École Nationale Supérieure
d'Informatique et d'Analyse des Systèmes
- Rabat

Audit Intelligent : Détection d'anomalies dans les journaux comptables via Machine Learning

Rapport de Stage de Fin d'Année

Du 15 Juin au 05 Septembre

Nom et prénom : Jad Falaq

Entreprise : AKWA Group

Adresse entreprise : Km 7, N1, Casablanca, Maroc

Maître de stage : Mr Kamal GHANAM

Encadrant académique : Mr Abdellatif EL AFIA

Période : Du 15 Juin au 05 Septembre

Jury : Mr Mohamed LAZAAR et Mr El Houssaine HSSAYNI

Année Universitaire 2024 – 2025

Remerciements

Je souhaite tout d'abord exprimer ma profonde gratitude à **Monsieur Kamal GHANAM**, mon maître de stage, pour son encadrement attentif, ses conseils avisés et sa disponibilité tout au long de mon stage. Son expertise, sa rigueur professionnelle et sa capacité à transmettre ses connaissances m'ont été d'une aide précieuse. Grâce à son accompagnement, j'ai pu comprendre les enjeux du projet et développer mes compétences techniques et organisationnelles.

Je remercie également mon chef de filière, **Monsieur Abdellatif EL AFIA**, pour m'avoir accordé l'opportunité de réaliser ce stage au sein d'AKWA Group. Son soutien constant et sa disponibilité ont été essentiels tout au long de cette expérience. Grâce à ses conseils avisés, j'ai pu mieux organiser mon travail et surmonter les défis rencontrés durant le projet. Sa confiance et ses encouragements m'ont permis de gagner en autonomie et en rigueur professionnelle, tout en approfondissant mes connaissances dans le domaine de l'intelligence artificielle appliquée à la détection d'anomalies financières. Je lui exprime ma sincère reconnaissance pour l'attention qu'il a portée à mon parcours et pour l'inspiration qu'il m'a apportée tout au long de ce stage.

Je tiens aussi à remercier mes **collègues stagiaires** avec lesquels j'ai eu le plaisir de collaborer durant cette période. Leur esprit d'entraide, leur disponibilité et la coopération dont nous avons fait preuve ont rendu cette expérience encore plus enrichissante et motivante. Les échanges que nous avons eus m'ont permis de progresser et de mieux appréhender le travail en équipe.

Enfin, je remercie toutes les personnes de l'entreprise qui, directement ou indirectement, ont contribué à la réussite de mon stage. Leur accueil chaleureux et leur soutien ont favorisé un environnement de travail agréable et propice à l'apprentissage.

Ce stage a été pour moi une expérience formatrice exceptionnelle, et je suis reconnaissant envers tous ceux qui ont participé à son bon déroulement.

Résumé

Ce travail de recherche se concentre sur la détection d'anomalies dans les transactions financières, plus spécifiquement pour identifier les transactions ¡présentes dans le fichier Afriware mais absentes des fichiers JDE. L'objectif principal est de réduire le risque d'erreurs ou de fraudes dans le processus d'audit en utilisant des modèles d'apprentissage supervisé.

Cinq modèles supervisés ont été utilisés : Random Forest [2], Support Vector Machines (SVM), [4], k-Nearest Neighbors (kNN) [3], Logistic Regression [9] et Decision Tree [6]. Chaque modèle a été évalué sur un ensemble de test séparé afin de mesurer sa performance à l'aide des métriques classiques telles que la précision, le rappel et le F1-score.

Les résultats montrent que Random Forest et SVM offrent les meilleures performance, avec un taux de faux positifs relativement faible. L'étude a également testé des modèles non supervisés tels que l'Isolation Forestet les autoencoders, mais ces derniers se sont révélés moins pertinents en raison de la grande quantité de données et de la spécificité des anomalies ciblées [8, 10].

Enfin, ce travail met en avant l'importance de l'automatisation des processus d'audit pour réduire le temps de traitement et améliorer la fiabilité de la détection d'anomalies. Les perspectives futures incluent l'optimisation des modèles non supervisés, l'utilisation de réseaux neuronaux pour détecter d'autres types d'anomalies et la détection de fraudes fiscales, notamment liées à la TVA [7].

Mots-clés : Détection d'anomalies, transactions financières, apprentissage supervisé, Logistic Regression, Decision Tree, Random Forest, SVM , KNN

Abstract

This research work focuses on anomaly detection in financial transactions, specifically to identify transactions present in the Afriware file but missing from the JDE files. The main objective is to reduce the risk of errors or fraud in the audit process using supervised learning models.

Five supervised models were used : Random Forest [2], Support Vector Machines (SVM), [4], k-Nearest Neighbors (kNN) [3], Logistic Regression [9] et Decision Tree [6]. Each model was evaluated on a separate test set to measure its performance using classical metrics such as accuracy, recall, and F1-score.

The results show that Random Forest and SVM offer the best performance, with a relatively low false positive rate. Unsupervised models such as Isolation Forest and autoencoders were also tested, but they proved less relevant due to the large amount of data and the specificity of the targeted anomalies [8, 10]..

Finally, this work highlights the importance of automating audit processes to reduce processing time and improve the reliability of anomaly detection. Future perspectives include optimizing unsupervised models, using neural networks to detect other types of anomalies, and detecting tax fraud, particularly related to VAT [7].

Keywords : anomaly detection, financial transactions, supervised learning, Logistic Regression, Decision Tree, Random Forest, SVM, kNN

Liste des Abréviations

ENSIAS	École Nationale Supérieure d'Informatique et d'Analyse des Systèmes
2IA	Ingénierie Intelligence Artificielle
TVA	Taxe sur la Valeur Ajoutée
Afriware	Nom du fichier de référence contenant les transactions validées
JDE	Système comptable (JD Edwards) utilisé pour l'audit
GL	Grand Livre
EDI	Échange de Données Informatisé (Electronic Data Interchange)
ERP	Enterprise Resource Planning (Progiciel de gestion intégré)
RP	Règlement de Paiement (ou Relevé de Paiement)
ML	Machine Learning (Apprentissage automatique)
RF	Random Forest
SVM	Support Vector Machines
kNN	k-Nearest Neighbors
LR	Logistic Regression
DT	Decision Tree (Arbre de décision)
ISA	International Standards on Auditing
CRISP-DM	Cross Industry Standard Process for Data Mining
SEMMA	Sample, Explore, Modify, Model, Assess (Méthodologie de Data Mining)
KDD	Knowledge Discovery in Databases (Découverte de connaissances dans les DB)
DB	DataBase
TDSP	Team Data Science Process (Processus d'équipe pour les projets Data Science)
PCA	Principal Component Analysis (Analyse en composantes principales)
TP	True Positive (Vrai positif)
TN	True Negative (Vrai négatif)
FP	False Positive (Faux positif)
FN	False Negative (Faux négatif)
MontantTTC	Montant Toutes Taxes Comprises
MontantHT	Montant Hors Taxes
Metric_avg	Moyenne des métriques de performance
F1-score	Mesure combinant précision et rappel

Introduction générale

0.1 Présentation de l'entreprise et de l'établissement

L'entreprise AKWA

AKWA Group est un acteur majeur du secteur industriel et énergétique au Maroc. Fondé avec pour objectif le développement durable et l'innovation, le groupe opère dans plusieurs domaines, notamment la production et la distribution de carburants, la chimie, la logistique et les services associés. AKWA s'engage à promouvoir l'excellence opérationnelle et la responsabilité sociale, en contribuant à la croissance économique et à la création d'emplois à l'échelle nationale.

Au sein d'AKWA, mon stage s'est déroulé dans la **Direction des Systèmes d'Information (DSI)**, qui joue un rôle stratégique dans la transformation numérique de l'entreprise. La DSI est chargée de la conception, de la maintenance et de la sécurisation des infrastructures informatiques, des systèmes de gestion et des applications métier. Elle vise à optimiser les processus internes et à faciliter la prise de décision à travers l'exploitation des données et le déploiement de solutions technologiques innovantes.

L'École ENSIAS

L'**École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS)** est un établissement public de renom au Maroc, spécialisé dans la formation d'ingénieurs de haut niveau dans le domaine des sciences et technologies de l'information. L'école propose plusieurs filières, dont la filière **2IA** (Ingénierie Intelligence Artificielle), qui vise à former des ingénieurs capables de concevoir, développer et déployer des systèmes intelligents basés sur l'intelligence artificielle, le machine learning et l'analyse de données.

La formation combine des cours théoriques approfondis, des travaux pratiques et des projets innovants, permettant aux étudiants d'acquérir des compétences techniques solides ainsi qu'une capacité d'analyse et de résolution de problèmes complexes. Cette approche prépare efficacement les diplômés à relever les défis technologiques et à contribuer à la transformation numérique des entreprises.

0.2 Pertinence et motivation du projet

Dans le cadre de mon projet de fin d'année de stage au sein de l'entreprise **AKWA**, l'application des techniques d'apprentissage automatique à l'analyse comptable et à l'audit interne revêt une importance particulière. En effet, avec l'augmentation continue du volume de données financières générées par les systèmes de gestion intégrés (ERP), le traitement manuel devient de plus en plus complexe, nécessitant des approches automatisées et intelligentes pour assurer fiabilité et efficacité.

Les opérations comptables de l'entreprise constituent une source essentielle d'informations sur son activité financière. L'intégration de l'apprentissage automatique vise à faciliter le travail d'analyse, à réduire les risques d'erreurs humaines et à détecter les anomalies significatives susceptibles d'affecter la fiabilité des états financiers.

La **détection des anomalies dans les journaux comptables** représente un enjeu central pour le service financier d'AKWA. Les écritures du grand livre contiennent des transactions représentant diverses opérations comptables, chacune associée à des comptes, montants et signes (débit/crédit). L'analyse isolée d'une écriture ne suffit pas à comprendre la nature de l'opération, tandis que la combinaison de plusieurs types de comptes permet de révéler des schémas d'activité financière. Les anomalies peuvent alors se manifester sous forme de combinaisons inhabituelles de comptes, de montants incohérents ou de signes erronés. Ces anomalies peuvent résulter d'erreurs humaines ou, dans certains cas, de manipulations intentionnelles.

Traditionnellement, la détection de telles anomalies repose sur l'expertise humaine et sur l'échantillonnage manuel, un processus long et limité qui ne garantit pas la couverture complète de toutes les transactions. L'objectif de ce projet est donc d'introduire une approche fondée sur **le machine learning** afin d'automatiser partiellement le processus d'audit interne et d'améliorer la capacité de détection des anomalies. Cette approche permettra de renforcer la fiabilité du contrôle comptable tout en réduisant le temps et le coût du traitement des données.

0.3 Définition du problème et objectifs du projet

L'un des principaux défis auxquels fait face le département financier d'AKWA réside dans la gestion d'un volume croissant de données comptables et transactionnelles. Face à cette masse d'informations, les auditeurs internes ne peuvent généralement examiner qu'un échantillon restreint des écritures, ce qui crée un risque de non-détection d'anomalies importantes dans les transactions non étudiées.

De plus, les schémas d'erreurs ou de fraude évoluent constamment, rendant difficile leur identification par des méthodes traditionnelles. Cette limitation nuit à la fiabilité du processus d'audit interne et peut engendrer des risques financiers et réglementaires.

L'objectif principal de ce projet est de **développer un système intelligent de détection d'anomalies dans les journaux comptables d'AKWA** en exploitant les techniques d'apprentissage automatique. Plus précisément, le projet vise à :

- Identifier et analyser les types d'anomalies potentielles présentes dans les écritures comptables ;
- Mettre en place un modèle de machine learning capable de détecter automatiquement les transactions suspectes ;
- Fournir aux auditeurs un outil d'aide à la décision leur permettant de cibler les zones à risque avec une plus grande précision ;
- Réduire le temps et le coût liés aux audits internes tout en augmentant la fiabilité des contrôles.

Ainsi, ce projet s'inscrit dans la démarche d'innovation et de digitalisation des processus internes d'AKWA, contribuant à renforcer la transparence et la qualité de l'information financière au sein de l'entreprise.

Dans ce cadre, nos données proviennent de trois documents extraits des systèmes Afriware et JDE. Afriware nous fournit un grand livre, tandis que JDE met à disposition un grand livre et un fichier auxiliaire de type RP. Ces derniers contiennent des informations détaillées sur les transactions, avec des champs tels que le type et numéro de document, le code client, les montants comptabilisés, ainsi que les périodes fiscales correspondantes. L'ensemble de ces trois sources couvre toutes les opérations réalisées entre le début de

l'année 2025 et le 14 juillet 2025, offrant une base de données complète et structurée pour l'analyse et la détection d'anomalies comptables.

Pour répondre à cette problématique, les questions suivantes ont été définies :

Question 1 : Comment exploiter les données du grand livre général, contenant des écritures comptables de tailles variées, pour détecter efficacement les anomalies à l'aide des techniques d'apprentissage automatique ?

Question 2 : Quelles approches permettent d'améliorer la configuration et la performance des modèles d'apprentissage automatique pour la détection des anomalies dans les données du grand livre général ?

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Travaux antérieurs

1.1 Détection d'anomalies dans les données financières

Ces dernières années, l'apprentissage automatique a été de plus en plus utilisé pour détecter des anomalies dans les données financières, notamment dans le cadre de l'audit et de la conformité. Contrairement aux méthodes basées sur des règles, les techniques ML permettent d'identifier automatiquement des motifs cachés et des écarts par rapport à la normalité, même dans des jeux de données volumineux ou évolutifs.

Les anomalies recherchées incluent les erreurs comptables, les fraudes, les violations de conformité et les écarts par rapport aux normes. Différentes approches ont été explorées, allant des méthodes supervisées et non supervisées aux réseaux de neurones auto-encodeurs, utilisant à la fois des données réelles et synthétiques issues des systèmes ERP. L'objectif principal est de détecter automatiquement les transactions présentes dans Afriware mais manquantes dans les fichiers JDE, en appliquant plusieurs modèles supervisés [2, 4, 3, 9, 6]

Certaines grandes entreprises comptables ont également développé des outils propriétaires basés sur l'apprentissage automatique pour améliorer l'efficacité des audits et détecter les anomalies dans le grand livre général, intégrant souvent des solutions explicatives pour interpréter les modèles « boîte noire » c'est-à-dire des modèles complexes dont le fonctionnement interne n'est pas directement compréhensible par l'utilisateur. Bien que ces modèles offrent des performances élevées, il est difficile de savoir pourquoi une certaine transaction est considérée comme anormale. Pour pallier cette limitation, les entreprises intègrent des solutions explicatives permettant d'interpréter les décisions du modèle et de fournir des justifications compréhensibles pour les auditeurs, garantissant ainsi la confiance et la transparence dans le processus de détection des anomalies.

En résumé, l'apprentissage automatique offre des méthodes adaptatives et précises pour détecter les irrégularités financières et optimiser les processus d'audit.

1.2 Techniques d'apprentissage automatique supervisées et non supervisées pour la détection d'anomalies

L'apprentissage automatique, qui consiste à concevoir des algorithmes capables d'apprendre des motifs réguliers à partir de données, se divise principalement en trois grandes catégories : **supervisé**, **non supervisé** et **semi-supervisé**.

Dans le cas de l'**apprentissage supervisé**, les données sont étiquetées, ce qui permet au modèle de connaître la sortie attendue pour chaque entrée. L'algorithme ajuste ses paramètres afin de prédire soit des classes (classification), soit des valeurs numériques continues (régression). Ce type d'apprentissage est le plus couramment utilisé en pratique.

À l'inverse, l'**apprentissage non supervisé** ne nécessite pas de données étiquetées, même si celles-ci peuvent exister dans le jeu de données d'origine. Le modèle apprend uniquement en fonction des propriétés et de la structure interne des données. L'objec-

tif principal est donc d'*extraire des structures cachées* plutôt que de prédire des classes prédéfinies. Parmi les approches les plus répandues, on retrouve les méthodes de *clustering* (basées sur la distance ou la densité) et l'extraction de règles d'association. Certains réseaux neuronaux, comme les auto-encodeurs, peuvent également être utilisés dans ce cadre, et parfois considérés comme des modèles semi-supervisés ou auto-supervisés.

Le **choix du type de modèle** pour la détection d'anomalies dépend principalement de la disponibilité des étiquettes et de la nécessité d'identifier des motifs inconnus :

— **Apprentissage supervisé** : chaque instance est étiquetée, indiquant si elle est normale ou anormale. La tâche peut être formulée comme une classification binaire (normal/anormal) ou une classification multi-classes (plusieurs types d'anomalies). Dans le domaine comptable, cette approche est très utilisée. Les principaux classificateurs appliqués incluent la régression logistique, les machines à vecteurs de support (SVM), les arbres de décision, les k-plus proches voisins (K-NN), le naïve Bayes et les réseaux neuronaux.
— **Apprentissage non supervisé** : aucune étiquette n'est utilisée lors de l'entraînement. Le modèle parcourt l'ensemble des données et détecte automatiquement les instances qui s'écartent de ce qu'il considère comme la "normalité". Cette approche est particulièrement utile en l'absence d'étiquettes, ou lorsque l'on suspecte l'existence d'anomalies nouvelles ou imprévues.

Les méthodes supervisées, incluant Random Forest, SVM, kNN, Logistic Regression et Decision Tree, ont été largement étudiées pour la classification et la détection d'anomalies [2, 4, 3, 9, 6]. Les approches non supervisées permettent de détecter des fraudes inconnues mais peuvent être moins performantes sur de grandes bases de données [8].

Dans ce travail, nous nous concentrerons sur la grande famille des modèle supervisés et nous procéderons à une comparaison de leurs performances et de leurs limites dans le cadre de la détection d'anomalies.

Chapitre 2

Méthodologie de recherche

2.1 Justification du choix des algorithmes

La popularité des techniques d'apprentissage automatique supervisé et non supervisé croît de manière quasi linéaire avec le temps, tandis que la performance des solutions dépend à la fois des caractéristiques qualitatives et quantitatives des données et de l'algorithme utilisé. Dans ce travail, nous avons mis en œuvre cinq techniques fondamentales d'apprentissage supervisé pouvant être utilisées pour construire un modèle de classification destiné à la détection d'anomalies. Ces approches incluent des méthodes probabilistes et statistiques, adaptées aux espaces de grande dimension, basées sur des arbres non paramétriques, des méthodes non généralisantes et des techniques d'apprentissage de représentation. Ces techniques ont déjà été appliquées avec succès dans le contexte de la comptabilité financière.

L'apprentissage supervisé nécessite des données étiquetées. Dans ce travail, nous disposons d'un jeu de données fortement déséquilibré, contenant un faible pourcentage d'anomalies . De plus, le jeu de données original peut contenir certaines anomalies non étiquetées. Les auditeurs s'intéressent également à la détection de motifs d'anomalies inconnues, tandis que nous cherchons ici à identifier celles qui sont étiquetées. L'apprentissage non supervisé permet de détecter des écarts rares par rapport à la normalité, couvrant à la fois les anomalies étiquetées et les anomalies inconnues. Afin de maintenir un équilibre pertinent entre les modèles supervisés et non supervisés dans cette étude, nous nous appuyons sur les observations antérieures qui soulignent la prévalence de l'apprentissage supervisé dans le contexte comptable.

2.2 Méthodologies en science des données

Les projets de science des données dans le milieu académique et industriel partagent des concepts communs de fouille de données, comprenant plusieurs processus et méthodes. Afin de standardiser ces activités, un certain nombre de méthodologies de fouille de données ont été formalisées. Bien que suivant généralement des étapes similaires, ces méthodologies diffèrent par la granularité des étapes, ainsi que par la manière et le moment de leur application.

Selon une étude approfondie, nous pouvons distinguer les trois méthodologies les plus populaires et largement reconnues : **SEMMA**, **KDD** et **CRISP-DM**. De plus, la méthodologie **TDSP** attire de plus en plus l'attention, car elle est hautement personnalisable et offre un support aux projets ayant initialement utilisé d'autres méthodologies.

— **SEMMA.** La méthodologie SEMMA se compose de cinq étapes qui définissent son acronyme : *Sample*, *Explore*, *Modify*, *Model* et *Assess*. Ces étapes forment un cycle fermé qui se répète jusqu'à ce que l'objectif soit atteint. Le nombre réduit d'étapes et leur simplification rendent cette méthodologie facile à comprendre et à adopter.

— **KDD.** KDD signifie *Knowledge Discovery in Databases*. Bien qu'elle comporte 5 étapes principales, des phases *Pré-KDD* et *Post-KDD* sont également reconnues, afin de comprendre les objectifs de l'utilisateur au départ, et enfin d'intégrer la

solution développée dans les processus existants. Les étapes incluent : sélection, prétraitement, transformation (fouille de données), fouille de données et interprétation (évaluation). Dans KDD, les étapes sont exécutées de manière itérative et interactive.

— **CRISP-DM.** La méthodologie *Cross-Industry Standard Process for Data Mining (CRISP-DM)* comporte 6 étapes, commençant par la compréhension du métier et se terminant par le déploiement de la solution développée. Les tâches liées aux données sont traitées dans les étapes de compréhension des données, préparation des données et modélisation, puis évaluées dans l'étape d'évaluation. Des itérations peuvent être effectuées à tout moment, de la compréhension métier jusqu'à l'évaluation.

— **TDSP.** Le *Team Data Science Process (TDSP)* est une méthodologie développée par Microsoft pour livrer efficacement des solutions en analyse prédictive et en intelligence artificielle. Elle comprend 5 étapes, dont 4 cycliques avec des connexions bidirectionnelles : compréhension métier, acquisition et compréhension des données, modélisation, déploiement, et une 5^{e} étape dite d'acceptation client, marquant la fin du cycle de livraison d'une solution. En milieu industriel, outre l'agilité d'exécution et le travail d'équipe, l'ajout de critères d'acceptation client constitue un atout important, prolongeant ainsi KDD et CRISP-DM.

Selon les comparaisons effectuées, la méthodologie SEMMA est d'origine industrielle et se base sur les principes de KDD, tandis que CRISP-DM, issue d'un partenariat entre l'industrie et le milieu académique, s'appuie également sur KDD.

CRISP-DM introduit directement des étapes de compréhension métier et de déploiement, absentes dans SEMMA et KDD. Étant la plus populaire et complète, CRISP-DM se distingue par sa flexibilité, permettant d'inverser l'ordre des étapes.

Quant à TDSP, elle est particulièrement adaptée à l'industrie, alors qu'en milieu académique, on accorde moins d'importance à l'agilité en équipe et à l'acceptation client. Dans ce travail, nous avons suivi le cadre standard CRISP-DM afin de mener efficacement et de manière structurée les activités de fouille de données.

La Figure 2.1 illustre les étapes du CRISP-DM dans leur séquence, tout en permettant des itérations et un ordre réversible. Cette méthodologie est souvent représentée en mettant les données au centre du processus analytique. La possibilité de revenir en arrière est essentielle, car les découvertes faites sur les données et les algorithmes peuvent remettre en question les hypothèses initiales, ce qui reflète fidèlement le déroulement réel des projets en science des données.

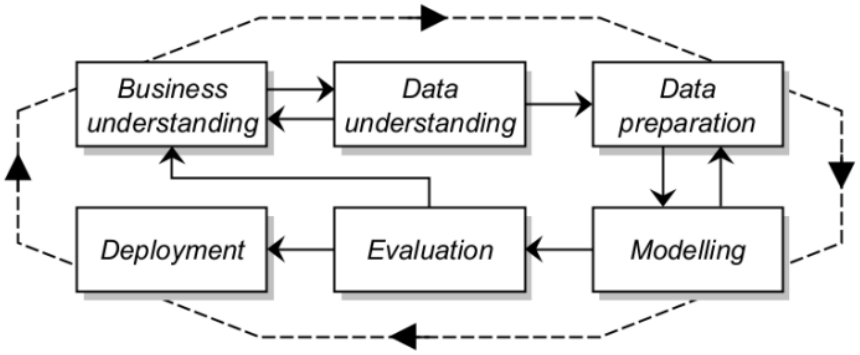

FIGURE 2.1 – Illustration des étapes CRISP-DM

2.3 CRISP-DM : Étapes principales

— **Business understanding :** Comprendre l'objectif métier est la base de tout processus analytique. Ne pas relier les activités prévues à l'objectif métier réel conduit à des échecs dans l'évaluation des résultats et dans la mise en œuvre de la solution développée. Lorsqu'un contexte est connu, il est nécessaire de définir les objectifs et les critères de succès. Tout cela n'est pas étudié de manière indépendante des données, car il est crucial de comprendre que les données nécessaires au projet peuvent être extraites et traitées.

— **Data understanding :** Les processus centrés sur les données nécessitent une compréhension de la manière dont elles peuvent être collectées et de ce qu'elles représentent, tout en reliant cette connaissance à la compréhension métier. À ce stade, nous décrivons le ou les jeux de données collectés et réalisons une analyse exploratoire. La qualité des données doit être évaluée et rapportée. D'autres transformations de données ne doivent pas être incluses dans cette étape.

— **Data preprocessing :** Avant que les données puissent être utilisées dans l'étape de modélisation, des transformations courantes doivent être effectuées, notamment la sélection (échantillonnage), le nettoyage, la sélection de variables, l'ingénierie des caractéristiques et les transformations requises par les algorithmes telles que le reformatage, l'encodage, la mise à l'échelle, l'équilibrage des classes et la réduction de dimensionnalité.

— **Modeling :** La mise en œuvre réelle de la partie algorithmique est réalisée à cette étape. Les données préparées permettent d'entraîner des modèles d'apprentissage automatique ou d'appliquer d'autres méthodes statistiques afin d'atteindre l'objectif défini précédemment. Dans le cas du machine learning, la sélection des algorithmes et des hyperparamètres, la composition et l'ajustement des modèles sont effectués. L'évaluation des performances des modèles et l'obtention des valeurs des métriques choisies suivent l'entraînement.

— **Evaluation :** Les résultats empiriques obtenus dans les étapes précédentes doivent

être évalués, et si plusieurs modèles ont été entraînés, comparés entre eux. Les critères de succès métier identifiés dans l'étape de compréhension du métier sont pris en compte. Le succès global du processus de modélisation est discuté et une décision est prise concernant la sélection du modèle et les étapes suivantes.

— **Deployment :** Si la solution s'avère concluante, une décision peut être prise de l'implémenter dans un environnement existant. Pour un modèle de machine learning, un plan de déploiement est élaboré, incluant le déploiement effectif, la maintenance et la documentation. Cette étape peut également servir à diffuser les connaissances acquises sur la modélisation des données de manière organisée. Dans ce cas, elle se limite à la discussion sur la valeur ajoutée de la solution développée dans l'atteinte de l'objectif métier, en comparant l'état antérieur du problème dans le milieu industriel ou académique et les implications de la solution appliquée.

Chapitre 3

Résultats

3.1 Business Understanding

Le domaine de l'audit intelligent émerge en réponse aux défis croissants auxquels sont confrontés les auditeurs. La massification des données financières et la complexité accrue des transactions rendent les méthodes de contrôle traditionnelles, souvent basées sur l'échantillonnage, à la fois inefficientes et insuffisantes. Ces approches conventionnelles, bien qu'éprouvées, présentent un risque inhérent de ne pas détecter des anomalies subtiles, disséminées au sein de vastes volumes de données, ce qui peut compromettre la fiabilité des états financiers. L'objectif business fondamental est donc de passer d'un audit par sondage à un audit continu et exhaustif, capable d'examiner la totalité des écritures comptables pour identifier des irrégularités potentielles avec une précision et une rapidité inégalées.

La valeur stratégique de la détection d'anomalies via le Machine Learning réside dans sa capacité à transformer la fonction d'audit. Elle permet non seulement d'améliorer l'efficacité opérationnelle en automatisant la revue de routines et en libérant les auditeurs pour des tâches à plus forte valeur ajoutée comme l'analyse approfondie et le jugement professionnel, mais aussi de renforcer significativement la couverture et la qualité du contrôle. En analysant l'intégralité des journaux, la solution vise à réduire le risque de fraude ou d'erreur non détectée, offrant ainsi une assurance plus robuste aux parties prenantes et une meilleure protection des actifs de l'organisation. Il s'agit in fine d'une démarche proactive de gestion des risques financiers et de préservation de l'intégrité comptable.

3.2 Data Understanding

Dans le cadre de notre analyse, nous utilisons trois sources de données principales : `afriware`, `jde_f0911` et `jde_f03b11`. Chaque fichier contient des colonnes décrivant des informations comptables, analytiques ou transactionnelles. Nous détaillons ci-dessous les attributs, en soulignant les différences importantes entre colonnes portant des noms similaires.

3.2.1 Fichier `afriware`

Le fichier `afriware` Figure 3.1 regroupe les factures issues du système Afriware. Les attributs disponibles sont :

- **TypeFacture** : nature de la facture (par exemple, facture client, avoir).
- **NumeroFacture** : identifiant unique attribué à chaque facture dans Afriware.
- **NumeroLigne** : numéro de ligne associé à une facture dans Afriware. Contrairement au fichier `jde_f0911`, ce numéro correspond à l'entrée (ou ligne) d'une facture donnée, et non pas au nombre de lignes rattachées à un client.
- **CodeClient** : code identifiant le client destinataire de la facture.
- **MontantHT** : montant hors taxes facturé.
- **MontantTTC** : montant toutes taxes comprises.
- **Taxes** : montant des taxes appliquées à la facture.
- **DateCreation** : date initiale de création de la facture dans le système.

— **DateModification** : date de la dernière mise à jour de la facture.
— **DateEDI** : date de transfert de la facture vers le système JDE via EDI.
— **ReferenceEDI** : identifiant de référence du transfert EDI. Si ce champ est renseigné, la facture doit exister dans JDE ; s'il est vide ou nul, la facture n'a pas encore été transférée.
— **CentreAnalyse** : centre analytique rattaché à la facture.
— **CompteProduit** : compte produit associé à la ligne de facture.
— **DateFacture** : date d'encaissement ou de facturation effective de la facture.

3.2.2 Fichier `jde_f0911`

Le fichier `jde_f0911` Figure 3.2 correspond aux écritures du grand livre issues du système JDE. Les colonnes principales sont :
— **GLKCO** : numéro de la société émettrice.
— **GLDCT** : type de facture (même rôle que `TypeFacture` dans Afriware).
— **GLDOC** : numéro de facture (équivalent à `NumeroFacture`).
— **GLDGJ** : jour de la facture exprimé en format julien, nécessitant une conversion.
— **GLJELN** : numéro de ligne associé à un client dans JDE. Contrairement à `NumeroLigne` d'Afriware (qui est le rang de la ligne au sein d'une facture), ce numéro indique le nombre de lignes rattachées au client et peut donc différer.
— **GLMCU** : centre analytique.
— **GLOBJ** : compte produit.
— **GLSUB** : sous-compte produit.
— **GLAA** : montant hors taxes.
— **GLU** : montant hors taxes restant à payer.
— **GLAN8** : code client.
— **GLCTRY** : siècle correspondant à la facture.
— **GLFY** : année fiscale de la facture.
— **GLPN** : mois de la facture.

3.2.3 Fichier `jde_f03b11`

Le fichier `jde_f03b11` Figure 3.3 contient des informations sur les factures et leur état de règlement. Les colonnes disponibles sont :
— **RPKCO** : numéro de la société.
— **RPDCT** : type de facture.
— **RPDOC** : numéro de facture.
— **RPSFX** : jour de la facture (en format julien, à convertir).
— **RPAG** : montant TTC.
— **RPAAP** : montant TTC restant à payer.
— **RPAN8** : code client.
— **RPCTY** : siècle associé à la facture.
— **RPFY** : année de la facture.
— **RPPN** : mois de la facture.

Il est inévitable que certaines erreurs financières passent inaperçues en raison du **risque**

d'échantillonnage et de l'existence de motifs d'incohérence peu connus. Compte tenu des applications étudiées dans l'industrie et le milieu académique, il existe un fort potentiel pour résoudre ce problème en utilisant des méthodes statistiques et de *machine learning*.

Les données issues des systèmes Afriware et JDE (fichiers `jde_f0911` et `jde_f03b11`) ont été collectées et mises à disposition pour ce travail. Dans ce contexte, l'anomalie que nous cherchons à détecter est la suivante : une transaction présente dans le fichier `afriware` mais absente du fichier `jde_f0911`, du fichier `jde_f03b11`, ou des deux simultanément. Ces anomalies représentent un intérêt majeur pour les auditeurs, car elles traduisent un défaut potentiel de transfert ou d'enregistrement comptable entre les systèmes.

Plusieurs facteurs augmentent la complexité de ce processus, notamment : la multiplicité des sources de données, les plans de comptes spécifiques à chaque entreprise et système, ainsi que la diversité des pratiques industrielles concernant l'enregistrement des transactions commerciales.

En utilisant les données fournies, nous appliquerons des techniques de *machine learning* afin d'identifier efficacement ces anomalies de correspondance entre Afriware et JDE, dans le but d'améliorer la qualité et la fiabilité des audits, tout en optimisant les processus d'échantillonnage.

TypeFacture	NumFacture	NumLigne	CodeClient	MontantHT	MontantTTC	Taxes	DateCreation	DateModification	DateEDI	ReferenceEDI	CentreAnalyse	CompteProduit	DateFacture
BJ	72448457	1	12008	-47090916	-47090916	0	7/11/2025 15:26	7/11/2025 15:26	7/12/2025 0:02	EDI	10001	449910.92	6/15/2025 0:00
BJ	72448432	1	12000	-67500	-67500	0	6/17/2025 15:02	6/17/2025 15:02	6/18/2025 0:02	EDI	11021	449910.95	6/17/2025 0:00
BJ	72448432	2	12000	-33750	-33750	0	6/17/2025 15:02	6/17/2025 15:02	6/18/2025 0:02	EDI	11021	449910.95	6/17/2025 0:00
BJ	72448434	1	118594	-1720364.4	-1720364.4	0	6/20/2025 9:53	6/20/2025 9:53	6/21/2025 0:02	EDI	10001	711900.84	6/20/2025 0:00
BJ	72448435	1	13332	-810000	-810000	0	6/20/2025 15:13	6/20/2025 15:13	6/21/2025 0:02	EDI	10001	449910.92	6/19/2025 0:00

FIGURE 3.1 – Échantillon de données anonymisées du fichier afriware.

GLKCO	GLDCT	GLDOC	GLDGJ	GLJELN	GLMCU	GLOBJ	GLSUB	GLSUB2	GLAA	GLU	GLAN8	GLCTRY	GLFY	GLPN
1	BF	22348535	125031	1	101.34211	101	342110		540000	0	11601	20	25	1
1	BF	22348536	125031	1	101.34211	101	342110		540135	0	10201	20	25	1
1	BF	22348537	125043	1	101.34211	101	342110		594000	0	11101	20	25	2
1	BF	22348539	125036	1	101.34211	101	342110		810000	0	801	20	25	2
1	BF	22348541	125065	1	101.34211	101	342110		1081863	0	1001	20	25	3

FIGURE 3.2 – Échantillon de données anonymisées du fichier jde_f0911.

RPKCO	RPDCT	RPDOC	RPSFX	RPAG	RPAAP	RPAN8	RPCTY	RPFY	RPPN
1	BJ	72447916	125003	-4860000	0	1200501	20	25	1
1	BJ	72447961	125006	-13500000	0	1200501	20	25	1
1	BJ	72447962	125006	-86400000	0	1333201	20	25	1
1	BJ	72447964	125010	-3099465	-3099465	1694101	20	25	1
1	BJ	72447965	125010	-2021004	-2021004	11349701	20	25	1

FIGURE 3.3 – Échantillon de données anonymisées du fichier jde_f03b11.

Table 3.1 – Colonnes du fichier Afriware avec leurs types.

N	Column Name	Column Type
0	TypeFacture	string
1	NumeroFacture	int
2	NumeroLigne	int
3	Code client	int
4	MontantHT	float
5	MontantTTC	float
6	Taxes	float
7	DateCreation	datetime
8	DateModification	datetime
9	DateEDI (Date du transfert)	datetime
10	RéférenceEDI	string
11	CentreAnalyse	float
12	CompteProduit	float
13	DateFacture	datetime

Table 3.2 – Colonnes du fichier JDE_F0911 (Grand Livre Général) avec leurs types.

N	Column Name	Column Type
0	GLKCO (Numero de la société)	int
1	GLDCT (TypeFacture)	string
2	GLDOC (TypeFacture)	int
3	GLDGJ (Date Julienne)	int
4	GLJELN (Numéro de ligne)	int
5	GLMCU (CentreAnalyse)	float
6	GLOBJ (Compte produit)	int
7	GLSUB (Sous-compte produit)	int
8	GLSUB2 (Sous-compte produit 2)	float
9	GLAA (MontantHT)	float
10	GLU (MontantHT restant à payer)	float
11	GLAN8 (Code client)	int
12	GLCTRY (siècle de la facture)	int
13	GLFY (année de la facture)	int
14	GLPN (Mois de la facture)	int

Table 3.3 – Colonnes du fichier JDE_F03B11 (Comptes Clients) avec leurs types.

N	Column Name	Column Type
0	RPKCO (Numero de la société)	int
1	RPDCT (TypeFacture)	string
2	RPDOC (NumeroFacture)	int
3	RPSFX (Date Julienne)	int
4	RPAG (MontantTTC)	float
5	RPAAP (MontantTTC restant à payer)	float
6	RPAN8 (Code client)	int
7	RPCTY (siècle de la facture)	int
8	RPFY (année de la facturee)	int
9	RPPN (Mois de la facture)	int

Dans le cadre de notre étude, nous avons enrichi le fichier *Afriware* en y ajoutant deux colonnes : *in_jde911* et *in_jde311*. Ces colonnes indiquent respectivement si une transaction est présente dans le fichier *JDE_F0911* et dans le fichier *JDE_F03B11*, en prenant la valeur 0 en cas de présence et 1 sinon. L'anomalie que nous cherchons à détecter correspond au cas où une transaction est enregistrée dans *Afriware*, mais absente à la fois de *JDE_F0911* et de *JDE_F03B11* (c'est-à-dire lorsque *in_jde911* = 1 et *in_jde311* = 1).

Nous considérons les autres transactions comme normales et leur attribuons la valeur 1, traduisant l'absence d'anomalie. Toutefois, il est important de souligner que, selon l'avis des auditeurs, les données initiales peuvent contenir jusqu'à 1 à 3% d'anomalies .

La répartition des écritures considérées comme normales et de celles identifiées comme anormales est présentée dans le Tableau 3.4 et Tableau 3.5.

Table 3.4 – Comptage des transactions présente dans le fichier Afriware et manquante dans le fichier JDE_311.

Label	Nombre
Données normales	432,826
Anomalies	49,846

Table 3.5 – Comptage des transactions présente dans le fichier Afriware et manquante dans le fichier JDE_911.

Label	Nombre
Données normales	432,826
Anomalies	49,846

Remarque : On observe que le nombre d'anomalies est identique dans les fichiers `JDE311` et `JDE911` (**49 846**). Étant donné que les anomalies correspondent aux transactions présentes dans le fichier `Afriware` mais absentes dans les fichiers `JDE`, cette égalité suggère que les deux fichiers JDE reflètent le même ensemble de transactions manquantes.

TABLE 3.6 – Nombre de valeurs distinctes par colonne dans le fichier Afriware.

Colonne	**Nb de valeurs uniques**
TypeFacture	14
NumFacture	142,554
NumLigne	2,087
CodeClient	6,967
DateCreation	44,665
DateModification	48,899
DateEDI	1,776
ReferenceEDI	1
CentreAnalyse	346
CompteProduit	31

On peut observer que certaines colonnes présentent un nombre élevé de valeurs uniques, comme *NumFacture* avec 142 554 valeurs uniques et *CodeClient* avec 6 967 valeurs uniques, ce qui en fait des attributs à haute cardinalité. Ces colonnes pourront être utilisées pour identifier ou grouper les transactions au niveau de la ligne de facture. D'autres colonnes, comme *TypeFacture* ou *CompteProduit*, ont un nombre plus limité de valeurs uniques et serviront davantage à catégoriser les transactions. Il est à noter que certaines colonnes présentent très peu de valeurs uniques, comme *ReferenceEDI* avec seulement 1 valeur unique, ce qui indique une information constante pour l'ensemble des lignes. Les colonnes de dates, telles que *DateCreation* et *DateModification*, montrent un nombre intermédiaire de valeurs uniques, reflétant la diversité des transactions dans le temps.

3.3 Data Preprocessing.

3.3.1 Data Cleaning

Nous avons vérifié le jeu de données pour détecter d'éventuelles valeurs manquantes dans les enregistrements, et il n'y en avait aucune. La suppression des doublons a été effectuée afin de garantir l'absence de lignes répétées dans les données. Les types de données des colonnes ont été harmonisés : les valeurs catégorielles sont devenues des chaînes de caractères et les colonnes de dates ont été converties en type `datetime`. Les colonnes numériques continues sont **MontantHT**, **MontantTTC** et **Taxes**. Aucune valeur dans les données ne paraît incohérente par rapport à la colonne à laquelle elle appartient.

3.3.2 Feature Selection

La sélection des variables (feature selection) est une étape cruciale qui influence directement l'efficacité de l'analyse des transactions. Dans le cadre de cette étude, les informations clés pour identifier les anomalies se trouvent dans les colonnes catégorielles telles que **TypeFacture**, **CodeClient**, **CompteProduit** ou **CentreAnalyse**, ainsi que dans le numéro de facture **NumFacture**. Les montants financiers (**MontantHT**, **MontantTTC**, **Taxes**) sont utiles pour des analyses globales ou des ratios, mais leur inclusion dans la détection des anomalies pourrait générer des faux positifs liés à des écarts monétaires ponctuels et ne contribuerait pas à identifier les factures manquantes ou incohérentes entre Afriware et JDE. De même, les colonnes de dates peuvent introduire des variations non pertinentes pour ce type d'analyse. Par conséquent, le modèle se concentre principalement sur les **valeurs catégorielles** et la structure des factures pour détecter les anomalies de présence ou d'absence dans les fichiers JDE.

3.3.3 Feature Engineering

Dans le cadre de l'ingénierie des caractéristiques (*feature engineering*), certaines colonnes du jeu de données ont été supprimées afin de ne conserver que les variables pertinentes pour l'analyse. Les colonnes `NumLigne`, `TypeFacture`, `DateFacture`, `DateCreation`, `DateModification`, `DateEDI` et `ReferenceEDI` ont été retirées car elles ne contribuent pas directement à la prédiction ou à l'analyse souhaitée. En effet, `NumLigne` et `ReferenceEDI` sont des identifiants uniques qui n'apportent pas d'information discriminante, tandis que `TypeFacture` et les dates sont principalement descriptives et peuvent introduire du bruit si elles ne sont pas transformées en variables numériques ou catégorielles exploitables. Cette étape permet ainsi de simplifier le modèle et de se concentrer sur les caractéristiques réellement informatives, telles que les montants ou les codes analytiques.

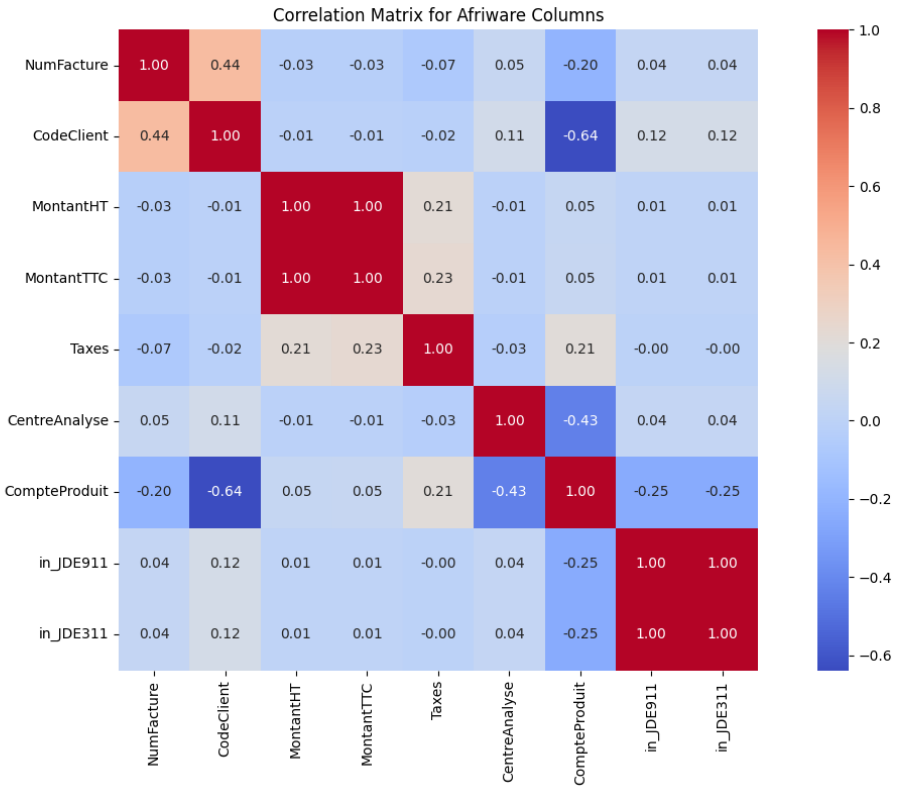

FIGURE 3.4 – Matrice de corrélation entre les colonnes du fichier Afriware.

L'analyse de la matrice de corrélation dans la Figure 3.4 a révélé des relations significatives entre certaines variables, guidant ainsi la sélection des caractéristiques pour la modélisation. Les colonnes CodeClient, CompteProduit et CentreAnalyse ont été retenues en raison de leurs corrélations fortes avec les variables cibles in_JDE911 et in_JDE311

Plus précisément, on observe une corrélation négative marquée entre CompteProduit et les variables cibles (-0.25), tandis que CodeClient présente une corrélation positive modérée (0.12). Bien que la valeur absolue de ces coefficients puisse sembler modeste, leur importance statistique est renforcée par le contexte métier et la stabilité de ces relations à travers les deux variables cibles.

De plus, la variable CentreAnalyse montre un pattern de corrélation cohérent (0.04) qui, combiné à sa pertinence métier dans le processus de détection d'anomalies, justifie son inclusion. Cette sélection stratégique permet de concentrer l'analyse sur les variables les plus informatives, améliorant ainsi l'efficacité du modèle d'apprentissage automatique tout en réduisant la dimensionalité des données.

TABLE 3.7 – Variables sélectionnées pour l'analyse des transactions.

NumFacture	**CodeClient**	**MontantHT**	**MontantTTC**	**Taxes**	**CentreAnalyse**	**CompteProduit**
	✓				✓	✓

3.3.4 Data Aggregation and Invoice-Level Grouping

Dans un premier temps, nous avons analysé les transactions présentes dans le fichier Afriware mais manquantes dans les fichiers JDE, et avons identifié 49 846 transactions manquantes. Ensuite, nous avons procédé à un regroupement des transactions par numéro de facture (NumFacture) afin de consolider les informations au niveau de la facture. Après ce regroupement, le nombre de transactions manquantes par facture est passé à 14 069, ce qui correspond au nombre unique de factures absentes dans JDE.

Pour confirmer la cohérence des données, nous avons également regroupé les 49 846 transactions initiales manquantes par NumFacture et retrouvé exactement 14 069 factures uniques, ce qui montre que chaque facture manquante peut correspondre à plusieurs lignes dans Afriware.

Enfin, nous avons calculé pour chacune de ces 14 069 factures le nombre d'occurrences de cette facture dans les 49 846 transactions initiales. En sommant toutes les occurrences, nous avons retrouvé le nombre initial de 49 846 transactions, ce qui confirme que si une facture est présente à la fois dans Afriware et dans JDE, alors toutes les lignes correspondant à cette même facture doivent également être présentes dans les deux fichiers.

Ainsi, cette étape a permis de valider la cohérence des regroupements et d'identifier précisément les transactions manquantes au niveau des factures.

Les transactions du fichier *Afriware* ont été consolidées au niveau de chaque facture afin de regrouper toutes les lignes correspondant à un même numéro de facture. Pour chaque facture, les montants financiers tels que le `MontantHT`, le `MontantTTC` et les `Taxes` ont été additionnés afin de refléter le total par facture, tandis que les autres informations — dates de création, de modification et de transfert EDI, type de facture, code client, centre analytique, compte produit et numéro de ligne — ont été conservées à partir de la première occurrence ou du maximum pour certaines valeurs, assurant ainsi la cohérence des données.

Cette approche a permis de réduire la granularité des transactions, en passant d'un niveau ligne à un niveau facture, tout en préservant l'ensemble des informations essentielles. Les montants ont été arrondis et les colonnes réorganisées pour faciliter l'analyse ultérieure. Cette consolidation a été une étape clé pour identifier les factures manquantes ou présentes dans les fichiers JDE, et préparer les données pour les contrôles et analyses suivantes.

3.4 Modeling

Nous avons entraîné cinq modèles d'apprentissage automatique supervisés , en utilisant un algorithme différent pour chaque modèle, afin de détecter les valeurs aberrantes dans l'ensemble de données prétraité. Les modèles supervisés prédiront une classe dichotomique, ce qui en fait un problème de classification binaire. Dans cette section, nous décrivons les processus de modélisation empirique et les résultats.

3.4.1 Train-Test Data Split

Pour l'apprentissage supervisé, il est nécessaire de diviser les données en ensembles d'entraînement et de test, ou en ensembles d'entraînement, de test et de validation. La manière dont les données doivent être divisées dépend fortement de la quantité de données et d'autres propriétés, parmi lesquelles la distribution des classes. Dans notre ensemble de données, nous avons très peu d'anomalies étiquetées par rapport à la quantité totale. De plus, bien que nous ayons défini un problème de classification binaire, nous voulons que notre modèle apprenne et soit testé sur l'anomalie definite.

Pour un modèle d'apprentissage automatique, il existe des paramètres inconnus qui doivent être identifiés à partir d'un ensemble d'entraînement lors de l'ajustement du modèle . Le ratio de division des données entre entraînement et test est important pour la performance du modèle. A. Gholamy, V. Kreinovich et O. Kosheleva [5] ont pédagogiquement expliqué un équilibre empiriquement reconnu de ratios de pourcentage de 70-30 ou 80-20. Ayant plus d'anomalies étiquetées, il serait raisonnable de réaliser une analyse de sensibilité de la performance d'un modèle pour différents ratios d'entraînement-test.

Dans notre scénario, nous avons divisé nos données en deux ensembles : des données d'entraînement et de test, représentant respectivement 80 % et 20 % du total. Lors de la division des données, nous avons dû stratifier par type d'anomalie, afin que des proportions égales de chaque type soient présentes dans les données d'entraînement et de test. Pour des fins de reproductibilité et de comparaison appropriée des modèles, nous avons défini une valeur d'état aléatoire (random state) lors de la division.

3.4.2 Models Performance Measure

Les modèles de classification binaire prédisent des classes négatives (0) et positives (1). Dans notre travail, nous traitons la classe d'anomalie comme positive. Il existe plusieurs métriques de classification qui peuvent être calculées pour évaluer les modèles de classification binaire, et les plus utilisées sont l'exactitude (accuracy), la précision (precision), le rappel (recall) et le score F1 . Ces métriques peuvent être calculées à l'aide d'une matrice de confusion qui résulte des valeurs de représentation des classes prédites par un modèle.

Dans la Figure 3.5, une structure de matrice de confusion est expliquée en termes de classes vrai/faux positifs et négatifs en utilisant la classe réelle sur l'axe des y et la classe prédite sur l'axe des x. En prédisant une classe avec un modèle d'apprentissage automatique entraîné, nous pouvons extraire les valeurs des décomptes correspondants des vrais négatifs (TN), vrais positifs (TP), faux négatifs (FN) et faux positifs (FP). Toutes les métriques mentionnées peuvent être calculées aux niveaux micro, macro moyenne et moyenne pondérée. Le choix des bonnes métriques pour l'évaluation d'un modèle dépend du problème et des propriétés d'équilibre des classes. Par exemple, la précision (accuracy) ne serait pas une métrique équitable pour évaluer un modèle sur un ensemble de données avec des classes déséquilibrées, comme dans notre cas.

		Predicted class	
		Non-anomaly (0)	**Anomaly (1)**
Actual class	**Non-anomaly (0)**	*True Negative (TN)*	*False Positive (FP)*
	Anomaly (1)	*False Negative (FN)*	*True Positive (TP)*

FIGURE 3.5 – Matrice De Confusion.

Ci-dessous, nous avons compilé une liste exhaustive des formules pour calculer les métriques que nous aborderons dans ce travail :

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$
$$Recall_1 / Specificity = \frac{TP}{TP + FN}$$
$$Recall_0 / Sensitivity = \frac{TN}{TN + FP}$$
$$Precision_1 = \frac{TP}{TP + FP}$$
$$Precision_0 = \frac{TN}{TN + FN}$$
$$F1 - score_1 = \frac{2 * Recall_1 * Precision_1}{Recall_1 + Precision_1}$$
$$F1 - score_0 = \frac{2 * Recall_0 * Precision_0}{Recall_0 + Precision_0}$$

$$Metric\_avg_{macro} = \frac{Metric_0 + Metric_1}{2}$$
$$Metric\_avg_{weighted} = \frac{Metric_0 * Support_0 + Metric_1 * Support_1}{Support_0 + Support_1}$$

FIGURE 3.6 – Liste des formules.

Dans la liste des formules 3.6, le rappel (*recall*), la précision (*precision*) et le score F1 sont calculés au **niveau micro (classe)**. Les **moyennes macro et pondérée** peuvent être calculées à l'aide des deux dernières formules.

Exactitude (Accuracy). La métrique d'exactitude représente le pourcentage de classes correctement prédites, qu'elles soient positives ou négatives. Elle est utilisée pour les ensembles de données équilibrés, lorsque les vrais positifs et les vrais négatifs sont tous deux importants.

Précision (Precision). La précision mesure la proportion de classes positives ou négatives correctement identifiées parmi toutes les classes prédites comme positives ou négatives. Elle est utilisée lorsque l'on souhaite **minimiser le nombre de faux positifs**. Cependant, ajuster un modèle selon cette métrique peut conduire à manquer certains vrais positifs.

Rappel (Recall). Le rappel estime combien d'occurrences d'une classe ont été correctement prédites parmi toutes les occurrences réelles de cette classe. Optimiser un modèle selon cette métrique signifie que l'on cherche à identifier toutes les instances d'une classe, même au risque d'avoir des fausses détections. Cette métrique est principalement utilisée dans les problèmes de détection d'anomalies et de fraude.

Score F1. Cette métrique correspond à la moyenne harmonique du rappel et de la précision, ce qui permet d'équilibrer les classes sur- ou sous-représentées. Elle est utilisée lorsque l'on souhaite minimiser à la fois les faux positifs et les faux négatifs.

Comme nous l'avons mentionné pour la métrique de rappel, elle est surtout utilisée pour la détection d'anomalies. Dans le contexte de la comptabilité financière, cela permet de réduire les risques liés aux erreurs ou irrégularités financières non détectées. Dans ce travail, nous avons utilisé la moyenne macro du rappel pour ajuster et évaluer les modèles.

La Figure 3.7 illustre quatre cas distincts de prédiction binaire, montrant la construction des matrices de confusion et le calcul des métriques d'évaluation correspondantes. La métrique de support indique le nombre réel d'occurrences par classe.

		Case 1			Case 2			Case 3			Case 4		
Confusion matrices		**0**	**1**	**Supp.**	**0**	**1**	**Supp.**	**0**	**1**	**Supp.**	**0**	**1**	**Supp.**
Non-anomaly	**0**	*3731*	86	3817	*3731*	86	3817	*3317*	500	3817	*3817*	0	3817
Anomaly	**1**	2	*19*	21	0	*21*	21	2	*19*	21	3	*18*	21
Metrics		**Mic.**	**Mac.**	**W.**	**Mic.**	**Mac.**	**W.**	**Mic.**	**Mac.**	**W.**	**Mic.**	**Mac.**	**W.**
Recall(1)/Specificity		0.90	0.94	0.98	1.00	***0.99***	0.98	0.90	0.89	0.87	0.86	0.93	***1.00***
Recall(0)/Sensitivity		0.98			0.98			0.87			1.00		
Precision(1)		0.18	0.59	0.99	0.20	0.60	***1.00***	0.04	0.52	0.99	1.00	***1.00***	***1.00***
Precision(0)		1.00			1.00			1.00			1.00		
F1-score(1)		0.30	0.64	0.98	0.33	0.66	0.98	0.07	0.50	0.92	0.92	***0.96***	***1.00***
F1-score(0)		0.99			0.99			0.93			1.00		

FIGURE 3.7 – Calcul des métriques pour quatre cas de prédiction de modèle.

Comme on peut le voir sur cette figure, le **Cas 2** est le plus approprié pour la détection des écritures comptables anormales, car il permet de couvrir **toutes les anomalies avec un minimum de faux positifs**. Dans la Figure 3.7, nous pouvons observer que d'autres métriques peuvent être trompeuses compte tenu de la nature de notre problème. Nous avons utilisé la **moyenne macro de la spécificité et de la sensibilité** pour ajuster et évaluer nos modèles d'apprentissage automatique. Dans la suite de ce travail, nous faisons référence à cette métrique sous le nom de *recall average macro*.

3.4.3 Supervised Machine Learning Modeling

Modèle de régression logistique

La régression logistique modélise la probabilité qu'une transaction soit absente dans les fichiers JDE en fonction de ses caractéristiques, fournissant une interprétation probabiliste des anomalies [9].

Le principe de la régression logistique repose sur le logarithme naturel du rapport des cotes (*logit*), permettant de calculer des probabilités en étudiant la relation entre les variables. Compte tenu de la nature probabiliste de cet algorithme, il constitue un **choix pertinent pour la prédiction de classes** (variables catégorielles) à partir de **caractéristiques uniques ou multiples**, qu'elles soient **catégorielles ou continues**.

La probabilité qu'une variable de sortie y soit égale à 1, donnée une variable d'entrée x, est définie par l'équation suivante :

$$P(y = 1 \mid x) = \frac{e^{a+bx}}{1 + e^{a+bx}} = \frac{1}{1 + e^{-(a+bx)}} \tag{3.1}$$

où a représente le biais (ou intercept) et b le coefficient associé à la variable d'entrée x. Cette fonction logistique transforme toute valeur réelle en une probabilité comprise entre 0 et 1, ce qui en fait un outil adapté à la classification binaire.

Les paramètres du modèle, a et b, correspondent aux **pentes de la fonction logistique**. Pour gérer un plus grand nombre de variables, une **formule générale** peut être dérivée.

Pour implémenter le modèle de régression logistique, nous avons utilisé la classe `LogisticRegression` de la bibliothèque Python `scikit-learn`. Cette classe propose différents solveurs pouvant être spécifiés comme **hyperparamètre**. Le modèle de régression logistique a obtenu une performance satisfaisante sur la détection des transactions manquantes. La valeur du **rappel** pour la classe minoritaire "Manquante" est de 0,80, ce qui montre que le modèle réussit à identifier une partie significative des transactions absentes. Le **rapport complet de classification** pour ce modèle est présenté dans la **Table 3.8**.

TABLE 3.8 – Rapport de classification du modèle de régression logistique.

Classe	**Précision**	**Rappel**	**F1-Score**	**Support**
Présente (0)	0,97	0,67	0,79	29 034
Manquante (1)	0,19	0,80	0,31	2 815
Exactitude (Accuracy)			0,68	31 849
Moyenne macro	0,58	0,73	0,55	31 849
Moyenne pondérée	0,90	0,68	0,75	31 849

Conclusion : Le modèle de régression logistique prédit correctement la majorité des transactions "Présente", mais peine à obtenir une bonne précision sur la classe minoritaire "Manquante". Cependant, le rappel relativement élevé (0,80) pour la classe minoritaire indique que le modèle identifie une proportion significative des transactions manquantes, ce qui est important pour l'objectif principal de détection des anomalies dans le dataset.

Modèle des Machines à Vecteurs de Support (SVM)

Les machines à vecteurs de support (SVM) utilisent un hyperplan pour séparer les classes de transactions et sont particulièrement efficaces pour les données multivariées [4].

L'application réussie des *Support Vector Machines* (SVM) dans un large éventail de cas de classification a fait de cet algorithme l'un des plus performants et les plus utilisés. Les SVM cherchent à déterminer un ou plusieurs hyperplans optimaux permettant de séparer au mieux les instances de données dans un espace de dimension N, en maximisant la marge fonctionnelle, qui constitue la mesure de qualité de la séparation.

Les vecteurs de support doivent être aussi éloignés que possible de l'hyperplan afin d'obtenir le meilleur pouvoir de classification. La fonction de décision d'un SVM est définie comme suit :

$$F(x) = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b$$

où x_i représente les échantillons d'entraînement, b est le biais, y_i la sortie désirée, α_i un facteur de pondération, et $k(x_i, x)$ le noyau (*kernel*) associé à chaque vecteur de support x_i.

Dans ce travail, nous avons utilisé le classificateur `LinearSVC` calibré de la bibliothèque Python `scikit-learn`. Ce classificateur permet d'obtenir des probabilités calibrées et d'ajuster le seuil de décision pour mieux détecter la classe minoritaire. Le modèle SVM optimisé a obtenu une valeur moyenne de rappel (*recall*) macro de 0.63, ce qui reste supérieur au rappel de la classe minoritaire avec le modèle de régression logistique. Le rapport complet de classification du modèle SVM est présenté dans la **Table 3.9**.

TABLE 3.9 – Rapport de classification du modèle SVM.

Classe	**Précision**	**Rappel**	**F1-Score**	**Support**
Présente (0)	0.94	0.77	0.85	29034
Manquante (1)	0.17	0.49	0.26	2815
Exactitude (Accuracy)			0.75	31849
Moyenne macro	0.56	0.63	0.55	31849
Moyenne pondérée	0.87	0.75	0.80	31849

Conclusion : Le modèle SVM prédit très bien la classe majoritaire "Présente" avec un rappel de 0.77. Cependant, la classe minoritaire "Manquante" reste difficile à détecter (rappel = 0.49), ce qui montre l'influence du déséquilibre du dataset. L'accuracy globale est de 0.75, mais cette métrique est fortement influencée par la classe majoritaire.

Modèle d'Arbre de Décision (Decision Tree)

L'arbre de décision segmente les données en règles hiérarchiques, permettant de détecter facilement quelles transactions présentent des anomalies en suivant le chemin des décisions [6].

Les *arbres de décision* sont des classificateurs qui génèrent automatiquement des règles à partir des données d'apprentissage en parcourant le jeu de données. La sortie d'un arbre de classification est de nature catégorielle. Les arbres de décision peuvent utiliser à la fois des attributs catégoriels et numériques dans leur processus de construction.

Le nœud racine de la structure d'un arbre de décision constitue le point de départ à partir duquel les nœuds intérieurs (ou de décision) sont générés. Chaque nœud non-feuille représente un attribut dont la division dépend d'une valeur de séparation (*split value*). Les tests sur les attributs et le processus de descente dans l'arbre se poursuivent jusqu'à atteindre les nœuds feuilles.

L'attribut le plus informatif est privilégié lors du découpage selon la mesure d'entropie, définie par la formule suivante, où P_i est la proportion d'exemples appartenant à la $i^{\text{ème}}$ classe :

$$Entropy = -\sum_{i=1}^{C} P_i \log_2 P_i$$

Plus la valeur de l'entropie est proche de 0, plus le jeu de données est impur. À l'inverse, le *gain d'information* (information gain), défini comme la réduction de l'entropie, permet de déterminer la qualité d'un découpage. Il est donné par la formule suivante :

$$Gain(A) = \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

où A est un attribut, $V(A)$ l'ensemble de ses valeurs possibles, et S_v le sous-ensemble de S contenant les échantillons pour lesquels l'attribut A prend la valeur v.

La division de l'arbre est arrêtée lorsqu'aucune amélioration supplémentaire ne peut être apportée à la prédiction.

Dans ce travail, nous avons implémenté le modèle d'arbre de décision à l'aide de la classe `DecisionTreeClassifier` de la bibliothèque Python `scikit-learn`. Le modèle d'arbre de décision optimisé a obtenu de bonnes performances sur le jeu de test.

Le rapport complet de classification du modèle est présenté dans la Table 3.10.

TABLE 3.10 – Rapport de classification du modèle Decision Tree.

Classe	Précision	Rappel	F1-score	Support
Présente (0)	0.98	0.98	0.98	29020
Manquante (1)	0.82	0.82	0.82	2829
Accuracy			0.97	31849
Macro avg	0.90	0.90	0.90	31849
Weighted avg	0.97	0.97	0.97	31849

Conclusion Le modèle Decision Tree atteint une accuracy globale élevée (0.97) et montre de bonnes performances sur les deux classes. La classe majoritaire "Présente"

est très bien prédite avec un rappel de 0.98, tandis que la classe minoritaire "Manquante" obtient un rappel de 0.82, ce qui est nettement meilleur que les modèles de régression logistique ou SVM sur ce dataset. Ces résultats indiquent que le modèle est capable de détecter la majorité des transactions manquantes tout en maintenant une précision élevée pour la classe majoritaire.

Modèle de Forêt Aléatoire (Random Forest)

Le Random Forest est un algorithme supervisé basé sur un ensemble d'arbres de décision qui permet de classifier les transactions et de détecter les anomalies avec une grande précision [2].

Afin de remédier au problème de surapprentissage (*overfitting*) rencontré avec les arbres de décision et d'améliorer leurs performances, des algorithmes fondés sur des ensembles d'arbres de décision (*ensemble learning*) ont été introduits. Parmi eux, la *forêt aléatoire* (*Random Forest*) est l'un des modèles les plus précis et les plus largement utilisés.

Le principe des classificateurs de type Random Forest consiste à construire une multitude d'arbres de décision, puis à combiner leurs prédictions à l'aide d'un mécanisme de vote majoritaire. La randomisation du modèle provient de deux sources principales :

— Le *bagging* (bootstrap aggregating), qui permet de générer différents sous-ensembles de données d'entraînement pour construire des arbres variés.
— La sélection aléatoire de sous-ensembles de caractéristiques (*features subsetting*) lors de la création de chaque arbre.

Ces deux types de randomisation rendent les arbres plus indépendants entre eux et permettent ainsi de consolider plusieurs *apprenants faibles* (*weak learners*) en un modèle globalement plus robuste et performant.

Le modèle de forêt aléatoire optimisé a obtenu une valeur moyenne de rappel (*recall*) macro de 0.89 et une précision pondérée de 0.97, surpassant ainsi tous les modèles précédemment entraînés. Le rapport complet de classification du modèle de forêt aléatoire est présenté dans la **Table 3.11**.

Table 3.11 – Rapport de classification du modèle de Forêt Aléatoire.

Classe	**Précision**	**Rappel**	**F1-Score**	**Support**
Présente (0)	0.98	0.99	0.99	29 020
Manquante (1)	0.93	0.78	0.85	2 829
Exactitude (Accuracy)			0.98	31 849
Moyenne macro	0.95	0.89	0.92	31 849
Moyenne pondérée	0.97	0.98	0.97	31 849

Conclusion : Le modèle Random Forest prédit très efficacement la classe majoritaire "Présente" avec un rappel proche de 1 et détecte correctement la classe minoritaire "Manquante" avec un rappel de 0.78. L'accuracy globale élevée (0.98) montre la robustesse du modèle face au déséquilibre des classes et sa capacité à généraliser sur l'ensemble de test.

Modèle des k Plus Proches Voisins (K-Nearest Neighbors, KNN)

Le kNN classe chaque transaction en fonction des transactions les plus proches dans l'espace des caractéristiques, ce qui en fait un modèle simple mais robuste pour la détection d'anomalies [3].

Les classificateurs des *k plus proches voisins* (*K-Nearest Neighbors*, KNN) reposent sur l'attribution d'une instance de données à une classe en fonction de la proximité avec les points de données les plus proches appartenant à cette classe.

L'une des distances les plus couramment utilisées est la distance euclidienne, définie comme suit :

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

où x et y sont deux points de données dans un espace de dimension n. La distance est obtenue en prenant la racine carrée de la somme des carrés des différences entre les composantes correspondantes x_i et y_i.

Le paramètre k de l'algorithme représente le nombre de voisins les plus proches à considérer pour déterminer la classe d'un point. Dans le cas de plusieurs classes, un mécanisme de vote majoritaire est utilisé pour choisir la classe prédite. En cas d'égalité entre classes, le choix peut être effectué de manière aléatoire ou pondéré selon les distances.

Une normalisation des caractéristiques (*feature scaling*) est nécessaire afin d'assurer une contribution proportionnelle de chaque variable dans le calcul des distances. Le paramètre k est un hyperparamètre manuel du modèle, dont la valeur optimale est obtenue par itération sur un ensemble de valeurs possibles.

L'unique hyperparamètre à ajuster dans ce modèle est donc le nombre de voisins k. Dans ce travail, nous avons testé plusieurs valeurs comprises entre 1 et 30 afin d'optimiser les performances du modèle KNN.

Le modèle KNN optimisé a obtenu une valeur moyenne de rappel (*recall*) macro de 0.84. Le rapport complet de classification du modèle KNN est présenté dans la **Table 3.12**.

TABLE 3.12 – Rapport de classification du modèle KNN.

Classe	**Précision**	**Rappel**	**F1-Score**	**Support**
Présente (0)	0.97	0.97	0.97	29034
Manquante (1)	0.72	0.71	0.71	2815
Exactitude (Accuracy)			0.95	31849
Moyenne macro	0.84	0.84	0.84	31849
Moyenne pondérée	0.95	0.95	0.95	31849

Conclusion Les modèles KNN ont tendance à surapprendre (*overfit*) lorsque le nombre de voisins k est faible. Contrairement aux autres modèles étudiés, il n'est pas possible d'intégrer des poids de classe dans le KNN, ce qui rend ce dernier sensible au déséquilibre entre classes. Cependant, dans ce cas, le modèle a montré de bonnes performances pour les deux classes, avec un *recall* relativement équilibré (0.97 pour la classe majoritaire et 0.71 pour la classe minoritaire). Pour des améliorations futures, il serait pertinent d'explorer un

KNN entraîné sur un jeu de données équilibré ou d'utiliser des techniques de pondération des distances afin d'améliorer la détection de la classe minoritaire.

3.4.4 Supervised Models Evaluation

Après avoir entraîné et testé différents modèles supervisés, une comparaison globale de leurs performances a été réalisée. Cette évaluation repose principalement sur le rappel (*recall*) pour la classe minoritaire "Manquante", ainsi que sur les mesures de classification standard pour la classe majoritaire "Présente". L'objectif est de déterminer quel modèle est le plus efficace pour détecter les transactions manquantes tout en maintenant une bonne précision globale.

La **Table 3.13** présente les métriques clés de performance pour tous les modèles supervisés étudiés, incluant le nombre de vrais négatifs (TN), faux négatifs (FN), faux positifs (FP), vrais positifs (TP), ainsi que la moyenne du rappel macro (*Recall Avg Macro*) pour chaque modèle.

TABLE 3.13 – Comparaison des performances des modèles supervisés appliqués à la détection des transactions manquantes.

No.	Algorithme	TN	FN	FP	TP	Recall Avg Macro
1	Logistic Regression	19453	9581	563	2252	0.735
2	Support Vector Machines	22364	6670	1463	1379	0.63
3	**Decision Tree**	**28440**	**580**	**509**	**2320**	**0.90**
4	**Random Forest**	**28730**	**290**	**622**	**2207**	**0.885**
5	K-Nearest Neighbour	28163	871	817	1998	0.84

Analyse : Le tableau montre que les modèles basés sur des arbres, en particulier le *Decision Tree* et le *Random Forest*, obtiennent les meilleures performances en termes de rappel moyen macro, respectivement 0,90 et 0,885, ce qui indique leur efficacité pour détecter les transactions manquantes.

Le *K-Nearest Neighbour* atteint un rappel moyen de 0,84, ce qui est correct mais inférieur aux modèles d'arbre, montrant une certaine sensibilité aux déséquilibres de classes. Les modèles *Logistic Regression* et *Support Vector Machines* présentent des performances plus modestes (rappel moyen de 0,735 et 0,63), indiquant que ces modèles détectent moins efficacement la classe minoritaire, malgré une bonne précision sur la classe majoritaire.

En conclusion, pour ce problème de détection des transactions manquantes, les modèles d'ensemble basés sur les arbres (*Decision Tree* et *Random Forest*) se révèlent les plus robustes et fiables, tandis que les modèles probabilistes ou basés sur la distance peuvent nécessiter des techniques supplémentaires, comme le rééquilibrage des classes ou l'ajustement des seuils de décision, pour améliorer la détection des anomalies.

3.4.5 Déploiement

Dans cette étude, l'objectif principal était de détecter les transactions présentes dans le fichier *Afriware* et absentes dans les fichiers *JDE*. L'étape de déploiement permet d'examiner les implications pratiques de cette approche et la manière dont elle peut transformer le processus d'audit.

Risque d'échantillonnage. Les audits manuels des transactions nécessitent souvent un échantillonnage aléatoire des données, ce qui peut laisser passer certaines anomalies. Grâce à notre approche supervisée, basée sur les modèles d'apprentissage automatique, nous sommes capables de détecter directement les transactions manquantes dans les fichiers *JDE* sans avoir à échantillonner l'ensemble du jeu de données. Le modèle attribue à chaque transaction un score ou une prédiction indiquant si elle est absente dans *JDE*, ce qui facilite la détection systématique et rapide.

Complexité des patterns. La diversité des transactions et des formats de fichiers rend la détection manuelle complexe. Certaines transactions peuvent être présentes dans *Afriware* sous des formats légèrement différents ou regroupées différemment dans les fichiers *JDE*. En entraînant des modèles supervisés sur des transactions labellisées (présentes ou absentes), nous avons pu apprendre efficacement les patterns de correspondance et détecter automatiquement les transactions manquantes, réduisant ainsi le risque d'erreur humaine.

Volume de données et efficacité temporelle. Les volumes de transactions peuvent être très importants, rendant les vérifications manuelles longues et coûteuses. Avec le modèle supervisé entraîné, il est possible de traiter de grands fichiers *Afriware* et *JDE* rapidement, en générant instantanément la liste des transactions absentes dans *JDE*. Cela permet de réduire significativement le temps nécessaire pour les audits et d'améliorer la réactivité des contrôles.

Rapport de déviation. Les résultats peuvent être présentés sous forme de rapport listant toutes les transactions détectées comme manquantes dans *JDE*, accompagnées d'indicateurs de risque. Cela facilite le suivi et le traitement des anomalies par les auditeurs. Chaque transaction peut être marquée avec un indicateur binaire (1 si absente dans *JDE*, 0 sinon), ce qui permet de filtrer rapidement les cas critiques pour un audit plus efficace.

3.5 Discussions

Dans cette étude, nous avons appliqué plusieurs modèles supervisés afin de détecter les transactions présentes dans le fichier *Afriware* mais absentes dans les fichiers *JDE*. L'analyse des résultats obtenus permet de comprendre la performance de chaque modèle ainsi que les implications pratiques pour le processus d'audit.

Analyse comparative des modèles

Les différents modèles supervisés utilisés présentent des caractéristiques et performances variées :

- **Modèles linéaires** : simples à interpréter et rapides à exécuter, mais limités lorsqu'il existe des patterns complexes ou non linéaires entre les données.
- **Arbres de décision et forêts aléatoires** : capables de capturer efficacement les interactions complexes entre les variables, offrant de bons résultats pour identifier les transactions manquantes.

— **KNN (K-Nearest Neighbors)** : performants pour certaines correspondances simples, mais sensibles aux variations dans les formats de données et aux transactions atypiques.
— **SVM (Support Vector Machines)** : robustes pour des séparations nettes entre classes (présente/absente), mais nécessitent un réglage fin des hyperparamètres et une bonne normalisation des données.

Limites de l'étude

Malgré des résultats satisfaisants, certaines limites doivent être soulignées :
— **Qualité et homogénéité des données** : les différences de format entre les fichiers *Afriware* et *JDE* peuvent entraîner des erreurs de correspondance ou des faux positifs.
— **Déséquilibre des classes** : le nombre de transactions absentes dans *JDE* est généralement très faible comparé aux transactions présentes, ce qui peut biaiser certains modèles.
— **Optimisation des hyperparamètres** : certains modèles nécessitent un ajustement précis pour maximiser la détection des transactions manquantes.

Implications pratiques

Les modèles supervisés permettent d'automatiser la détection des transactions manquantes, ce qui offre plusieurs avantages pour les auditeurs :
— Réduction du temps et des coûts liés à la vérification manuelle des transactions.
— Détection systématique et fiable des transactions absentes, limitant le risque d'erreur humaine.
— Possibilité de générer des rapports détaillés, indiquant quelles transactions nécessitent un contrôle prioritaire.

3.6 Conclusion

Dans ce travail, nous avons principalement étudié et appliqué des modèles supervisés pour détecter les transactions présentes dans le fichier *Afriware* mais absentes dans les fichiers *JDE*. Les modèles supervisés se sont révélés efficaces pour identifier ces anomalies, permettant une détection systématique et rapide, réduisant ainsi le risque d'erreur humaine et le temps nécessaire aux audits manuels.

Nous avons également tenté d'implémenter des modèles non supervisés pour la détection des anomalies. Cependant, ceux-ci se sont montrés peu pertinents dans le contexte de notre étude, notamment à cause du volume important de données et du faible nombre de transactions réellement anormales, ce qui rend leur apprentissage et leur performance moins fiables.

3.7 Perspectives d'amélioration et travaux futurs

Bien que les modèles supervisés aient fourni des résultats satisfaisants pour la détection des transactions présentes dans *Afriware* et absentes dans *JDE*, plusieurs pistes d'amélioration sont possibles pour rendre le système plus robuste et étendu :

- **Optimisation des modèles non supervisés :** Nous avons tenté d'appliquer des modèles non supervisés pour détecter les anomalies. Cependant, ils se sont révélés peu performants à cause du faible ratio d'anomalies et du volume important de données. Il serait possible d'optimiser ces modèles en ajustant le seuil de détection, en utilisant des méthodes de réduction de dimension ou en combinant plusieurs modèles pour améliorer la précision.
- **Réseaux de neurones et Deep Learning :** L'utilisation de réseaux de neurones, y compris des architectures profondes, pourrait permettre d'apprendre des patterns complexes dans les transactions et de détecter des anomalies plus subtiles ou non évidentes, qu'un modèle classique pourrait manquer.
- **Détection d'autres types d'anomalies :** Le framework peut être étendu pour détecter différents types d'incohérences et fraudes, par exemple :
  - Fraudes liées à la TVA ou à des déclarations fiscales inexactes.
  - Doublons ou omissions de transactions.
  - Manipulations dans les écritures comptables (transferts fictifs, montants incorrects, etc.).
- **Ensemble learning et combinaisons de modèles :** La combinaison de modèles supervisés et non supervisés peut améliorer la robustesse globale, réduire les faux positifs et détecter à la fois les anomalies connues et inconnues.
- **Amélioration de la qualité des données :** La standardisation et la normalisation des fichiers *Afriware* et *JDE* permettraient de réduire les erreurs de correspondance et d'améliorer la précision du modèle.

Ces perspectives ouvrent la voie à un système d'audit automatisé plus complet et capable de détecter efficacement une variété de fraudes et d'incohérences dans les données financières.

Conclusion Générale

Cette étude a démontré que l'application de modèles supervisés pour la détection de transactions manquantes est efficace et constitue une avancée dans l'audit automatisé. Les travaux futurs pourraient inclure l'utilisation de réseaux neuronaux et l'extension à d'autres types d'anomalies financières [2, 10, 7].

Ce travail a porté sur la détection automatique des transactions présentes dans le fichier *Afriware* et absentes dans les fichiers *JDE*, en utilisant principalement des modèles supervisés. Les résultats obtenus ont montré que ces modèles permettent une identification fiable et rapide des anomalies, réduisant le temps et les coûts liés aux vérifications manuelles et limitant le risque d'erreurs humaines.

L'étude a également mis en évidence les limites des modèles non supervisés dans ce contexte, notamment à cause du faible ratio d'anomalies et du volume important de données. Néanmoins, ces modèles pourraient être optimisés pour devenir plus pertinents dans de futurs travaux.

Durant ce stage, j'ai pu maîtriser plusieurs compétences clés dans le domaine de l'ingénierie en intelligence artificielle, notamment :

- la conception et l'implémentation de modèles d'apprentissage supervisé tels que Random Forest, Decision Tree, SVM, kNN et Logistic Regression ;
- l'application de techniques d'apprentissage non supervisé, incluant Isolation Forest et Autoencoder, pour l'analyse de grandes quantités de données ;
- le prétraitement des données et l'ingénierie des features pour améliorer la performance des modèles ;
- l'évaluation et la comparaison des modèles à l'aide de métriques classiques telles que précision, rappel et F1-score ;
- l'utilisation d'outils et frameworks de machine learning pour le traitement de données financières complexes.

Enfin, cette étude ouvre plusieurs perspectives pour le développement de systèmes d'audit automatisés plus avancés, notamment l'utilisation des réseaux de neurones, l'extension à d'autres types d'anomalies (fraudes fiscales, doublons, manipulations comptables), ainsi que l'application de techniques d'ensembling pour améliorer la robustesse et la précision des modèles.

En résumé, le travail réalisé constitue une base solide pour la mise en place d'outils de détection d'anomalies dans les données financières, illustre le potentiel du machine learning pour transformer les pratiques d'audit, et témoigne de la maîtrise des compétences techniques et analytiques acquises dans le cadre de ma formation en ingénierie en intelligence artificielle.

Bibliographie

[1] Bart Baesens, V. Van Vlasselaer, and W. Verbeke. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques : A Guide to Data Science for Fraud Detection*. Wiley, New York, NY, USA, 2015.

[2] L. Breiman. Random forests. *Machine Learning*, 45 :5–32, 2001.

[3] P. Cunningham and S.J. Delany. k-nearest neighbour classifiers : 2nd edition (with python examples). *arXiv*, arXiv :2004.04523, 2020.

[4] T. Evgeniou and M. Pontil. Support vector machines : Theory and applications. *Lecture Notes in Machine Learning*, 2049 :249–257, 2001.

[5] A. Gholamy, V. Kreinovich, and O. Kosheleva. Why 70/30 or 80/20 relation between training and testing sets : A pedagogical explanation. Technical Report UTEP-CS-18-09, University of Texas at El Paso, 2018. Accessed on 19 April 2022.

[6] B.T. Jijo and A.M. Abdulazeez. Classification based on decision tree algorithm for machine learning. In *J. Appl. Sci. Technol. Trends*, volume 2, pages 20–28, 2021.

[7] J. Lahann, M. Scheid, and P. Fettke. Utilizing machine learning techniques to reveal vat compliance violations in accounting data. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, pages 1–10, Moscow, Russia, 2019.

[8] J. Nonnenmacher and J.M. Gómez. Unsupervised anomaly detection for internal auditing : Literature review and research agenda. *Int. J. Digit. Account. Res.*, 21 :1–22, 2021.

[9] C.Y.J. Peng, K.L. Lee, and G.M. Ingersoll. An introduction to logistic regression analysis and reporting. *J. Educ. Res.*, 96 :3–14, 2002.

[10] M. Schreyer, T. Sattarov, C. Schulze, B. Reimer, and D. Borth. Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks. In *2nd KDD Workshop on Anomaly Detection in Finance*, pages 1–10, Anchorage, AK, USA, 2019.

[11] A. Zemankova. Artificial intelligence in audit and accounting : Development, current trends, opportunities and threats - literature review. In *Proceedings of the 2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 148–154, Athens, Greece, 2019.