit de l'ensemble de données sur lequel nous appliquons notre modèle.

- Client_test.csv : Informations client test avec un total de **58.069,00** clients.

- Invoice_test.csv : Factures client dans l'ensemble de test (**1.939.730,00**).

- SampleSubmission.csv

C'est un exemple d'un fichier de soumission dont nous devons suivre, avec la colonne "ID" reflétant celle de Test.csv et la colonne "cible(Target)" contenant nos prédictions. L'ordre des lignes n'a pas d'importance mais les noms de l'ID doivent être corrects.

Dans la figure qui suit, nous vous montrons par exemple à quoi ressemble un fichier SampleSubmission.csv :

	A	B	C	D
1	client_id,target			
2	test_Client_0,0.9572805294000105			
3	test_Client_1,0.9964246849731611			
4	test_Client_10,0.6123585962704688			
5	test_Client_100,0.7769328506273416			
6	test_Client_1000,0.5710463934651373			
7	test_Client_10000,0.38483374725255304			
8	test_Client_10001,0.7887840424602702			
9	test_Client_10002,0.12359570742670423			
10	test_Client_10003,0.7650333851379789			
11	test_Client_10004,0.41621598858862463			
12	test_Client_10005,0.315294604149003			
13	test_Client_10006,0.8342876122222105			
14	test_Client_10007,0.3752079937654629			
15	test_Client_10008,0.09280253840637187			
16	test_Client_10009,0.44266808111681977			
17	test_Client_1001,0.15550137038819578			
18	test_Client_10010,0.3124628259172776			
19	test_Client_10011,0.7567871072183201			
20	test_Client_10012,0.11839859505391737			
21	test_Client_10013,0.8514200483572812			
22	test_Client_10014,0.11887654577526241			
23	test_Client_10015,0.04928623830855483			
24	test_Client_10016,0.31640441371387884			
25	test_Client_10017,0.6292729829716683			

Figure IV.4 – Aperçu du fichier SampleSubmission.csv

II.5.2 Définitions des structures de données

Client	Donnée	Description
	Client_id	Identifiant unique pour le client
	District	District où se trouve le client
	Client_catg	Catégorie à laquelle appartient le client
	Région	Zone où se trouve le client
	Creation_date	Date à laquelle le client a rejoint
	Target	fraude : 1, pas de fraude : 0

Tableau IV.1 – Description de la structure « Client »

Facturation	Donnée	Description
	Client_id	Identifiant unique pour le client
	Invoice_date	Date de la facture
	Tarif_type	Type de taxe
	Counter_number	Le numéro de compteur
	Counter_statue	Prend jusqu'à 5 valeurs telles que fonctionne bien, ne fonctionne pas, statue en attente, etc.
	Counter_code	Modèle du compteur

Reading_remarque	Note que l'agent STEG prend lors de sa visite chez le client (ex : Si le compteur affiche quelque chose d'anormal, l'agent donne une mauvaise note)
Counter_coefficient	Un coefficient supplémentaire à ajouter lorsque la consommation standard est dépassée
Consommation_level_1	Niveau_de_consommation_1
Consommation_level_2	Niveau_de_consommation_2
Consommation_level_3	Niveau_de_consommation_3
Consommation_level_4	Niveau_de_consommation_4
Old_index	Ancien index
New_index	Nouvel index
Months_number	Numéro du mois
Counter_type	Type de compteur (ELEC/GAZ)

Tableau IV.2 – Description de la structure « Facturation »

II.6 Partitionnement des données

Dans le cadre de notre approche, nous utiliserons la technique de validation croisée pour partitionner les données d'entraînement et de test en k-fold. Les modèles LightGBM et XGBoost seront ensuite entraînés sur les sous-ensembles de données obtenus. Par la suite, nous sélectionnerons le meilleur modèle en évaluant différentes métriques telles que la matrice de confusion, l'exactitude

(accuracy), etc. Une fois le modèle choisi, nous procéderons à la prédiction sur les données de test fournies par Zindi. Enfin, nous enverrons les résultats obtenus à la plate-forme Zindi pour évaluation.

III. APPLICATION DU PROCESSUS D'ECD DANS NOTRE APPROCHE

Les différentes étapes d'extraction de connaissances sont généralement intégrées dans le flux global de traitement des données pour préparer les données, entraîner les modèles et évaluer leur performance. Voici comment ces étapes peuvent être utilisées dans ce contexte :

① Avant d'entraîner les modèles LightGBM et XGBoost

Intégration des données : Si les données sources proviennent de différentes sources, l'étape d'intégration des données peut être effectuée pour fusionner les différents jeux de données en un seul, cohérent et complet (jointure).

Transformation des données : Avant d'entraîner les modèles LightGBM et XGBoost, les données peuvent nécessiter une transformation pour les préparer à l'apprentissage automatique. Cela peut inclure la normalisation des valeurs, la conversion des variables catégorielles en variables numériques, et d'autres transformations nécessaires pour satisfaire les exigences des modèles.

Évaluation et validation : La validation croisée peut être utilisée pour évaluer les performances des modèles sur différents sous-ensembles de données.

② Lors de l'application des modèles LightGBM et XGBoost

Une fois que les données ont été nettoyées et transformées, l'étape de data mining consiste à appliquer les modèles LightGBM et XGBoost sur les données pour extraire des informations utiles et détecter les schémas de fraude.

③ Après l'entraînement des modèles LightGBM et XGBoost

Évaluation et validation : L'évaluation des performances des modèles se fait généralement en utilisant des métriques telles que l'exactitude (accuracy), la précision, le rappel, etc. Ces mesures permettent de quantifier l'efficacité des modèles dans la détection de la fraude.

IV. EXPERIMENTATION ET RESULTATS

Dans l'objectif de simplifier notre démarche empirique en privilégiant un apprentissage supervisé qui opère sur des données dont les variables à prédire sont labellisées en fraude ou non-fraude. Le travail s'effectue en langage de programmation Python via la plateforme Jupyter Notebook et accessoirement la plateforme Google Colab pour la puissance de calcul (nous verrons les détails de ces outils en Annexe 1).

IV.1 Architecture globale

La figure suivante illustre l'architecture générale de la solution retenue de notre système.

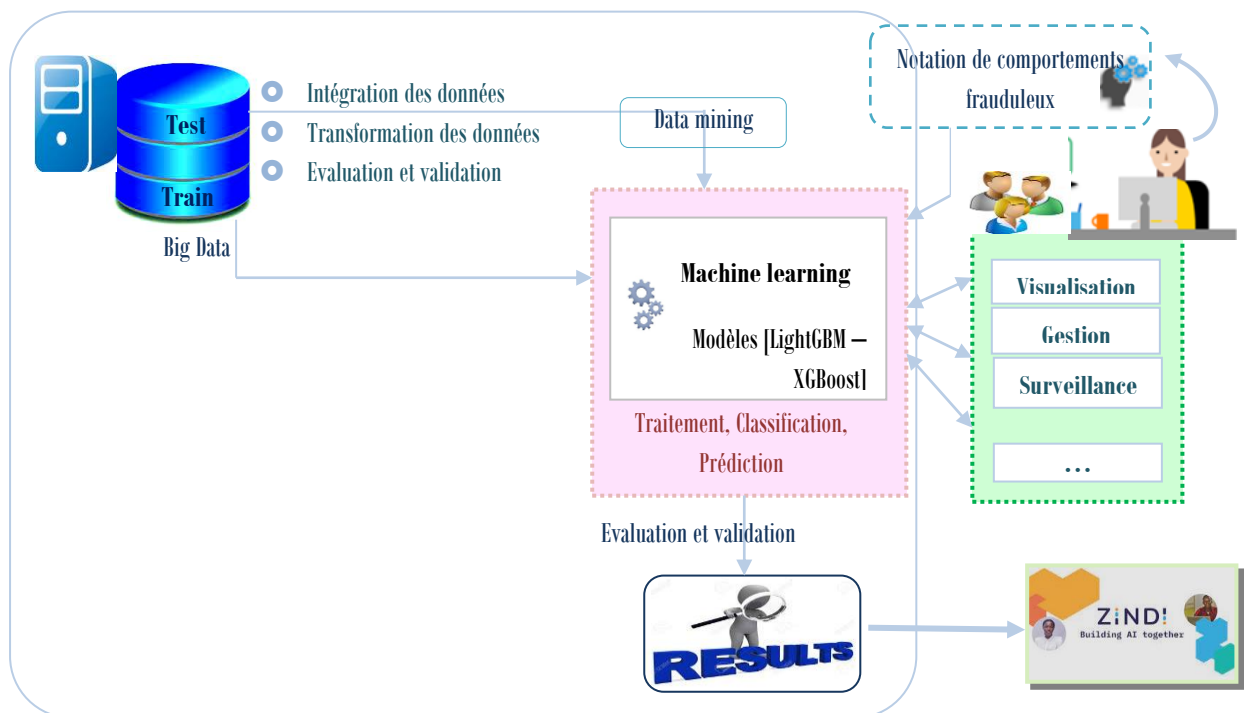


Figure IV.5 – Architecture globale du système.

IV.2 Application des modèles (XGBoost – Lightgbm)

Nous décrivons dans cette section le déroulement détaillé des deux modèles (XGBoost – Lightgbm), puis nous ferons une comparaison entre les deux dans la section qui suivra afin de motiver et valider notre choix porté sur le modèle Lightgbm. Par la suite, nous passons à la généralisation du modèle choisi.

1. Importation des bibliothèques

```
import numpy as np
import pandas as pd
import datetime
import gc
import matplotlib.pyplot as plt
import seaborn as sns
import lightgbm as lgb
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings('ignore')
np.random.seed(4590)
```

Figure IV.6 – Importation des bibliothèques

2. Chargement des datasets

Lorsque nous exécutons ces lignes de code, cela nous permet de monter notre espace de stockage Google Drive dans l'environnement de Google Colab. Cela signifie que nous pouvons accéder aux données présentes dans notre Google Drive directement à partir de notre espace de travail Colab.

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Figure IV.7 – Accès à google drive

Il convient maintenant de charger les données

```
train_client=pd.read_csv('/content/drive/My Drive/data/STEG/client_train.csv')
train_invoice=pd.read_csv('/content/drive/My Drive/data/STEG/invoice_train.csv')
test_client=pd.read_csv('/content/drive/My Drive/data/STEG/client_test.csv')
test_invoice=pd.read_csv('/content/drive/My Drive/data/STEG/invoice_test.csv')
sub=pd.read_csv('/content/drive/My Drive/data/STEG/SampleSubmission.csv')
```

Figure IV.8 – Chargement des données

Il s'agit d'une lecture des données à partir des fichiers CSV :

- **train_client**= pd.read_csv('/content/drive/My Drive/data/STEG/client_train.csv') : ce fichier contient les informations d'apprentissage (Train) sur les clients de STEG.
- **train_invoice**=pd.read_csv('/content/drive/My Drive/data/STEG /invoice_train.csv'): ce fichier contient l'historique de consommation de chaque client à partir de 2005.
- **test_client**=pd.read_csv('/content/drive/My Drive/data/STEG /client_test.csv'):ce fichier contient les données Test des client sans la classe résultat (target).
- **test_invoice**=pd.read_csv('/content/drive/My Drive/data/STEG /invoice_test.csv'):ce fichier contient l'historique de consommation des client Test.

3. Affichage des dataframes importées

- ✓ **train_client**: affiche les cinq premières et cinq dernières lignes du fichier client, qui contient un total de 135 492 lignes de six (06) attributs.

```
#Afficher le contenu du fichier importé "train_client"
train_client
```

	disrict	client_id	client_catg	region	creation_date	target
0	60	train_Client_0	11	101	31/12/1994	0.0
1	69	train_Client_1	11	107	29/05/2002	0.0
2	62	train_Client_10	11	301	13/03/1986	0.0
3	69	train_Client_100	11	105	11/07/1996	0.0
4	62	train_Client_1000	11	303	14/10/2014	0.0
...
135488	62	train_Client_99995	11	304	26/07/2004	0.0
135489	63	train_Client_99996	11	311	25/10/2012	0.0
135490	63	train_Client_99997	11	311	22/11/2011	0.0
135491	60	train_Client_99998	11	101	22/12/1993	0.0
135492	60	train_Client_99999	11	101	18/02/1986	0.0

135493 rows x 6 columns

Figure IV.9 – Afficher les cinq premières et cinq dernières lignes du fichier client.

- ✓ **train_client.info()**: afficher le détail du fichier importé des clients « train_client » comme le nombre d'enregistrement, le nombre des colonnes, type des champs,... etc.

```
#Informations sur les colonnes du fichier "train_client"
train_client.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 135493 entries, 0 to 135492
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   disrict         135493 non-null int64
1   client_id      135493 non-null object
2   client_catg    135493 non-null int64
3   region         135493 non-null int64
4   creation_date  135493 non-null object
5   target         135493 non-null float64
dtypes: float64(1), int64(3), object(2)
memory usage: 6.2+ MB
```

Figure IV.10 – Afficher le détail du fichier importé des clients.

- ✓ **train_invoice**: afficher les cinq premières lignes et cinq dernières lignes du fichier des factures, qui contient un total de 4 476 749 lignes de seize (16) colonnes.

```
#Afficher le contenu du fichier importé "train_invoice"
train_invoice
```

	client_id	invoice_date	tarif_type	counter_number	counter_statue	counter_code	reading_remarque	counter_coefficient	consommation_level_1	consommation_level_2
0	train_Client_0	2014-03-24	11	1335667	0	203	8	1	82	0
1	train_Client_0	2013-03-29	11	1335667	0	203	6	1	1200	184
2	train_Client_0	2015-03-23	11	1335667	0	203	8	1	123	0
3	train_Client_0	2015-07-13	11	1335667	0	207	8	1	102	0
4	train_Client_0	2016-11-17	11	1335667	0	207	9	1	572	0
...
4476744	train_Client_99998	2005-08-19	10	1253571	0	202	9	1	400	135
4476745	train_Client_99998	2005-12-19	10	1253571	0	202	6	1	200	6
4476746	train_Client_99999	1996-09-25	11	560948	0	203	6	1	259	0
4476747	train_Client_99999	1996-05-28	11	560948	0	203	6	1	603	0
4476748	train_Client_99999	1996-01-25	11	560948	0	203	6	1	516	0

4476749 rows x 16 columns

Figure IV.11 – Afficher les cinq premières et cinq dernières lignes du fichier des factures.

- ✓ **train_invoice.count()** : affiche le nombre des factures.

```
#Nombre des factures
train_invoice.count()
```

client_id	4476749
invoice_date	4476749
tarif_type	4476749
counter_number	4476749
counter_statue	4476749
counter_code	4476749
reading_remarque	4476749
counter_coefficient	4476749
consommation_level_1	4476749
consommation_level_2	4476749
consommation_level_3	4476749
consommation_level_4	4476749
old_index	4476749
new_index	4476749
months_number	4476749
counter_type	4476749
dtype: int64	

Figure IV.12 – Afficher le nombre des factures.

- ✓ Nous pouvons afficher les cinq premières lignes du fichier "test_client" à l'aide de la méthode **test_client.head()** et fournir des détails sur le fichier, y compris le nombre de lignes et de colonnes, les types de colonnes et la mémoire utilisée, en utilisant la méthode **test_client.info()**.

```
#Afficher l'entête du fichier importé "test_client"
test_client.head()
```

	disrict	client_id	client_catg	region	creation_date
0	62	test_Client_0	11	307	28/05/2002
1	69	test_Client_1	11	103	06/08/2009
2	62	test_Client_10	11	310	07/04/2004
3	60	test_Client_100	11	101	08/10/1992
4	62	test_Client_1000	11	301	21/07/1977

Figure IV.13 – Afficher les cinq premières lignes du fichier "test_client".

```
#Afficher les détails du fichier importé "test_client"
test_client.info()
```

```
<<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58069 entries, 0 to 58068
Data columns (total 5 columns):
# Column Non-Null Count Dtype
---  ---
0 disrict 58069 non-null int64
1 client_id 58069 non-null object
2 client_catg 58069 non-null int64
3 region 58069 non-null int64
4 creation_date 58069 non-null object
dtypes: int64(3), object(2)
memory usage: 2.2+ MB
```

Figure IV.14 – Afficher les détails du fichier "test_client".

- ✓ **Sub.head()** : affiche les cinq premières lignes du fichier de soumission (sub) sur Zindi, qui devrait inclure les clients et le taux de fraude de chaque client.

```
sub.head()
```

	client_id	target
0	test_Client_0	0.957281
1	test_Client_1	0.996425
2	test_Client_10	0.612359
3	test_Client_100	0.776933
4	test_Client_1000	0.571046

Figure IV.15 – Afficher les cinq premières lignes du fichier de soumission (sub) sur Zindi.

4. Fusion du fichier clients avec factures (application de l'étape intégration des données du processus ECD)

Dans cette étape, nous devons fusionner les deux fichiers "client_train" et "invoice_train" afin d'obtenir les 21 attributs (5 attributs du fichier client_train + 16 attributs du fichier invoice_train avec un attribut en commun) dans une seule ligne pour chaque client. L'objectif est de représenter graphiquement les données :

```
#joiture des données client_train et invoice_train
data = pd.merge(train_client,train_invoice, on='client_id', how='left')
```

Figure IV.16 – Jointure des données client_train et invoice_train

```
#Afficher le résultat de la jointure
data
```

	disrict	client_id	client_catg	region	creation_date	target	invoice_date	tarif_type	counter_number	counter_statue	...	reading_remarque	counter_coefficient	cc
0	60	train_Client_0	11	101	31/12/1994	0.0	2014-03-24	11	1335667	0	...	8	1	
1	60	train_Client_0	11	101	31/12/1994	0.0	2013-03-29	11	1335667	0	...	6	1	
2	60	train_Client_0	11	101	31/12/1994	0.0	2015-03-23	11	1335667	0	...	8	1	
3	60	train_Client_0	11	101	31/12/1994	0.0	2015-07-13	11	1335667	0	...	8	1	
4	60	train_Client_0	11	101	31/12/1994	0.0	2016-11-17	11	1335667	0	...	9	1	
...	
4476744	60	train_Client_99998	11	101	22/12/1993	0.0	2005-08-19	10	1253571	0	...	9	1	
4476745	60	train_Client_99998	11	101	22/12/1993	0.0	2005-12-19	10	1253571	0	...	6	1	
4476746	60	train_Client_99999	11	101	18/02/1986	0.0	1996-09-25	11	560948	0	...	6	1	
4476747	60	train_Client_99999	11	101	18/02/1986	0.0	1996-05-28	11	560948	0	...	6	1	
4476748	60	train_Client_99999	11	101	18/02/1986	0.0	1996-01-25	11	560948	0	...	6	1	

4476749 rows x 21 columns

Figure IV.17 – Affichage du résultat dans 'data'.

5. Représentation graphique des données

Nous avons effectué dans cette phase une analyse visuelle de la répartition des clients par rapport aux différentes variables afin de mieux comprendre le rôle de chaque variable. Notre objectif était d'observer la dispersion des clients en fonction de ces attributs.

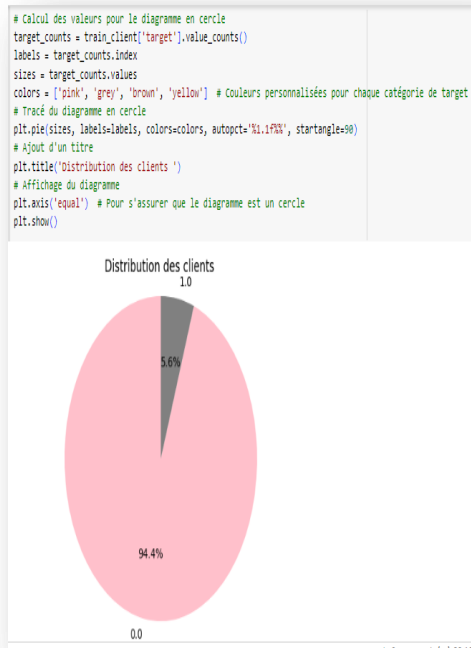


Figure IV.18a– Distribution des clients (frauduleux (=1), non-frauduleux (=0)) par un diagramme en cercle

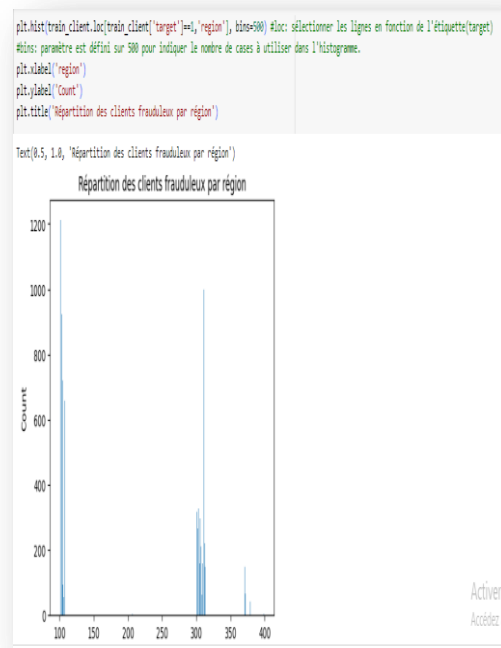


Figure IV.18b – Répartition des clients frauduleux par région représentée par un histogramme

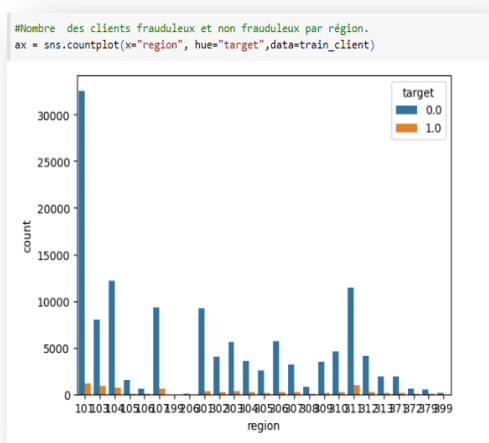


Figure IV.19a – Le nombre des clients (frauduleux (=1), non-frauduleux (=0)) par «Région »



Figure IV.19b – Le nombre des clients (frauduleux (=1), non frauduleux (=0)) par «District»

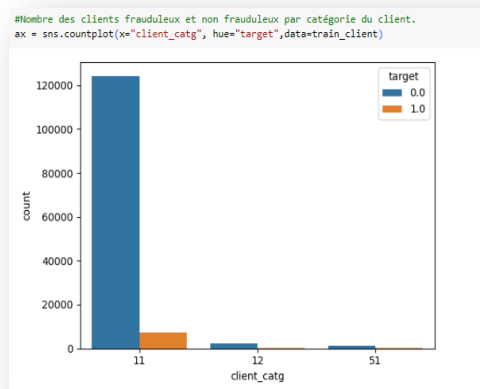


Figure IV.20a – Le nombre des clients (frauduleux (=1), non-frauduleux (=0)) par « catégorie du client »

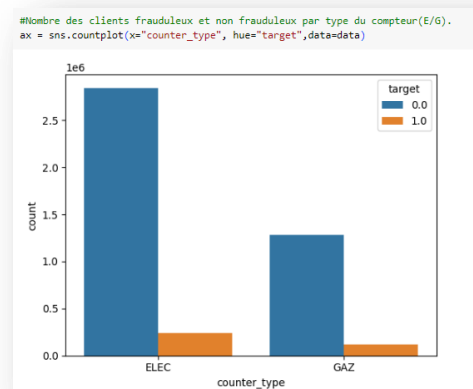


Figure IV.20b – Le nombre des clients (frauduleux (=1), non- frauduleux (=0)) par «type de compteur»

6. Formater les dates (date création abonné, date facture) : (application de l'étape transformation des données du processus ECD)

Le codage des données est le processus de conversion des attributs qualitatifs en valeurs quantitatives pour permettre l'utilisation de méthodes et d'algorithmes de calcul. Dans ce cas, la date de création de l'abonné est formatée en deux champs entiers, "year" et "month", à partir des fichiers importés "train_invoice", "test_invoice", "train_client" et "test_client".

```
5] #Convertir la date de facture en entier est transformer la date en deux colonne(year,month)
for df in [train_invoice,test_invoice]:
    df['invoice_date'] = pd.to_datetime(df['invoice_date'])
    df['year'] = df['invoice_date'].dt.year
    df['month'] = df['invoice_date'].dt.month
```

Figure IV.21 – Convertir la date de facture en entier.

```
6] #Convertir la date de création en entier est transformer la date en deux colonne(year,month)
for df in [train_client,test_client]:
    df['creation_date'] = pd.to_datetime(df['creation_date'])
    df['year'] = df['creation_date'].dt.year
    df['month'] = df['creation_date'].dt.month
```

Figure IV.22 – Convertir la date de création en entier.

7. Convertir le type des compteurs (ELEC, GAZ) en entier (0,1) :(application de l'étape transformation des données du processus ECD)

```
#Conversion de type du compteur en entier (0,1)
d={"ELEC":0,"GAZ":1}
train_invoice['counter_type']=train_invoice['counter_type'].map(d)
d={"ELEC":0,"GAZ":1}
test_invoice['counter_type']=test_invoice['counter_type'].map(d)
```

Figure IV.23 – Convertir le type des compteurs (ELEC, GAZ) en entier (0, 1).

8. Définir la fonction d'agrégation 'aggs'

Cette fonction sert à regrouper les factures par client et calculer les variables ci-dessous pour les champs concernés :

```
#Définir la fonction d'agrégation avec les paramètres (sum,max,min,mean,nunique) sur les colonnes ci-dessous
aggs = {}
aggs['consommation_level_1'] = ['sum','max','min','mean']
aggs['consommation_level_2'] = ['sum','max','min','mean']
aggs['consommation_level_3'] = ['sum','max','min','mean']
aggs['consommation_level_4'] = ['sum','max','min','mean']

aggs['month'] = ['mean','max','min']
aggs['year'] = ['nunique','max','min','mean']

aggs['months_number'] = ['max','min','mean','sum']
aggs['reading_remarque'] = ['max','min','mean','sum']
aggs['counter_coefficient'] = ['max','min','mean','sum']
aggs['counter_number'] = ['nunique','max','min']
aggs['counter_type'] = ['nunique']
aggs['counter_statue'] = ['nunique']
aggs['tarif_type'] = ['nunique','max','min','sum']
aggs['counter_code'] = ['nunique','max','mean','min']

aggs['old_index'] = ['nunique','mean','max','min']
aggs['new_index'] = ['nunique','mean','max','min']
```

Figure IV.24 – Définir la fonction d'agrégation 'aggs'.

```
[ ] #Appliquer l'agrégation
agg_train = train_invoice.groupby(['client_id']).agg(aggs)
agg_test = test_invoice.groupby(['client_id']).agg(aggs)
```

Figure IV.25 – Afficher l'agrégation.

9. Afficher les résultats d'agrégation

```
#Afficher les résultats d'agrégation
agg_train
```

client_id	consommation_level_1_sum	consommation_level_1_max	consommation_level_1_min	consommation_level_1_mean	consommation_level_2_sum	consommation_level_2_max	consommation_level_2_min	consommation_level_2_mean
train_Client_0	12334	1200	38	352.400000	370	0	0	0
train_Client_1	20629	1207	190	557.540541	0	0	0	0
train_Client_10	14375	2400	168	798.611111	682	0	0	0
train_Client_100	24	15	0	1.200000	0	0	0	0
train_Client_1000	9292	800	124	663.714286	1468	0	0	0
...
train_Client_99995	139	139	0	1.957746	0	0	0	0
train_Client_99996	7620	800	0	185.853659	31	0	0	0

Figure IV.26 – Afficher les résultats d'agrégation.

10. Affichage des champs

```
[ ] #Afficher les champs calculés
agg_train.info()

<class 'pandas.core.frame.DataFrame'>
Index: 135493 entries, train_Client_0 to train_Client_99999
Data columns (total 56 columns):
#   Column                                     Non-Null Count  Dtype
---  ---
0   consommation_level_1_sum                 135493 non-null  int64
1   consommation_level_1_max                 135493 non-null  int64
2   consommation_level_1_min                 135493 non-null  int64
3   consommation_level_1_mean                135493 non-null  float64
4   consommation_level_2_sum                 135493 non-null  int64
5   consommation_level_2_max                 135493 non-null  int64
6   consommation_level_2_min                 135493 non-null  int64
7   consommation_level_2_mean                135493 non-null  float64
8   consommation_level_3_sum                 135493 non-null  int64
9   consommation_level_3_max                 135493 non-null  int64
10  consommation_level_3_min                 135493 non-null  int64
11  consommation_level_3_mean                135493 non-null  float64
12  consommation_level_4_sum                 135493 non-null  int64
13  consommation_level_4_max                 135493 non-null  int64
14  consommation_level_4_min                 135493 non-null  int64
15  consommation_level_4_mean                135493 non-null  float64
```

Figure IV.27 – Afficher les champs de agg_train.

11. Fusionner client_train et invoice_train :(application de l'étape intégration des données du processus ECD)

Faire la jointure à gauche entre client_train et invoice_train et client_test et invoice_test

```
[ ] #Faire la jointure à gauche entre client_train et invoice_train et client_test et invoice_test
train = pd.merge(train_client,agg_train, on='client_id', how='left')
test = pd.merge(test_client,agg_test, on='client_id', how='left')
```

Figure IV.28 – Jointure à gauche entre client_train et invoice_train et client_test et invoice_test.

Extraction de la colonne "target" de la base d'entraînement et la transformer en une matrice à une dimension pour pouvoir l'utiliser directement dans la classification.

```
#Récupérer la colonne target sur une matrice à une dimension
target=train['target']
```

12. Convertir la colonne client_id sur train et test en entier :(application de l'étape transformation des données du processus ECD)

```
1 #conversion le champ client_id sur train et test en entier
2 from sklearn import preprocessing
3 lbl = preprocessing.LabelEncoder()
4 lbl.fit(list(train['client_id'].values))
5 train['client_id'] = lbl.transform(list(train['client_id'].values))
6 #test
7 lbl.fit(list(test['client_id'].values))
8 test['client_id'] = lbl.transform(list(test['client_id'].values))
```

Figure IV.29 – Conversion du champ client_id sur train et test en entier.

13. Modèle d'Apprentissage (Classification)

a). Séparation des données

Dans l'optique d'obtenir une évaluation précise des performances de notre système, nous avons décidé d'adopter la méthode de validation croisée. Cette approche consiste à partitionner notre jeu de données d'entraînement et de test en cinq sous-ensembles équilibrés (kfold=5), afin d'assurer une représentativité et une robustesse statistique optimales. Par la suite, les modèles XGBoost et LightGBM ont été entraînés et évalués en utilisant ces cinq sous-ensembles distincts de données d'entraînement et de test.

À chaque itération de la validation croisée, l'un des sous-ensembles a été utilisé comme ensemble de test, tandis que les quatre autres ont été regroupés pour former l'ensemble de données d'entraînement. Ainsi, chaque sous-ensemble a été successivement utilisé comme ensemble de test, permettant aux modèles d'être évalués de manière équitable sur l'ensemble du jeu de données.

```
[ ] #Définir la validation croisée avec k=5
    random_state=0

    kf = KFold(n_splits=5, shuffle=True, random_state=random_state)
    for train_index, test_index in kf.split(X, y):
        X_train = X.iloc[train_index]
        X_test = X.iloc[test_index]

        y_train = y.iloc[train_index]
        y_test = y.iloc[test_index]
        print('Train',X_train.shape,y_train.shape,'Test',X_test.shape,y_test.shape)

Train (108394, 62) (108394,) Test (27099, 62) (27099,)
Train (108394, 62) (108394,) Test (27099, 62) (27099,)
Train (108394, 62) (108394,) Test (27099, 62) (27099,)
Train (108395, 62) (108395,) Test (27098, 62) (27098,)
Train (108395, 62) (108395,) Test (27098, 62) (27098,)
```

Figure IV.30 – Définir la validation croisée avec k=5.

b). Paramétrage des modèles

Nous avons effectué une exploration systématique des hyperparamètres en utilisant une méthode appelée recherche en grille pour les modèles LightGBM et XGBoost. Cette approche nous a permis de trouver les meilleures combinaisons de paramètres pour chaque modèle.

La recherche en grille d'hyperparamètres consiste à définir une grille de valeurs pour chaque hyperparamètre à tester. Nous avons défini ces grilles en prenant en compte les hyperparamètres pertinents tels que le taux d'apprentissage, la profondeur maximale de l'arbre, le nombre d'estimateurs, etc.

```

▶ # Définir les paramètres à tester pour le modèle LightGBM
from sklearn.model_selection import GridSearchCV
param_grid = {
    'learning_rate': [0.1, 0.01, 0.001],
    'max_depth': [2, 4, 6, 8],
    'n_estimators': [50, 200, 300, 350],
    'reg_lambda': [1, 10, 100],
    'subsample': [0.5, 0.75, 1.0]
}
# Initialiser le modèle LightGBM
lgbm = LGBMClassifier()
# Créer une instance de la recherche de grille
grid_search = GridSearchCV(estimator=lgbm, param_grid=param_grid, cv=5)

# Exécuter la recherche de grille sur les données d'entraînement
grid_search.fit(X_train, y_train)

# Afficher les meilleurs paramètres et le score de la recherche de grille
print("Meilleurs paramètres: ", grid_search.best_params_)
print("Meilleur score: ", grid_search.best_score_)

Meilleurs paramètres: {'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 300, 'reg_lambda': 100, 'subsample': 0.5}
Meilleur score: 0.9476820886572259

```

Figure IV.31 – Les paramètres à tester pour le modèle Lightgbm

Les meilleurs paramètres pour lightgbm : learning_rate= 0.1, max_depth= 4 et n_estimators= 300.

```

[ ] # Définir les paramètres à tester pour le modèle XGBoost
from sklearn.model_selection import GridSearchCV
import xgboost as xgb
from xgboost import XGBClassifier
# définir les paramètres à tester
param_grid = {
    'max_depth': [2, 4, 6, 8],
    'learning_rate': [0.1, 0.01, 0.001],
    'n_estimators': [50, 200, 300, 350]
}

# créer un classificateur XGBoost
xgb = XGBClassifier()

# effectuer une recherche de grille en utilisant une validation croisée de 5 plis
grid_search = GridSearchCV(estimator=xgb, param_grid=param_grid, cv=5)

# adapter le modèle aux données d'entraînement
grid_search.fit(X_train, y_train)

# afficher les meilleurs paramètres et le score
print("Best parameters: ", grid_search.best_params_)
print("Best score: ", grid_search.best_score_)

Best parameters: {'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 350}
Best score: 0.9477743438350478

```

Figure IV.32 – Les paramètres à tester pour le modèle Xgboost.

Les meilleurs paramètres pour xgboost : learning_rate= 0,01, max_depth= 8 et n_estimators= 350.

c). Définir les modèles LGBMClassifier et XGBClassifier comme modèles d'apprentissage : (application de l'étape data mining du processus ECD)

```
# Création du modèle LightGBM avec les meilleurs paramètres générés par la recherche de grille
import lightgbm as lgb
from lightgbm import LGBMClassifier
model = lgb.LGBMClassifier(boosting_type='gbdt', objective='binary', metric = 'auc',
                           num_leaves=31, learning_rate = 0.1, feature_fraction = 0.9, n_estimators=300, subsample= 0.5,
                           max_depth=4 )
```

Figure IV.33 – Définir le modèle LGBClassifier.

```
# Création du modèle XGBoost avec les meilleurs paramètres générés par la recherche de grille
import xgboost as xgb
from datetime import datetime
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import StratifiedKFold
clf = xgb.XGBClassifier(
    n_estimators=350,
    max_depth=8,
    learning_rate=0.01
)
```

Figure IV.34 – Définir le modèle XGBClassifier.

d). Entraînement des modèles

1 Lightgbm

```
#Faire entraîner avec le modèle 'model'
model.fit(X_train, y_train)
```

[LightGBM] [Warning] feature_fraction is set=0.9, colsample_bytree=1.0 will be ignored. Current value: feature_fraction=0.9

```
LGBMClassifier
LGBMClassifier(feature_fraction=0.9, learning_rate=0.05, metric='auc',
               objective='binary')
```

Figure IV.35 – Entraîner le modèle Lightgbm.

2 Xgboost

```
#Faire entraîner avec le modèle clf
clf.fit(X_train,y_train)
```

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.05, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=8, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=350, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=None, ...)
```

Figure IV.36 – Entraîner le modèle Xgboost

e). Prédiction sur l'échantillon X_test

Après l'entraînement des modèles, la prédiction est effectuée pour classer les nouvelles données. L'échantillon de test est utilisé pour la prédiction.

```
#Prédiction sur l'échantillon X_test
y_pred = model.predict(X_test)
```

Figure IV.37.a – Prédiction du modèle lightgbm.

```
predictions = clf.predict(X_test)
```

Figure IV.37.b – Prédiction du modèle xgboost.

IV.3 Comparaison des modèles et synthèse

a). Mesurer les performances (application de l'étape d'évaluation et validation du processus ECD)

```
[ ] #Matrice de confusion pour LightGBM
cm = confusion_matrix(y_test, y_pred)
print(cm)

[[25413  200]
 [ 1225  260]]
```

Figure IV.38.a – Calculer la Matrice de confusion de lightgbm (cm).

```
[ ] #Matrice de confusion pour XGBoost
CM = confusion_matrix(y_test, predictions)
print(CM)

[[25429  184]
 [ 1219  266]]
```

Figure IV.38.b – Calculer la Matrice de confusion de xgboost (CM).

b). Représentation graphique des deux matrices de confusion

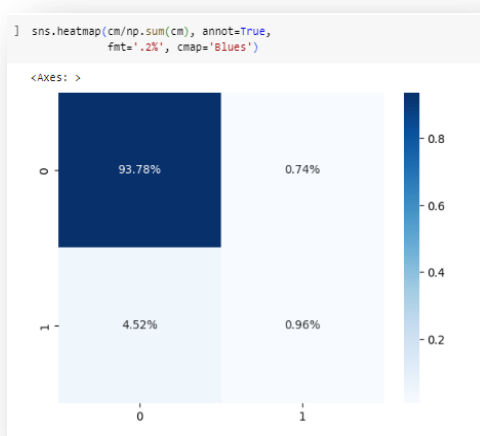


Figure IV.39.a – Matrice de confusion de lightgbm

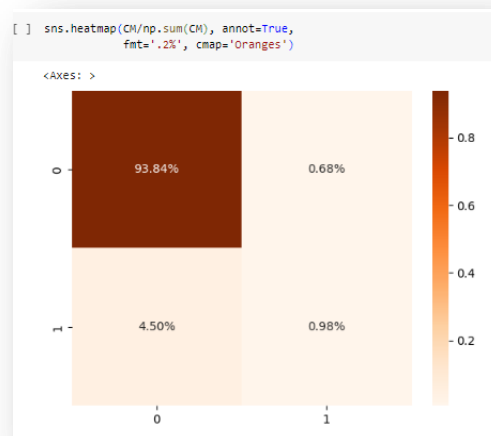


Figure IV.39.b – Matrice de confusion de xgboost

c). Calcul de la précision des deux modèles à partir de la matrice de confusion

```
[ ] print(accuracy_score(y_test, y_pred))
0.9474130932172116
```

Figure IV.40.a – Taux de précision pour lightgbm

```
[ ] print(accuracy_score(y_test, predictions))
0.9482249612517529
```

Figure IV.40.b – Taux de précision pour xgboost

Le taux d'erreur de lightgbm : $T1 = 1 - P1 = 1 - 0.9474 = 0.0526$

Le taux d'erreur de xgboost : $T2 = 1 - P2 = 1 - 0.9482 = 0.0518$

d). Représentation de la courbe ROC

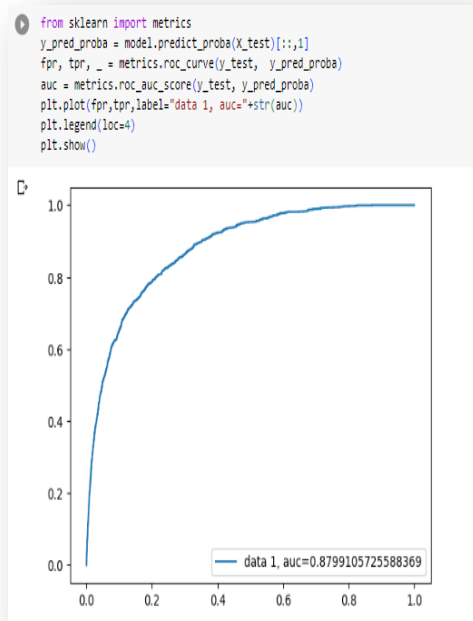


Figure IV.41.a – La courbe ROC de lightgbm

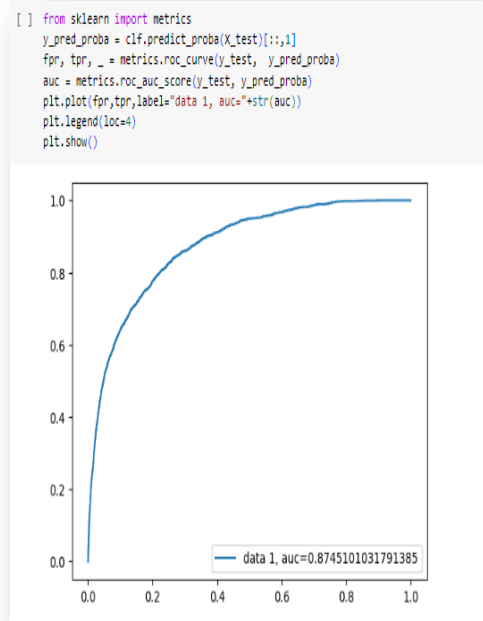


Figure IV.41.b – La courbe ROC de xgboost

Comparaison entre les deux modèles d'après les résultats

Nous récapitulons dans cette section les résultats des performances des deux modèles, LightGBM et XGBoost sur le jeu de données. La comparaison est détaillée dans ce qui suit :

Pour LightGBM

- Matrice de confusion : [[25413, 200], [1225, 260]]
- Accuracy (précision) : 0,9474
- Rappel : 0,1751
- Spécificité : 0,9922
- AUC : 0,8799

Pour XGBoost

- Matrice de confusion : [[25429, 184], [1219, 266]]
- Accuracy (précision) : 0,9482
- Rappel : 0,1791
- Spécificité : 0,9928
- AUC : 0,8745

Performance Modèles	Matrice_confusion	Accuracy	Rappel	Spécificité	AUC
LightGBM	[[25413, 200], [1225, 260]]	0,9474	0,1751	0,9922	0,8799
XGBoost	[[25429, 184], [1219, 266]]	0,9482	0,1791	0,9928	0,8745

Tableau IV.3 – Comparaison entre Lightgbm et Xgboost d'après les résultats obtenus.

En se basant sur ces mesures de performance, nous observons les différences suivantes :

1. *Matrice de confusion* : Les deux modèles ont des matrices de confusion légèrement différentes, mais les erreurs de prédiction sont assez comparables.
2. *Accuracy (précision)* : Les deux modèles ont une précision similaire, avec LightGBM ayant une précision de 0,9474 et XGBoost ayant une précision de 0,9482. Les performances sont proches en termes de précision globale.
3. *Rappel* : LightGBM a un rappel de 0,1751, tandis que XGBoost a un rappel légèrement supérieur de 0,1791. Le rappel mesure la capacité du modèle à identifier les vrais positifs. Dans ce cas, XGBoost a un meilleur rappel.
4. *Spécificité* : LightGBM a une spécificité de 0,9922, tandis que XGBoost a une spécificité légèrement supérieure de 0,9928. La spécificité mesure la capacité du modèle à identifier les vrais négatifs. XGBoost a une légèrement meilleure spécificité.
5. *AUC (Area Under the Curve)* : LightGBM a une AUC de 0,8799, tandis que XGBoost a une AUC légèrement inférieure de 0,8745. L'AUC est une mesure globale de la performance d'un

modèle qui prend en compte à la fois la sensibilité (rappel) et la spécificité. Dans ce cas, LightGBM a une légère avance.

En faisant une synthèse en fonction de ces résultats, il semble que LightGBM ait une performance globalement meilleure que XGBoost dans cette comparaison spécifique (LightGBM a une meilleure AUC). Cependant, il est important de noter que les performances des modèles peuvent varier en fonction des données spécifiques et des paramètres utilisés lors de l'entraînement.

Après avoir fait une analyse approfondie selon les résultats de comparaison obtenus des deux modèles, notre choix est porté sur le modèle Lightgbm pour faire la prédiction en fonction de ses performances supérieures en fonction des métriques.

IV.4 Généralisation du modèle [LightGBM]

```
#Faire entraîner le modèle 'model' avec la totalité des données
model.fit(train,target)

[LightGBM] [Warning] feature_fraction is set=0.9, colsample_bytree=1.0 will be ignored. Current value: feature_fraction=0.9
LGBMClassifier
LGBMClassifier(feature_fraction=0.9, max_depth=6, metric='auc',
               n_estimators=200, objective='binary')
```

Figure IV.42 – Entraîner le modèle avec la totalité des données.

```
] #Tester le modèle sur des nouveaux exemples de zindi
pred = model.predict_proba(test)
print(pred)

[[0.97363631 0.02636369]
 [0.797079 0.202921 ]
 [0.97696319 0.02303681]
 ...
 [0.32442242 0.67557758]
 [0.99117747 0.00882253]
 [0.92614828 0.07385172]]
```

Figure IV.43 – Tester le modèle sur des nouveaux exemples de Zindi.

Pour prédire la probabilité de fraude de chaque abonné, nous avons utilisé la fonction "**predict_proba()**". Le tableau de sortie contient trois colonnes : la première colonne correspond à l'identifiant du client, la deuxième représente la probabilité d'appartenir à la classe "0" (non-frauduleux) et la troisième représente la probabilité d'appartenir à la classe "1" (frauduleux).

```
[ ] y_pred = model.predict_proba(test)
pred = pd.DataFrame(y_pred)
print(pred)
```

	0	1
0	0.973636	0.026364
1	0.797079	0.202921
2	0.976963	0.023037
3	0.998084	0.001916
4	0.926662	0.073338
...
58064	0.998372	0.001628
58065	0.980260	0.019740
58066	0.324422	0.675578
58067	0.991177	0.008823
58068	0.926148	0.073852

[58069 rows x 2 columns]

Figure IV.44 – Prédire la probabilité de fraude de chaque abonné.

V. VALIDATION DES RESULTATS PAR ZINDI

Afin que notre modèle d'apprentissage développé s'intègre dans un référentiel international, nous avons procédé à cette dernière phase dans notre approche, qui est la soumission des résultats de prédiction dans la plateforme Zindi. Ces résultats seront convertis en un fichier csv.

```
submission = pd.DataFrame({
    "client_id": sub["client_id"],
    "target": pred[1]
})
submission.to_csv('/content/drive/My Drive/data/STEG/steg8.csv', index=False)
```

Figure IV.45 – Soumission

Nous avons procédé comme suit :

- 1). Nous avons créé des comptes où nous pouvons nous connecter à la plateforme Zindi.

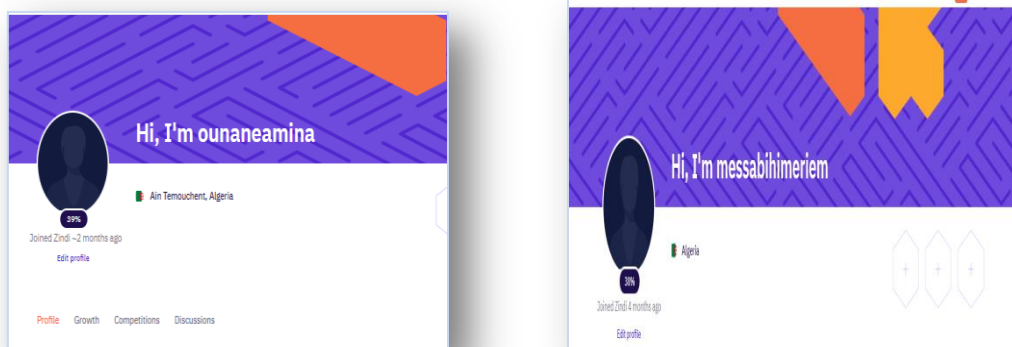


Figure IV.46 – Nos profils sur Zindi

2). Une fois que la connexion est établie, nous accédons à la liste des compétitions et nous sélectionnons le challenge, sujet de notre participation (Fraud Detection in Electricity and Gas Consumption challenge).

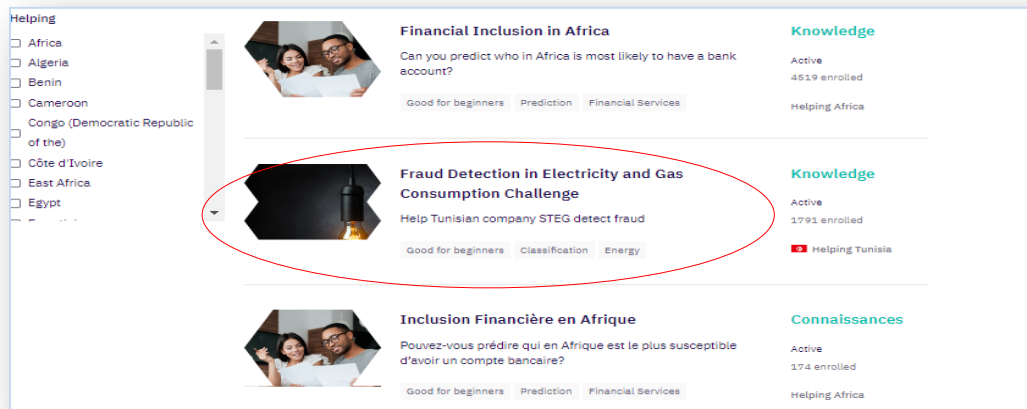


Figure IV.47 – Fraud Detection in Electricity and Gas Consumption Challenge in Zindi

3). Nous devons par la suite cliquer sur Submit pour sélectionner notre fichier de soumission.

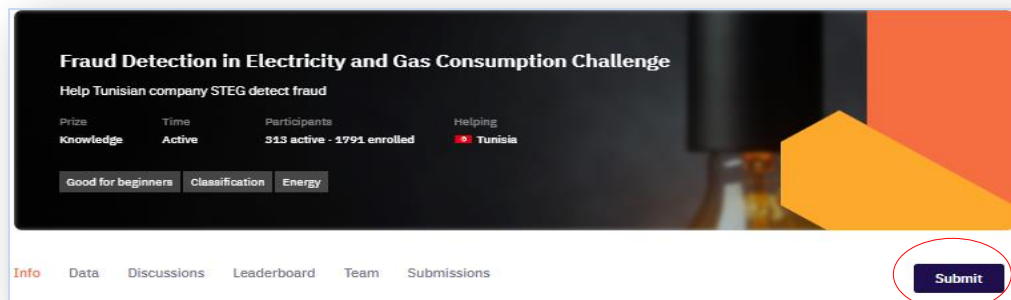


Figure IV.48 – Sélectionner le fichier de soumission

4). Une fois le fichier est soumis, nous pouvons visualiser notre score. Dans notre cas, nous avons eu un score de 88.71%.

<input type="checkbox"/>	snXDyrHY	~2 months ago	ounaneamina	st2023.csv	↓	0.884899085	-
<input type="checkbox"/>	Zby1Gumo	~2 months ago	ounaneamina	st203.csv	↓	0.884899085	-
<input type="checkbox"/>	BDAJfUtG	~2 months ago	ounaneamina	st202.csv	↓	0.885840400	-
<input type="checkbox"/>	EBD1w2f	~2 months ago	ounaneamina	st20.csv	↓	0.879674560	-
<input type="checkbox"/>	isb3eBRZ	~2 months ago	ounaneamina	steg1001.csv	↓	0.880485413	-
<input type="checkbox"/>	ZpMzrD4X	~2 months ago	ounaneamina	steg9.csv	↓	0.857375538	-
<input type="checkbox"/>	5CYmopfY	~2 months ago	ounaneamina	steg8.csv	↓	0.887115461	-
<input type="checkbox"/>	Dt9nCH36	~2 months ago	ounaneamina	steg61.csv	↓	0.885336910	-

Figure IV.49 – Score de soumission

5). Il est également possible de connaître notre classement, cela se fait en cliquant simplement sur le Leader board :



19		messabihimeriem	0.887115461	3 months ago	35
20		ounaneamina	0.887115461	~2 months ago	10

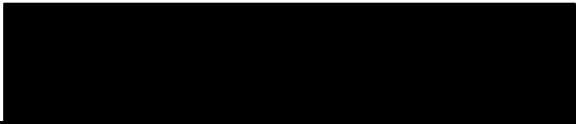
Figure IV.50 – Le classement final sur Zindi

VI. CONCLUSION

Nous avons proposé dans ce chapitre deux approches de classification supervisées pour les données de consommation (normale /anormale) de l'électricité et du gaz. Nous avons détaillé toute la démarche conceptuelle relative à l'élaboration des modèles ainsi que leurs principes de fonctionnement.

Nous avons utilisé le jeu de données de la base de référence Zindi pour le test et l'entraînement. Ces données seront considérées comme les entrées des algorithmes Machine learning développés (XGBoost et LightGbm) afin de résoudre les différents problèmes posés dans ce mémoire, à savoir la classification et la prédiction des consommations ainsi que la détection des anomalies.

Ces deux approches ont été mises œuvre complètement et ont été testées dans le but d'identifier les comportements suspects des consommateurs. A travers des expérimentations et les résultats obtenus, nous avons pu motiver le choix de l'algorithme ML- LightGbm.



CONCLUSION
GENERALE

Conclusion générale

Des pertes importantes sont réalisées chaque année en raison des consommations frauduleuses d'énergie (électricité et gaz). La solution clé pour la réduction de ces pertes réside dans le développement des systèmes fiables de détections et prédiction des consommations anormales. Cependant, la détection des fraudes est une tâche particulièrement difficile et complexe, car les activités frauduleuses sont des événements difficiles à modéliser et en constante évolution.

Les techniques de l'intelligence artificielle et le data mining ont grandement contribué au développement de tels systèmes de détection de fraudes.

Dans cette conclusion générale, nous présentons un sommaire de contributions principales de ce travail. De plus, nous résumons les principaux résultats et nous traçons quelques futures directions de recherche.

Un premier objectif de ce travail, était d'étudier comment les algorithmes d'apprentissage automatique ainsi que les principes d'extraction de connaissances à partir de bases de données pourraient être utilisés pour traiter ces problèmes. Nous avons également travaillé sur une base de données fournie par la Société Tunisienne d'Electricité et du Gaz (STEG), accessible via la plateforme ZINDI.

Afin de choisir des modèles adéquats pour ces problèmes, nous avons présenté dans le chapitre 2 un panorama de techniques d'apprentissage, et nous avons étudié dans le chapitre 3, en détail, la famille d'algorithme du gradient boosting, ce qui nous a permis de motiver notre choix de modèles.

Une seconde contribution était de créer des modèles d'apprentissage supervisé basés sur les arbres de décision, LightGBM et XGBoost qui ont été implémentés (en utilisant un ensemble d'outils décrits en Annexe 1) et testés.

Ces deux modèles sont réputés par leur force et capacité d'apprentissage, cependant nous avons justifié le choix du LightGbm en termes de performances. Effectivement, nous avons obtenu des résultats prometteurs pour la classification de ce type de problèmes.

Un autre objectif qui visait à démontrer l'efficacité de notre approche a été vérifié. Cette dernière qui incluait la classification d'un ensemble de données contenant environ un million d'abonnés, a permis d'obtenir des taux de classification élevés, atteignant 95,00% dans la phase d'apprentissage et un score de 88,71% dans la phase de test. Nous avons également obtenu une position encourageante, classées 19ème sur 295 participants dans le challenge lancé par Zindi.

Ces résultats encourageants nous incitent à améliorer davantage notre approche et à participer à d'autres défis similaires. De plus, nous envisageons d'appliquer cette solution dans nos propres organisations en Algérie, sachant que nous avons déjà fait cette initiative au sein de l'entreprise Sonelgaz où nous avons effectué un stage, à partir duquel nous avons cerné sa stratégie globale vis-à-vis ces problèmes de fraudes.

En conclusion, notre travail démontre le potentiel des techniques d'extraction de connaissances et d'intelligence artificielle dans la détection de fraude dans la consommation d'électricité et de gaz. Ces approches offrent des résultats précis et prometteurs, ouvrant ainsi la voie à des applications pratiques dans ce domaine.

Nous sommes motivés à continuer notre progression et à exploiter cette solution dans d'autres contextes et défis, avec l'objectif d'améliorer la prévention et la gestion des fraudes énergétiques. Nous pouvons discuter alors certains problèmes qui s'ouvrent à des travaux futurs dans la détection de fraude, qui, selon nous, méritent d'être étudiés, tels que ;

- i) Définir d'autres mesures de performance,
- ii) Modéliser le principe d'Alerte-Feedback,
- iii) Combiner entre les informations supervisées et non supervisées,
- iv) Proposer des algorithmes de clustering qui permettent d'attribuer des profils journaliers particuliers aux consommations.
- v) Intégrer des méthodes de collecte de données en temps réel en utilisant des compteurs intelligents.
- vi) Analyse de la corrélation entre la consommation d'électricité & gaz avec l'eau, afin d'extraire d'autres informations relatives à ces deux grandeurs
- vii) Il serait même intéressant de modéliser cette consommation d'énergie avec un système multi-agent.
- viii) On peut envisager de créer un ensemble de modèles en utilisant les prédictions de plusieurs modèles. Cela peut être réalisé en agrégeant les prédictions individuelles par moyenne, vote majoritaire ou stacking. Les ensembles de modèles ont souvent démontré de meilleures performances prédictives que les modèles individuels, ce qui pourrait augmenter la précision de votre système de détection de fraude.
- ix) Suivi et mise à jour régulières : Gardez à l'esprit que la détection de fraude est un défi en constante évolution, car les fraudeurs développent de nouvelles techniques pour éviter la détection. Continuez à surveiller et à collecter des données sur les fraudes réelles afin de mettre à jour régulièrement votre modèle et d'adapter votre système aux nouveaux schémas de fraude émergents.



ANNEXES

OUTILS DE
DEVELOPPEMENT

I

Sommaire

I INTRODUCTION.....	69
II OUTILS ET LANGAGE DE DEVELOPPENT.....	69
III CONCLUSION	71

I. INTRODUCTION

Dans le cadre du développement de notre système, nous avons utilisé un ensemble d'outils de développement et de bibliothèques logicielles pour créer une solution robuste et efficace. Cette annexe vise à décrire en détail les outils et les bibliothèques que nous avons utilisées.

II. OUTILS ET LANGAGE DE DEVELOPPEMENT

▪ Jupyter Notebook

C'est une application web open-source permettant de créer et de partager des documents contenant du code, des équations, des visualisations, et du texte narratif. Anciennement appelé IPython Notebooks, il s'agit d'un environnement de calcul interactif basé sur le web permettant de créer des documents notebooks.



Figure 1 : Logo Jupyter

▪ Python

Python est le langage de programmation open source créé par le programmeur Guido van Rossum en 1991. Il tire son nom de l'émission Monty Python's Flying Circus.

Il s'agit d'un langage de programmation interprété qui ne nécessite donc pas d'être compilé pour fonctionner. « Un programmeur interpréteur » permet d'exécuter le code Python sur n'importe quel ordinateur. Ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, ceci rend ce langage plus lent qu'un langage compilé comme le C.

En tant que langage de programmation de haut niveau, Python permet aux programmeurs de se focaliser sur ce qu'ils font plutôt que sur la façon dont ils le font. Ainsi, écrire des programmes prend moins de temps que dans un autre langage. Il s'agit d'un langage idéal pour les débutants.

Il est le langage de programmation le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science.



Figure 2 : Logo Python

▪ Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.



Figure 3 : Logo Google Colab

Nous avons utilisé les packages :

- **Numpy** : est une bibliothèque pour le langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.



Figure 4 : Logo NumPy.

- **Pandas** : la bibliothèque logicielle open-source Pandas est spécifiquement conçue pour la manipulation et l'analyse de données en langage Python. Elle est à la fois performante, flexible et simple d'utilisation.

Grâce à Pandas, le langage Python permet enfin de charger, d'aligner, de manipuler ou encore de fusionner des données.



Figure 5 : Logo Pandas.

- **Scikit-learn** : encore appelé sklearn, est la bibliothèque la plus puissante et la plus robuste pour le machine learning en Python. Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering via une interface cohérente en Python.



Figure 6 : Logo Scikit-learn

- **Matplotlib** : est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous forme de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy.



Figure 7 : Logo Matplotlib.

- **Seaborn** : est une bibliothèque pour créer des graphiques statistiques en Python. Il s'appuie sur Matplotlib et s'intègre étroitement aux structures de données pandas.



Figure 8 : Logo Seaborn.

III. CONCLUSION

Ces choix d'outils et de langages ont joué un rôle clé dans la réussite de notre projet en me permettant d'effectuer des analyses avancées et de présenter mes résultats de manière claire et convaincante.

ANNEXE

RAPPORT DE STAGE

II



Université –Ain Temouchent- Belhadj
Bouchaib
Faculté des Sciences et de Technologie
Département des Mathématiques et
Informatique



RAPPORT DE STAGE

Réalisé par :

- MESSABIHI Meriem
- OUNANE Amina

Tuteur académique

- Mme BOUHALOUAN Djamila

Superviseur de stage :

- Mr BENFODDA Mohamed

Entreprise d'accueil : Société algérienne de l'électricité et du gaz
(SONELGAZ) –Distribution d'Ain Temouchent

Sommaire

I INTRODUCTION	72
II MIEUX COMPRENDRE L'ORGANISATION DE SONELGAZ.....	72
II.1 Historique.....	72
II.2 Définition de la société	73
II.3 Les missions	74
II.4 Description de la structure	75
II.5 Fonctionnement spécifique de notre environnement de travail.....	76
III CONSOLIDATION DE COMPETENCES.....	76
III.1 Fiche de poste	76
III.2 Résultat du travail effectué	79
III CONCLUSION	79

I. INTRODUCTION

Comment nous sommes parvenues à frayer un chemin au sein de cette entreprise ? C'est ce que ce rapport de stage a pour but de vous faire découvrir.

Du [06/02/23] au [06/04/23], nous avons eu la chance de rejoindre la société Sonelgaz, la plus grande entreprise d'énergie en Algérie, en tant qu'étudiantes en Master 2 informatique de la spécialité « Réseaux et Ingénierie de Données » de l'université d'Ain Temouchent – BELHADJ Bouchaib –, naturellement attirées par le développement informatique, notre stage a été effectué dans le service d'informatique de l'entreprise.

Nous avons été sous la gouvernance de Mr BENFODDA Mohamed ingénieur en informatique, avec qui nous avons pu nous former dans d'excellentes conditions.

Cette expérience a été l'opportunité pour nous de percevoir comment une entreprise spécialisée en énergies se développe et de relever avec elle le défi à laquelle elle était confrontée et notamment face à la question de **comment détecter les fraudes dans la consommation d'électricité et de gaz à partir de l'historique de facturation des clients ?**

L'élaboration de ce rapport a pour principale source la pratique journalière des missions qui nous étaient affectées, mise en parallèle avec les enseignements théoriques de notre formation.

Afin de rendre compte de manière fidèle et analytique le temps passé au sein de Sonelgaz, il semble approprié et logique de se pencher au préalable sur l'organisation de Sonelgaz (historique, définition, missions et structure de la société), par la suite, nous présentons le fonctionnement spécifique de l'environnement de travail auquel nous nous sommes intéressées, puis il sera question de la nature des missions qui nous ont été confiées et les compétences consolidées avec une présentation d'une fiche de poste. Nous présentons après également les résultats du travail effectué et enfin nous concluons par une synthèse des savoirs que nous avons pu acquérir.

II. MIEUX COMPRENDRE L'ORGANISATION DE SONELGAZ

II.1 Définition de la société

La SONALGAZ, abréviation de la Société Nationale d'Électricité et de Gaz, sise à l'adresse « 150 logements Baraka Ain Temouchent » a été créée le 28 juillet 1969 pour remplacer l'ancienne société Électricité et Gaz d'Algérie (EGA), qui était le résultat des lois de nationalisation françaises de 1947. Cette mesure visait à consolider et à centraliser la gestion de l'électricité et du gaz en Algérie.



Figure 1 – Société algérienne de l'électricité et du gaz (SONELGAZ) – Distribution d'Ain Temouchent

II.2 Historique

La première entité Électricité et Gaz d'Algérie (EGA), qui était responsable de la production, de la distribution, de l'importation et de l'exportation d'électricité et de gaz a été remplacée par Sonelgaz en 1969.

En 1977, un plan national d'électrification et de développement de l'espace rural a été mis en place.

Dès lors, six entreprises autonomes de travaux ont été créées, en 1983 :

- KAHRIF pour l'électrification ;
- KAHRAKIB pour les infrastructures et installations électriques ;
- KANAGAZ pour la réalisation des réseaux de gaz ;
- INERGA pour le génie civil ;
- Montage industriel et l'entreprise AMC pour le montage industriel ;
- Fabrication des compteurs et appareils de mesure et de contrôle.

Vers les années 90, il y a eu un changement de la nature juridique, SONELGAZ est devenu un établissement public à caractère industriel et commercial (EPIC). Et en 1998, neuf filiales périphériques ont été créées.

En 2002, l'EPIC a été transformé en une holding de sociétés par actions.

Trois sociétés "métiers" ont été créées en 2004 : SPE pour la production d'électricité, GRTE pour le transport de l'électricité et GRTG pour le transport du gaz.

En 2005, la Société Civile de Médecine du Travail (SMT) et une société de recherche et développement de l'électricité et du gaz (CREDEG) ont été créées.

En 2006, quatre sociétés de distribution de l'électricité et du gaz (SDA, SDC, SDE et SDO) ainsi qu'une société de gestion du système électrique national (OS) ont émergé.

II.3 Les missions

La SONALGAZ a pour mission, à travers ses centrales électriques, ses réseaux et ses postes de transformation, de mettre à la disposition de ses abonnés les énergies électriques et gazières, qui sont aujourd'hui indispensables au développement de l'économie, au progrès technique et au bien-être des citoyens algériens.

De part, sa compétence nationale, la société nationale de l'électricité et du gaz détient le monopole de la production, du transport et de distribution de l'énergie d'électricité et du gaz naturel. D'autre part, sa mission de service public elle assure :

- La distribution d'une énergie de bonne qualité avec une présentation de la sécurité des personnes et des biens.
- Le dépannage rapide.
- La prise en charge des réclamations et la dispense de conseils utiles à ses abonnés.
- Le maintien de ses réseaux en bon fonctionnement

SONELGAZ a pour but l'alimentation et la distribution dans les zones suivantes :

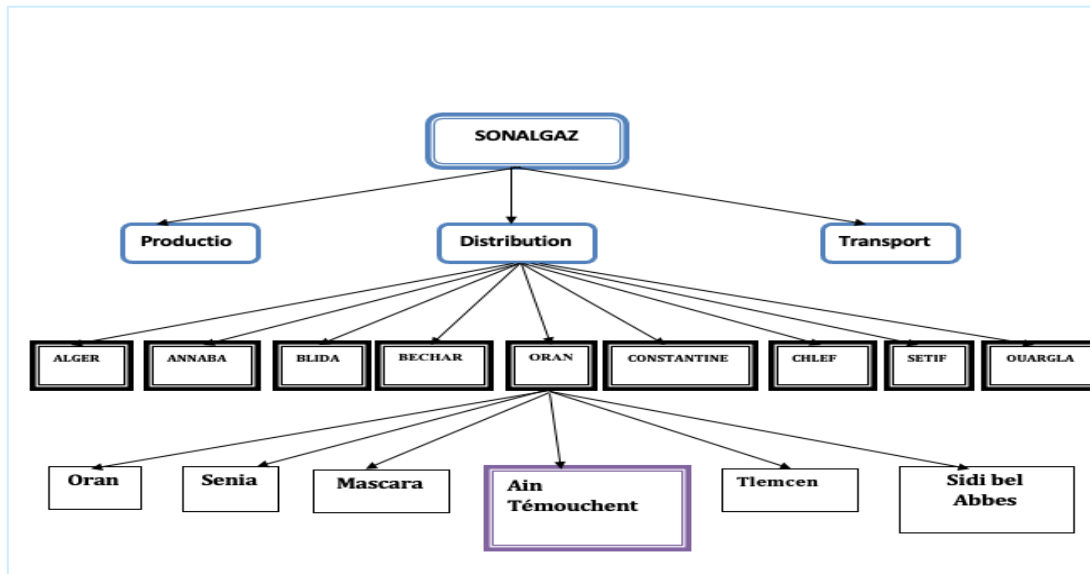


Figure 2 – Les zones de distribution

II.4 Structure de Sonelgaz

L'organigramme présenté dans la figure suivante illustre la structure de la société :

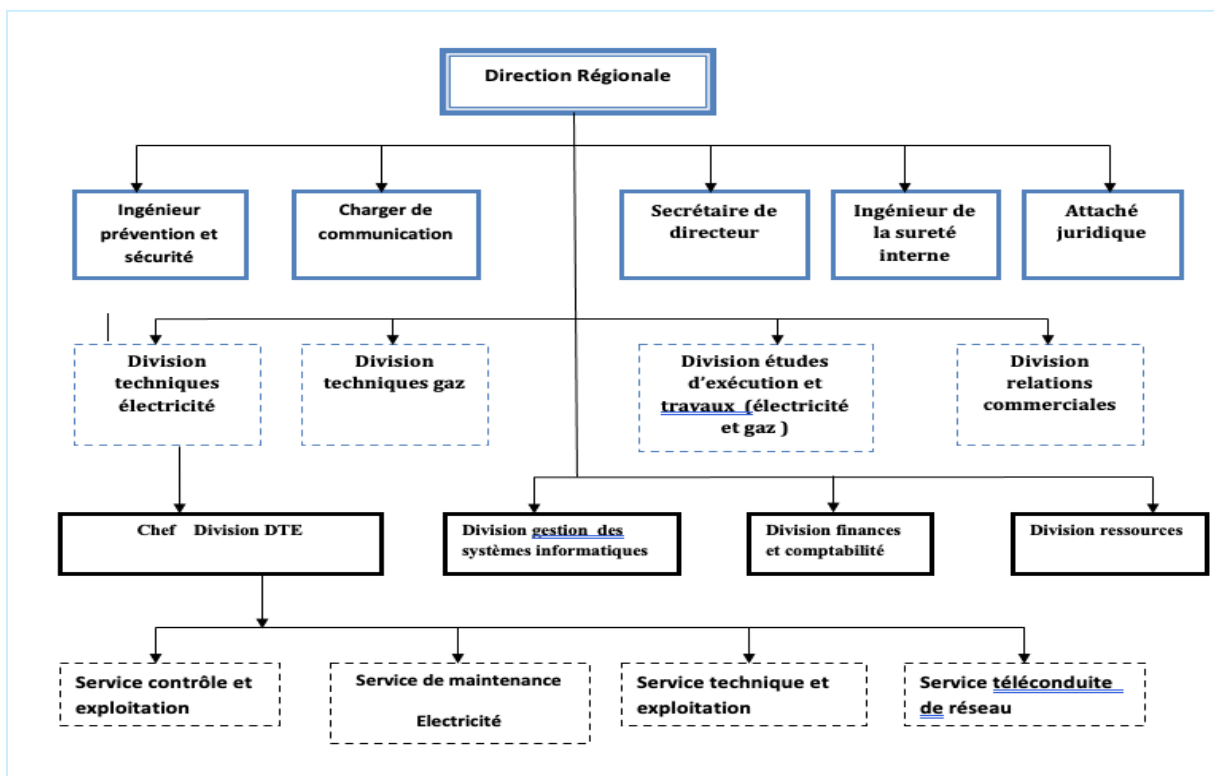


Figure 2 – Organigramme de l'unité d'AIN TEMOUCHENT

II.5 Fonctionnement spécifique de notre environnement de travail

Au sein de cette société nous avons eu la chance de percevoir une véritable synergie entre les services décrits au sein de l'organigramme.

Compte tenu du fait que notre stage a été exclusivement réalisé au **Division de gestion des systèmes informatiques**, la description suivante détaille uniquement cet environnement :

La Division de gestion des systèmes informatiques de Sonelgaz joue un rôle essentiel dans la gestion et l'optimisation des systèmes informatiques de l'entreprise. Cette division est responsable de la maintenance, du développement et de la sécurité des infrastructures informatiques de Sonelgaz. Elle veille à ce que les systèmes soient opérationnels, fiables et sécurisés, garantissant ainsi la continuité des activités de l'entreprise. Grâce à une équipe d'experts en informatique qualifiés, la Division de gestion des systèmes informatiques met en œuvre des solutions innovantes pour améliorer l'efficacité, la productivité et la performance des systèmes informatiques de Sonelgaz. Elle joue également un rôle clé dans l'adoption des technologies émergentes et la transformation numérique de l'entreprise, contribuant ainsi à son évolution et à son succès dans le secteur de l'énergie.

III. CONSOLIDATION DES COMPETENCES

III.1 Fiche de poste

Au cours de ce stage, nous avons eu l'opportunité de découvrir un métier sous toutes ses formes, ce qu'il apporte à Sonelgaz, la satisfaction personnelle que nous avons pu en tirer et les difficultés auxquelles nous avons pu faire face. Pour comprendre exactement de quoi il en retourne, nous vous proposons une description détaillée de notre fiche de poste.

Titre du poste : Stagiaires en Informatique

Informations personnelles

- **Stagiaire 1**

- Nom : MESSABIHI
- Prénom : Meriem
- Adresse : 14 Rue Mahroug Abed, Ain Temouchent, Algérie
- Numéro de téléphone : +213 657734922
- Adresse e-mail : Messabihimeriem9@gmail.com

- **Stagiaire 2**

- Nom : OUNANE
- Prénom : Amina
- Adresse : Rue 1er Novembre 1954, Ain Temouchent, Algérie
- Numéro de téléphone : +213 778149701
- Adresse e-mail : ounaneamina@gmail.com

Période de stage

- Date de début : 06 février 2023
- Date de fin : 06 Avril 2023
- Durée totale du stage : 2 mois

Objectifs du stage

- Acquérir une expérience pratique dans le domaine de l'informatique en milieu professionnel.
- Appliquer les connaissances théoriques acquises lors de notre formation universitaire.
- Contribuer aux projets informatiques de Sonelgaz et en apprendre davantage sur les technologies utilisées dans l'industrie.

Responsabilités et tâches

- Assister l'équipe informatique dans la maintenance et le support des systèmes informatiques.
- Participer à la résolution des problèmes techniques et à la configuration des logiciels.
- Contribuer à la mise en place de tests et à la documentation des procédures techniques.
- Collaborer avec l'équipe pour le développement et l'amélioration des applications internes.
- Effectuer des recherches et des analyses pour proposer des solutions informatiques efficaces.

Compétences requises

- Connaissance des langages de programmation tels que Java, Python et SQL.
- Compréhension des concepts de base des réseaux informatiques.
- Capacité à résoudre les problèmes techniques et à effectuer des dépannages.
- Compétences en communication écrite et verbale.
- Capacité à travailler en équipe et à s'adapter à un environnement professionnel.

Formation et qualifications requises

- Étudiantes en Master Réseaux et ingénierie de données à l'Université d'Ain Temouchent.
- Connaissances pratiques en programmation, bases de données et réseaux.
- Expérience préalable dans des projets académiques pertinents.

Superviseur

- Nom du superviseur : Mr. BENFODDA Mohamed
- Poste du superviseur : Ingénieur en informatique

Prise en charge

- Nous avons été prises en charge par Sonelgaz.

Lieu du stage

Adresse complète du lieu du stage : Sonelgaz, 150 logements Baraka Ain Temouchent, Algérie

Signature

III.2 Résultats du travail effectué

Au cours de notre stage de deux mois à la Société Nationale d'Électricité et de Gaz (SONALGAZ), nous avons identifié un besoin urgent d'un système de détection de fraude dans la consommation d'électricité et de gaz. La société est confrontée à d'importantes pertes financières résultant de cas de fraude tels que la manipulation des compteurs et les connexions illégales. Les systèmes de surveillance actuels se révèlent limités dans leur capacité à détecter de manière proactive ces activités frauduleuses. Ainsi, il est impératif de mettre en place un système plus avancé qui intègre des fonctionnalités d'analyse en temps réel, de surveillance des manipulations et de génération d'alertes, afin de renforcer les mesures de prévention et de détection de la fraude. Un tel système permettrait à SONALGAZ de réduire ses pertes financières, d'améliorer son efficacité opérationnelle et d'assurer une fourniture d'énergie plus fiable pour les consommateurs.

IV. CONCLUSION

Pour conclure, nous confirmons que notre stage effectué à Sonelgaz a été une expérience très satisfaisante. Nous avons travaillé dans un environnement professionnel stimulant et enrichissant, et avons acquis de nouvelles compétences en développement de logiciels. Nous avons également eu la chance de travailler aux côtés d'un ingénieur en informatique expérimenté qui nous a soutenus tout au long de notre stage. Enfin, nous avons constaté que la société aurait besoin d'un système de machine learning pour la détection de fraude dans la consommation d'électricité et de gaz, et nous sommes convaincus que ce serait une solution efficace pour améliorer les performances de l'entreprise dans ce domaine.



BIBLIOGRAPHIE

[AGA & All, 05]

B. AGARD, A. KUSIAK. « Exploration Des Bases De Données Industrielles à L'aide Du Data Mining –Perspectives », 9ème Colloque National AIP PRIMECA, avril 2005.

[ALL, 22]

Amandine Allmang. « Principaux algorithmes d'apprentissage non supervisé » Linedata. (n.d.-b) [.https://fr.linedata.com/principaux-algorithmes-dapprentissage-non-supervise/](https://fr.linedata.com/principaux-algorithmes-dapprentissage-non-supervise/). 2009.

[ARO & All, 16]

Siddharth Arora and James W. Taylor. « Forecasting electricity smart meter data using conditional kernel density estimation Omega» livre (47–59). 2016.

[AQU, 22]

Aquila Data Enabler , «Semi-supervised Learning & Active Learning : comment tirer profit des données non-labellisées ? » .AQUILA DATA ENABLER. <https://www.aquiladata.fr/insights/semi-supervised-learning-active-learning-comment-tirer-profit-des-donnees-non-labellisees/>. (2022, July 19).

[AWS, 22]

Amazon Web Services «Qu'est-ce que le boosting ? – Le boosting dans le cadre du machine learning expliqué – AWS». <https://aws.amazon.com/fr/what-is/boosting/>. (26 mai 2022).

[AYB & All, 20]

Ayub, N., Irfan, M., Awais, M., Ali, U., Ali, T. Hamdi, M., Alghamdi, A., 19Muhammad, F. «Big data analytics for short and medium-term electricity load forecasting using an AI techniques ensemble » . Energies, 1–21. <https://doi.org/10.3390/en1319519>, (2020).

[BEL, 11]

BELGACEM, Brahim. «Extraction de connaissances à partir de données incomplètes et imprécises ». Thèse de doctorat, Université Mohamed Boudiaf de Msila. 2011.

[BEL, 22a]

Belaidi Nada. (n.d.-a). «Arbres de décision en Machine Learning : tout comprendre». Formation Tech et Data en ligne Blent.ai. <https://blent.ai/blog/a/arbres-de-decision-en-machine-learning>.(20 janvier 2022).

[BEN, 13]

R. BENKHELIFA. «Fouille de données d'opinion des usagers de sites E-commerce». Mémoire Master, University Ouargla. Juin 2013.

[BEN, 18]

Seif-Eddine Benkabou. « Détection d'anomalies dans les séries temporelles: application aux masses de données sur les pneumatiques». PhD thesis. Université Claude Bernard, 2018.

[BEN, 19]

BENSIAH Oussama Akram, « La proposition d'une nouvelle approche basée Deep Learning pour la prédiction du cancer du sein ». Mémoire de fin d'étude en vue de l'obtention du diplôme de Master en informatique .2019/2020.

[BER & All, 00]

M. J. BERRY. «Mastering Data Mining: The Art and Science of Customer Relationship Management» Science Books @ Amazon.com. (n.d.). » <https://www.amazon.com/Mastering-Data-Mining-Relationship-Management/dp/0471331236>. 2000.

[BER & All, 04]

M. J. BERRY, G. S. LINOFF, «Data Mining Techniques For Marketing, Sales, and Customer Relationship.Management.Second Edition», livres. 2004.

[BER, 21]

BERKANE MABROUKA, « Bank fraud detection using sequential pattern mining ». Mémoire présenté pour obtenir le diplôme de master académique en Informatique.2021.

[BIS & All, 17]

Félix Biscarri, Iñigo Monedero, Antonio García, Juan Ignacio Guerrero, and Carlos León. « Electricity clustering framework for automatic classification of customer loads». *Expert Systems with Applications*, 86:54–63, 2017.

[BOU& All, 19]

BOULTACHE Thanina et LAMZAOUI Amar, , « Reconnaissance d'activités humaines à l'aide de capteurs de smartphone », Mémoire de fin d'étude en vue de l'obtention du diplôme de Master en informatique Réseaux, Mobilité et Systèmes Embarqués. 2019/2020.

[Cay, 22]

Cayla, B. « La star des algorithmes de ML : XGBoost. Datacorner Par Benoit Cayla », <https://datacorner.fr/xgboost/>.2022.

[CDA, 22]

Crochet-Damais, A. «Régression linéaire multiple : définition, principes et cas d'usage ». www.journaldunet.fr-<https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501893-regression-lineaire-multiple-definition-principes-et-cas-d-usage/>. (2022).

[Ceo, 19]

Glossaire - Intelligence artificielle - www.coe.int. (n.d.). « Intelligence Artificielle». (6 janvier 2019). <https://www.coe.int/fr/web/artificial-intelligence/glossary>. (6 janvier 2019).

[CHE & All, 14]

J Chen and D Boccelli. «Demand forecasting for water distribution systems». Environmental Engineering Program, University of Cincinnati, Cincinnati, OH (USA).2014.

[Che, 23]

Chen, J.«What Is a Neural Network? »
.Investopedia.<https://www.investopedia.com/terms/n/neuralnetwork.asp>. (2023).

[CHI & All, 17]

Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. « A review on time series forecasting techniques for building energy consumption ». Renewable and Sustainable Energy Reviews, 74:902–924. 2017.

[CHO, 22]

Chow, R. « Gradient Boosting Algorithms: Busting Bias Error». History of Data Science.
*<https://www.historyofdatascience.com/gradient-boosting-algorithms->.(2022).

[CHU & All, 22]

Selina. Chu, Eamonn. Keogh, David. Hart, and Michael. Pazzani. «Iterative deepening dynamic time warping for time series » .In Proceedings of the 2002 SIAM International Conference on Data Mining, Proceedings, pages 195–212. Society for Industrial and Applied Mathematics. 2002.

[Dab, 22]

Dabbura, I « K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks». *Medium*. <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>. . (2022, September 27).

[DID, 16]

Didrik Nielsen. «Tree Boosting With XGBoost».Disponible sur :
<http://pzs.dstu.dp.ua/DataMining/boosting/bibl/Didrik.pdf>. (2016).

[DJA, 21]

Mohamed Abd Elmoumen DJABALLAH, « Système de prédiction de la consommation d'énergie basé Deep Learning » . Mémoire de fin d'études Master Systèmes et Technologie de l'Information et de la Communication. Septembre 2021.

[DJE, 14]

Dr. Abdelhamid DJEFFAL «Fouille de données avancée » . Cours. Disponible sur :
http://www.abdelhamid-djeffal.net/web_documents/polycopefda.pdf. 2014/2015.

[DJE, 22]

DJEFFAL AMANI, « La détection d'anomalie pour la lutte contre la fraude documentaire », Mémoire présenté pour obtenir le diplôme de master académique en Informatique, parcours : Intelligence Artificielle (IA), 2022.

[DUM & All, 98]

William DuMouchel and Matthias Schonlau. «A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities». in Proc. KDD, page 5, 1998.

[FAW & All, 97]

Tom FAWCETT and Foster PROVOST. « Adaptive fraud detection». Data Mining and Knowledge Discovery, 1:291–316, 1997.

[FAY & All, 96a]

Smyth.P Fayyad U. Piatesky-shapiro G. «Knowledge discovery and data mining : Towards a unifying framework». In knowledge discovery and data mining. Pages 16-34, 1996.

[FAY & All, 96b]

Smyth.P Fayyad U., Piatesky-shapiro G. «Knowledge discovery and data mining : Towards a unifying framework». In knowledge discovery and data mining. Pages 82-88, 1996.

[FAY & All, 98]

Smyth.P Fayyad U. Piatesky-shapiro G. « From data mining to knowledge discovery in databases advices in knowledge discovery and data mining». MIT Press, pages 1-36, 1998.

[FER & All, 12]

Ferreira, A. & Figueiredo, «Boosting Algorithms: A Review of Methods, Theory, and Applications». In *Springer eBooks* (pp. 35–85). https://doi.org/10.1007/978-1-4419-9326-7_2. M. a. T. (2012).

[GEE, 23]

GeeksforGeeks. «LightGBM Light Gradient Boosting Machine». GeeksforGeeks. <https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/>.(2023).

[GAU & All, 19]

Megha Gaur, Stephen Makonin, Ivan V. Bajic, and Angshul Majumdar. « Performance evaluation of techniques for identifying abnormal energy consumption in buildings ». IEEE Access, 7:62721–62733, 2019.

[GUE & KAZ, 19]

GUEMIDI Abdelwahab et KAZITANI Noredine, « Détection et classification des émotions des personnes » Mémoire de projet de fin d'étude pour l'obtention du diplôme de Master Académique en Génie Electrique, 2019/2020.

[GUI, 19]

Guillaume Saint-Cirgue, « Apprendre le Machine Learning en une semaine». Machine Learnia. <https://machinelearnia.com/apprendre-le-machine-learning-en-une-semaine/> .2019.

[GUS & All, 19]

Gustavo de Souza Groppo, Marcelo Azevedo Costa, and Marcelo Libânio. « Predicting water demand: a review of the methods employed and future possibilities». Water Science & Technology: Water Supply, 19(8):2179–2198, 2019.

[Had, 02]

Med Haddad. «Extraction et impact des connaissances sur les performances des systèmes de recherche d'information», pour obtenir le grade de DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER Discipline : Informatique 2002.

[HAS & All, 21]

HASSANI Nihal et ZABAT Asma, « Une approche d'optimisation pour une meilleure efficacité d'un modèle d'estimation de temps restant utile du moteur à double flux à base de deep learning », Mémoire préparé en vue de l'obtention du diplôme de Master en informatique Sciences et Technologies de l'information et de la Communication. 2021/2022.

[HEL, 23]

Helm, C. « Was ist Random Forest? – Kinderleicht und unvergesslich erklärt. *Konfuzio*. » <https://konfuzio.com/fr/random-forest-tale/>. Blog sue AI. (2023).

[HOU, 07]

HOUMADI, Benamar, «Étude exploratoire d'outils pour le Data Mining», Thèse de doctorat, Université du Québec à Trois-Rivières, 2007.

[JOB, 21]

Joby, A. «What Is K-Nearest Neighbor? An ML Algorithm to Classify Data. » <https://learn.g2.com/k-nearest-neighbor>. (19 juillet 2021).

[Jol, 03]

François-Xavier Jollois. « Contribution de la classification automatique à la Fouille de Données. » PhD thesis, Université de Metz, 2003.

[KAN, 03]

Kantardzic M. « Data mining – concepts, models, methods, and algorithms. » IEEE Press, Piscataway, NJ, USA, 2003.

[KAP & All, 10]

R. Puust, Z. Kapelan, D. A. Savic, and T. Koppel. «A review of methods for leakage management in pipe networks. ». *Urban Water Journal*, 7(1):25–45, 2010.

[KAZ & All, 17]

Kazeem B Adedeji, Yskandar Hamam, Bolanle T Abe, and Adnan M Abu-Mahfouz. «Leakage detection and estimation algorithm for loss reduction in water piping networks. » *Water*, 9(10):773, 2017.

[KER, 20]

KERDOUD Fateh, « Un système de compression d'images basé sur les réseaux de neurones profonds ».Mémoire de Master Systèmes et Multimédias, 2020/2021.

[Kha, 05]

Jamal Kharroubi. « Etude de techniques de classement "Machines à vecteurs supports" pour la vérification automatique du locuteur. » <https://pastel.archives-ouvertes.fr/pastel-00001124/document>. (11 mars 2005).

[KIM, 20]

Kimouche Achouak. « Modèles de Markov Cachés : Application en Biologie. » Mémoire (2020/2021).

[KUM, 22]

Kumar, S. « C-Means Clustering Explained. » Built In. <https://builtin.com/data-science/c-means>. (2022b).

[KUR, 21]

Kurama Vihar. « Gradient Boosting for Classification | Paperspace Blog. » Paperspace Blog. <https://blog.paperspace.com/gradient-boosting-for-classification>. (2021).

[LAR, 05]

D.T. LAROSE, « Discovering Knowledge In Data: An Introduction to Data Mining ». Central Connecticut State University. 2005.

[LAS & All, 15]

Chrysi Laspidou, Elpiniki Papageorgiou, Konstantinos Kokkinos, Sambit Sahu, Arpit Gupta, and Leandros Tassioulas. « Exploring patterns in water consumption by clustering. » *Procedia Engineering*, 119:1439–1446, 01 2015.

[LAT, 23]

Zulaikha Lateef. « A Comprehensive Guide To Boosting Machine Learning Algorithms. *Edureka*. <https://www.edureka.co/blog/boosting-machine-learning/>. (2023).

[LIA, 05]

T. Warren Liao. « Clustering of time series data—a survey ». *Pattern Recognition*, 38(11):1857–1874, 2005.

[Lie, 07]

J. Lieber. « fouille de données : note de cours. » . fortement mais librement inspire du cours d'amedeo napoli 2007.

[LOH, 11]

Séraphin LOHAMBA OMATOKO .«Analyse et détection de l'attrition dans une entreprise de Télécommunication. » Mémoire de fin d'études en vue de l'obtention d'un diplôme de Master Académique en Informatique.2011.

[Loï, 22]

Loïc Bremme, «Définition : Qu'est-ce que le Big Data? » Article, Disponible sur : <https://www.lebigdata.fr/definition-big-data>, le 1 décembre 2022.

[LUI & All, 18]

Chiara Luciani, Francesco Casellato, Stefano Alvisi, and Marco Franchini . « From water consumption smart metering to leakage characterization at district and user level The GST4 water project ». 2(11):675, 2018.

[MEN, 09]

MENOUER Tarek, DERMOUCHE Mohamed, Application de techniques de data mining pour la classification automatique des données et la recherche d'associations, Mémoire de fin d'études pour l'obtention du diplôme d'Ingénieur d'Etat.

[MER & All, 17]

MERDOUD Kenza et BOUSBAIN Karim.« Détection de maladies par traitement d'image ». Mémoire de fin d'étude en vue de l'obtention du diplôme de Master Génie Système Informatique.2017/2018.

[MEZ, 20]

MEZILI Houcine, « Vers une amélioration de la détection d'intrusion par les méthodes de sélection des fonctionnalités à l'aide des arbres de décision ». Mémoire pour l'obtention du diplôme de Master Réseaux et Télécommunication.<http://dspace.univtiaret.dz/bitstream/123456789/5509/1/TH.M.INF.FR.2021.29.pdf>. .2020/2021.

[Micr, 22]

Microsoft Disponiblesur :<https://experiences.microsoft.fr/articles/intelligence-artificielle/comprendre-utiliser-intelligence-artificielle/>.9 févr. 2022.

[Min, 23]

MinnaLearn c2rses.« Les types d'apprentissage automatique » Disponible sur : <https://course.elementsofai.com/fr/4/1.9/01/2023>.

[Mobi, 21]

Mobiskill. « Apprentissage supervisé vs apprentissage non supervisé ». *Mobiskill*. (2021). <https://mobiskill.fr/blog/conseils-emploi-tech/apprentissage-supervise-vs-apprentissage-non-supervise/>.2021

[PIA, 91]

G. Piatetsky-Shapiro. « Discovery, Analysis, and Presentation of Strong Rules». Knowledge Discovery in Databases. AAAI/MIT Press, Cambridge, 248, 255-264. 1991.

[Pit, 21a]

Pitpitt. (n.d.). « Modèle de mélange gaussien — DataFranca ». [Datafranca.org](https://datafranca.org/wiki/Mod%C3%A8le_de_m%C3%A9lange_gaussien). https://datafranca.org/wiki/Mod%C3%A8le_de_m%C3%A9lange_gaussien.(1 février 2021).

[Pit, 21b]

Pitpitt. (n.d.). « Régression Lasso — DataFranca ». Datafranca.org. https://datafranca.org/wiki/Lasso_Regression. (9 novembre 2021).

[RED, 22]

Rédac, T« *La régression logistique, qu'est-ce que c'est?* Formation Data Science». | DataScientest.com. <https://datascientest.com/regression-logistique-quest-ce-que-cest>. (4 août 2022).

[RUI & All, 17]

Ruizhe Ma and Rafal Angryk. « Distance and density clustering for time series data ». In 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pages 25–32, 2017.

[SAI, 21]

SAIDJ Soumia Douâa, « Techniques de NLP pour la détection des fausses nouvelles ».Mémoire pour l'obtention du diplôme de Master en Génie Informatique. 2021/2022.

[SAL & All, 19]

SALMI Nadia et BAHRI Rym, . « Modèle de prédiction et de détection de Malwares informatiques ». Mémoire de Master en Informatique ISIL, GSL, 2019/2020.

[SAM, 19]

Senani SAMY. « Réseaux de neurones convolutionnels pour la détection précoce de la rétinopathie diabétique ». Mémoire de fin d'études en vue de l'obtention du diplôme de Master Système Informatique.10 juillet 2019.

[SAU, 21]

Saul Dobilas. «XGBoost: Extreme Gradient Boosting — How to Improve on Regular Gradient Boosting? » Disponible sur : <https://towardsdatascience.com/xgboost-extreme-gradient-boosting-how-to-improve-on-regular-gradient-boosting-5c6acf66c70a>. 2021.

[SCH & All, 01]

Matthew G. Schultz..Eleazar Eskin, F.Zadok, Salvatore J. Stolfo, Article, « Data Mining Methods for Detection of New Malicious Executables». University of New York at Stony Brook ezk@cs.sunysb.edu.2001.

[Sh , 23]

Shreyanshisingh28. «LightGBM Light Gradient Boosting Machine». GeeksforGeeks. <https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/>.(2023).

[SPA, 23]

Spanton, R. «Polynomial Regression: An Introduction ». *Built In*. <https://builtin.com/machine-learning/polynomial-regression>. (2023).

[Spi, 23]

Spiceworks « What is Linear Regression? ». <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/amp/>.(3 avril 2023).

[Swa & All, 93]

Swami A. Agrawal R., Imielinski T. « Proceedings of the ACM SIGMOD International Conference on Management of Data Washington ». Mining association rules between sets of items in large database. DC, 10 :207-201, May 1993.

[Team, 21b]

Team, D. S. «Validation croisée K-Fold. *DATA SCIENCE* ». <https://datascience.eu/fr/apprentissage-automatique/k-fold-cross-validation/>.(2021b).

[TIB & All, 01]

Robert Tibshirani, Guenther Walther, and Trevor Hastie. «Estimating the number of clusters in a data set via the gap statistic ». *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[TUF, 02]

S. Tufféry. «Bases de données et gestion de la relation client groupe bancaire francais».Data mining et scoring. 2002.

[VIC, 21]

Victor Landeau. «*XGBoost, LightGBM, CatBoost : lequel choisir ?*» (n.d.). <https://blog.ekinox.io/ml/gradient-boosting>. (24 mars 2021).

[VOX, 23]

Voxco. « Régression multivariée : définition, exemple et étapes | Voxco ». *Voxco*. (22 Mars 2023).

[WAN & All, 04]

Xiaozhe Wang, Kate A Smith, Rob Hyndman, and Daminda Alahakoon. « A scalable method for time series clustering ». *Proc. Unrefereed Res. Papers*, 2004.

[WEI & All, 14]

Weihui Deng, Guoyin Wang, Xuerui Zhang, Yishuai Guo, and Guangdi Li. «Water quality prediction based on a novel hybrid model of ARIMA and RBF neural network». In *IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, pages 33–40. IEEE. 2014.

[Wiki, 22]

Contributeurs aux projets Wikimedia. « Classification naïve bayésienne. ». *wikipedia.org*. https://fr.wikipedia.org/wiki/Classification_na%C3%AFve_bay%C3%A9sienne.(2022).

[Wiki, 23]

«Expectation–maximization algorithm. *Wikipedia* ». https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm. (2023).

[Wil & All, 18]

M.William et N.TOLOFON, «Etude et mise en place d'un système basé sur le machine learning pour la détection de fraudes monétiques ». mémoire pour l'obtention du master professionnel en réseaux et systèmes informatiques 2018/2019, Disponible sur :

https://www.doyoubuzz.com/var/f/4s/ky/4sky2Fg4ZTeEG8SVnumxWDabz1J7RQKLPXs3ohjl5vU_master.pdf

[Yon, 03]

Yonv. «Régression multivariée : définition, exemple et étapes ». <https://www.voxco.com/fr/blog/regression-multivariee-definition-exemple-et-etapes-2/>. 2003

[ZHA, 21]

Zhang, Z. « Boosting Algorithms Explained ». Towards Data Science. Medium. <https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30>. (2021, December 10).

[Zig & All, 00]

Zighed D. and Rakotomalala R. «Graphes d'induction : apprentissage automatique et data mining ». Hermes., pages 82-88, 2000.