FSBM
Faculté des sciences ben m'sick
Université Hassan II de Casalanca

جـامعة الحسن الثـاني بالدار البيضاء
UNIVERSITÉ HASSAN II DE CASABLANCA

Data Science & Big Data
Master DSBD

RAPPORT DE PROJET DE FIN D'ANNÉE

DÉTECTION DE FRAUDE DANS LES TRANSACTIONS FINANCIÈRES

EN UTILISANT DES TECHNIQUES DE DEEP LEARNING ET D'APPRENTISSAGE AUTOMATIQUE

Réalisé par :

EL HAFIDI OUSSAMA

Encadré par :

Mme Ounacer Soumaya Professeur à FSBM Encadrante

Mr Soufiane ARDCHIR Professeur à FSBM Examinateur

Année universitaire 2023 – 2024

Remerciement

Je tiens à dédier ce rapport à l'ensemble de mes enseignants et encadrants en particulier ***Madame OUNACER SOUMAYA*** *et* ***Monsieur Soufiane ARDCHIR,*** *dont les connaissances et l'engagement ont grandement contribué à l'élaboration de ce projet. Leur expertise et leur soutien constant ont été une source d'inspiration tout au long de mon parcours.*

Résumé :

Ce projet vise à développer des modèles de détection de fraude dans les transactions financières en utilisant des techniques d'apprentissage automatique et de deep learning. À partir d'un ensemble de données de transactions, incluant des informations sur les montants, les commerçants et les caractéristiques des clients, nous avons réalisé une analyse exploratoire pour identifier les valeurs aberrantes et les variables pertinentes. Nous avons ensuite appliqué diverses techniques de classification, y compris la régression logistique, les arbres de décision, les forêts aléatoires, K-Nearest Neighbors, et les réseaux de neurones. La méthode SMOTE a été utilisée pour traiter le déséquilibre des classes. Les performances des modèles ont été évaluées à l'aide de métriques telles que l'exactitude, le rappel, le F1-score, et le Coefficient de Corrélation de Matthews (MCC). Les résultats montrent que l'utilisation de méthodes avancées de traitement des données et de modèles de deep learning permet d'améliorer significativement la détection des fraudes, offrant ainsi des outils prometteurs pour la sécurisation des transactions financières.

Abstract

This project aims to develop fraud detection models for financial transactions using machine learning and deep learning techniques. Utilizing a dataset of transactions that includes information on amounts, merchants, and customer characteristics, we performed exploratory data analysis to identify outliers and relevant variables. Various classification techniques, including logistic regression, decision trees, random forests, K-Nearest Neighbors, and neural networks, were applied. The SMOTE method was employed to address class imbalance. Model performance was evaluated using metrics such as accuracy, recall, F1-score, and the Matthews Correlation Coefficient (MCC). Results indicate that advanced data processing methods and deep learning models significantly enhance fraud detection capabilities, providing promising tools for securing financial transactions.

Liste de Tableaux

Table de Matières

Introduction générale :

LA FRAUDE DANS LES TRANSACTIONS FINANCIÈRES REPRÉSENTE UN ENJEU MAJEUR POUR LES INSTITUTIONS BANCAIRES ET LES ENTREPRISES DE SERVICES FINANCIERS À L'ÉCHELLE MONDIALE. AVEC L'ESSOR DES TECHNOLOGIES NUMÉRIQUES ET L'AUGMENTATION DES PAIEMENTS EN LIGNE, LES MÉTHODES DE FRAUDE SE DIVERSIFIENT ET SE COMPLEXIFIENT, RENDANT LEUR DÉTECTION DE PLUS EN PLUS DIFFICILE. EN 2023, LES PERTES FINANCIÈRES DUES À LA FRAUDE ONT ATTEINT DES MILLIARDS DE DOLLARS, INCITANT LES ORGANISATIONS À INVESTIR DANS DES SYSTÈMES DE DÉTECTION ROBUSTES ET EFFICACES.

DANS CE CONTEXTE, LES TECHNIQUES D'APPRENTISSAGE AUTOMATIQUE ET DE DEEP LEARNING SE RÉVÈLENT ÊTRE DES OUTILS PUISSANTS POUR IDENTIFIER ET PRÉVENIR LES FRAUDES. CES MÉTHODES PERMETTENT D'ANALYSER D'ÉNORMES VOLUMES DE DONNÉES EN TEMPS RÉEL, D'IDENTIFIER DES MODÈLES COMPORTEMENTAUX SUSPECTS ET DE DÉTECTER DES ANOMALIES QUI POURRAIENT INDIQUER UNE ACTIVITÉ FRAUDULEUSE. CEPENDANT, L'EFFICACITÉ DE CES MODÈLES DÉPEND LARGEMENT DE LA QUALITÉ DES DONNÉES, DE LA SÉLECTION DES CARACTÉRISTIQUES PERTINENTES ET DE LA GESTION DES DÉSÉQUILIBRES ENTRE LES CLASSES.

CE PROJET DE FIN D'ANNÉE VISE À DÉVELOPPER ET À ÉVALUER PLUSIEURS MODÈLES DE DÉTECTION DE FRAUDE À PARTIR D'UN ENSEMBLE DE DONNÉES DE TRANSACTIONS. NOUS COMMENCERONS PAR UNE ANALYSE EXPLORATOIRE POUR MIEUX COMPRENDRE LES DONNÉES ET IDENTIFIER LES VARIABLES CLÉS. ENSUITE, NOUS APPLIQUERONS DIVERSES TECHNIQUES DE CLASSIFICATION, EN INTÉGRANT DES MÉTHODES DE SURÉCHANTILLONNAGE TELLES QUE SMOTE POUR TRAITER LE DÉSÉQUILIBRE DES CLASSES. LES PERFORMANCES DES MODÈLES SERONT ÉVALUÉES À L'AIDE DE PLUSIEURS MÉTRIQUES, PERMETTANT AINSI DE DÉTERMINER LES APPROCHES LES PLUS EFFICACES POUR LA DÉTECTION DE FRAUDE.

À TRAVERS CE TRAVAIL, NOUS ESPÉRONS APPORTER DES SOLUTIONS CONCRÈTES POUR AMÉLIORER LA SÉCURITÉ DES TRANSACTIONS FINANCIÈRES ET CONTRIBUER À LA LUTTE CONTRE LA FRAUDE, UN DÉFI CRUCIAL DANS LE PAYSAGE ÉCONOMIQUE ACTUEL.

CHAPITRE 1

CONTEXTE GÉNÉRALE

CHAPITRE 1 : Contexte générale

Introduction

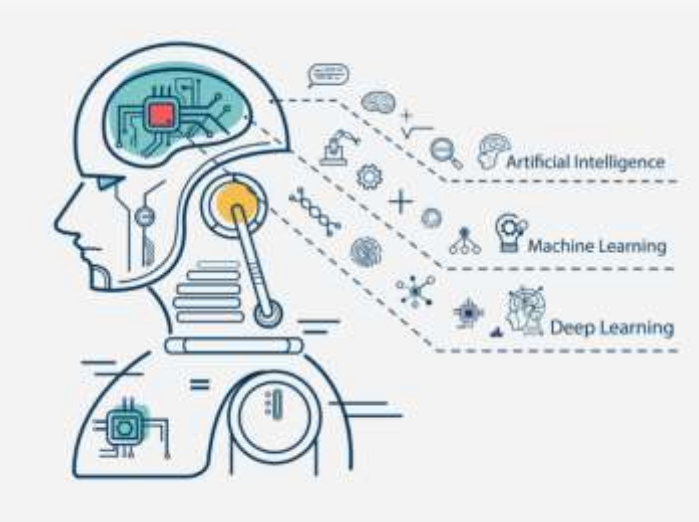
Ce chapitre introduit le contexte général dans lequel le projet s'intègre, en présentant à la fois son thème principal, la problématique générale et les objectifs du projet, ainsi que la conduite du projet. Identifier toutes les fonctionnalités de notre futur système, et ceci en recensant les besoins fonctionnels.

1 . Importance de la détection de la fraude dans les transactions financières

Dans un monde de plus en plus digitalisé, les transactions financières en ligne et via les plateformes numériques sont devenues omniprésentes. Avec cette évolution, le risque de fraude financière a considérablement augmenté, touchant les institutions financières, les entreprises et les consommateurs. La détection de la fraude est devenue cruciale pour minimiser les pertes économiques, protéger les utilisateurs et maintenir la confiance dans les systèmes de paiement.

2 . Objectifs du Projet

Ce projet a pour but de développer un modèle de détection de fraude efficace en utilisant des techniques de Machine Learning et de Deep Learning. L'objectif principal est d'améliorer les capacités de détection dans des situations où les données sont fortement déséquilibrées, c'est-à-dire lorsqu'il y a beaucoup plus de transactions légitimes que de transactions frauduleuses. Le projet se concentre également sur l'évaluation de l'impact de la technique SMOTE (Synthetic Minority Over-sampling Technique) pour le rééquilibrage des classes.



3 . Contribution du Projet

Le projet propose une approche comparative entre plusieurs algorithmes de Machine Learning (régression logistique, arbres de décision, KNN, etc.) et des modèles de Deep Learning (LSTM, DNN, RNN), avec et sans l'application de SMOTE. Cette étude permettra d'identifier les modèles les plus efficaces pour détecter la fraude dans un contexte de données déséquilibrées. Les résultats de ce projet pourraient être utilisés pour améliorer les systèmes de sécurité dans les institutions financières et réduire les pertes liées aux fraudes.

CHAPITRE 2

REVUE DE LITTÉRATURE

Chapitre 2 : Revue de Littérature

1 . Revue de la Littérature

La détection de fraude dans les transactions financières est un domaine de recherche crucial, notamment dans les paiements par carte de crédit. Diverses approches utilisant des techniques d'apprentissage automatique et de deep learning ont été proposées dans la littérature pour améliorer la détection des fraudes. Chaque étude apporte des perspectives nouvelles en termes de méthodologies, de performance, et d'applications sur des jeux de données spécifiques.

2 . Étude Comparative des Articles Similaires

Pour comparer les approches proposées par différents articles, nous avons analysé plusieurs dimensions clés : les algorithmes utilisés, les mesures de performance appliquées, les facteurs influençant la performance, les caractéristiques des jeux de données, ainsi que les avantages et limites de chaque étude. Le tableau ci-dessous résume les résultats de cette analyse comparative :

Article	Algorithmes d'apprentissage automatique	Mesures de performance			Description du jeu de données
Credit Card Fraud Detection Using Machine Learning	***K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Régression logistique***	**KNN**	99.88%		**Source :** Kaggle.com **Taille :** 284,808 Lignes **Caractéristique :** 31 attributs (28 num, 3 non transformés [Time, Amount. Class])
		RL	99.92%		
		SVM	99.94%		
Credit Card Fraud Detection Using Auto encoder	***- Auto encodeur / Oversampling***		**Recall**	**Acc**	**Source :** Kaggle.com **Taille :** 28,315 Lignes **Caractéristique :** 28 attributs (3 non transformés [Time, Amount. Class])
		Sans	0%	100%	
		Avec	84%	97.93%	

Credit Card Fraud Detection using Deep Learning Techniques	***- Réseau neuronal profond***	**Recall**		**Acc**		**Source :** date_frauda.csv **Taille :** 151,114 Lignes **Caractéristique :** 11 attributs
		100%		99.76%		
Fraud detection using deep learning	***- Deep Neural Networks***	**Réseau neuronal LSTM**	**Sans**	**ACC**	99.88%	**Source :** Kaggle.com **Taille :** 284,808 Lignes **Caractéristique :** 31 attributs (28 num, 3 non transformés [Time, Amount. Class])
				REC	99.88%	
				PRE	99.88%	
			Avec	**ACC**	99.84%	
				REC	72.05%	
				PRE	89.91%	
		Régression logistique		**ACC**	98.07%	
				REC	96.96%	
				PRE	99.17%	
Modified Focal Loss in Imbalanced XGBoost for Credit Card Fraud Detection	***- XGBoost avec perte focale modifiée***			**REC**	**ACC**	**Source :** Université Libre de Bruxelles (ULB) **Taille :** 284,807 Lignes **Caractéristique :** 30 attributs (2 non transformés [Time,Amount])
		SANS		0%	100%	
		AVEC		83.67%	96.98%	
ANALYSIS OF DATA ENGINEERING FOR FRAUD DETECTION USING deep LEARNING AND ARTIFICIAL INTELLIGENCE TECHNOLOGIES	***- Régression logistique.*** ***- Forêt aléatoire (RF).*** ***- Gradient boosting machine (GBM).***			**Rec**	**ACC**	Données synthétiques (1 000 000 d'échantillons)
		RL		0.6190	0.7647	
		RF		0.6190	0.8125	
		GBM		0.6667	0.8750	
Deep Learning for Credit Card Fraud Detection : A Review of Algorithms,	***MLP, LSTM, Simple RNN.***			F1-score	ACC	
		MLP		08016	0.9980	

Challenges, and Solutions		**LSTM**	0.7721	0.9994	Européen Crédit Card Dataset Contient 284,807 transactions de pays européens Comprend 28 caractéristiques.
		Simple RNN	0.8132	0.9994	

Tableau 1:Comparaison des Articles Scientifiques sur les Algorithmes ML & DL pour la Détection de Fraudes

3 . Synthèse des Méthodologies et Approches Utilisées dans les Travaux Précédents :

Les articles étudiés mettent en lumière différentes techniques pour améliorer la détection des fraudes dans les transactions financières. Voici un résumé des méthodologies et approches couramment utilisées :

Algorithmes d'apprentissage automatique classiques (Random Forest, Logistic Régression, SVM) sont efficaces pour les petits jeux de données, mais leur performance diminue lorsque les données sont fortement déséquilibrées ou complexes. Ils sont plus faciles à interpréter, mais nécessitent un bon équilibrage des classes et un ajustement fin des hyperparamètres.

Autoencoders et **Isolation Forest** sont utilisés pour détecter les anomalies sans étiquettes de classe explicites, en s'appuyant sur la reconstruction des données. Bien que cette approche fonctionne bien dans les environnements contrôlés, elle est sensible aux données bruitées et peut être difficile à ajuster correctement.

Deep Learning avec des architectures comme **DNN**, **CNN**, et **LSTM** offre des performances élevées, particulièrement dans les ensembles de données volumineux. Ces modèles ont l'avantage de capturer des relations complexes dans les données, notamment grâce à la modélisation des séquences temporelles (comme dans LSTM). Cependant, ils nécessitent une grande capacité de calcul et sont susceptibles de surapprentissage (overfitting) si les données ne sont pas suffisamment équilibrées.

XGBoost avec Focal Loss montre une grande promesse pour les ensembles de données déséquilibrés grâce à l'ajustement du poids des classes dans la fonction de perte. Cependant, cette méthode peut être plus difficile à ajuster en raison de la complexité accrue du tuning des hyperparamètres.

En conclusion, chaque approche présente des avantages et des limitations spécifiques, en fonction de la nature des données et des ressources disponibles. L'intégration de techniques de rééquilibrage des classes, comme le SMOTE ou des techniques de focal loss, combinées à des modèles robustes comme XGBoost ou des réseaux de neurones profonds, semble être une direction prometteuse pour améliorer la détection de fraude dans les transactions financières.

CHAPITRE 3

MÉTHODOLOGIE ET APPROCHE

Chapitre 3 : Méthodologie et Approche

1 . Présentation du Dataset

Le dataset utilisé pour cette analyse contient des transactions financières, et chaque transaction est classée comme frauduleuse ou non. Voici quelques informations essentielles :

- Le dataset d'entraînement contient **1,296,675** lignes et **23** colonnes, tandis que le dataset de test contient **555,719** lignes.
- Les colonnes incluent des informations sur le montant de la transaction (amt), le marchand (merchant), la catégorie de la transaction (category), les coordonnées géographiques, la population de la ville, et une variable cible (is_fraud) indiquant si la transaction est frauduleuse.

Nom de caracteristiques	Description	Type
Unnamed : 0	*Index ou identifiant unique de la transaction*	*Quantitatif (discret)*
Trans_date_trans_time	*Date et heure de la transaction*	*Catégoriel (nominal)*
Cc_num	*Numéro de carte de crédit*	*Quantitatif (discret)*
Merchant	*Nom du commerçant*	*Catégoriel (nominal)*
Category	*Catégorie de la transaction*	*Catégoriel (nominal)*
Amt	*Montant de la transaction*	*Quantitatif (continu)*
First	*Prénom du titulaire de la carte*	*Catégoriel (nominal)*
Last	*Nom de famille du titulaire de la carte*	*Catégoriel (nominal)*
Gender	*Genre du titulaire de la carte*	*Catégoriel (nominal)*
Street	*Adresse (rue) du titulaire de la carte*	*Catégoriel (nominal)*
City	*Ville du titulaire de la carte*	*Catégoriel (nominal)*
State	*État du titulaire de la carte*	*Catégoriel (nominal)*
Zip	*Code postal du titulaire de la carte*	*Quantitatif (discret)*
Lat	*Latitude de la localisation du commerçant*	*Quantitatif (continu)*
Long	*Longitude de la localisation du commerçant*	*Quantitatif (continu)*
City_pop	*Population de la ville*	*Quantitatif (discret)*
Job	*Profession du titulaire de la carte*	*Catégoriel (nominal)*
Dob	*Date de naissance du titulaire de la carte*	*Catégoriel (nominal)*
Trans_num	*Numéro de transaction unique*	*Catégoriel (nominal)*
Unix_time	*Heure de la transaction en format Unix*	*Quantitatif (discret)*

Merch_lat	*Latitude de la localisation du commerçant*	*Quantitatif (continu)*
Merch_long	*Longitude de la localisation du commerçant*	*Quantitatif (continu)*
Is_fraud	*Indicateur de fraude (1 si frauduleux, 0 sinon)*	*Catégoriel (nominal)*

Tableau 2 : Description des Caractéristiques des dataset

2 . Analyse Exploratoire des Données (EDA)

2.1 Détection des valeurs manquantes dans le dataset d'entraînement :

J'ai réalisé un tableau qui présente les noms des colonnes du dataset d'entraînement ainsi que le nombre de valeurs manquantes dans chacune d'elles. Cette analyse vise à identifier l'étendue des données manquantes, fournissant ainsi une base pour décider des stratégies appropriées de gestion des valeurs manquantes.

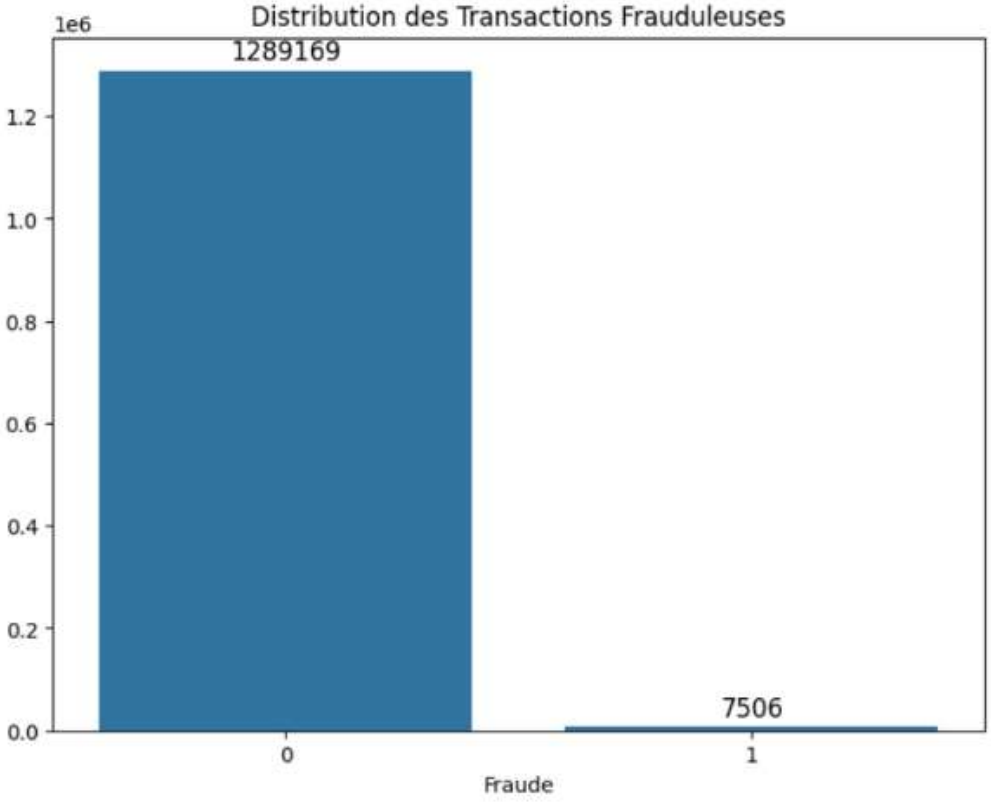
Nom de caracteristiques	Nombres des valeurs manquants :
Unnamed : 0	0
Trans_date_trans_time	0
cc_num	0
Merchant	0
Category	0
Amt	0
First	0
Last	0
Gender	0
Street	0
City	0
State	0
Zip	0
Lat	0
Long	0
City_pop	0
Job	0
Dob	0
Trans_num	0
Unix_time	0
Merch_lat	0
Merch_long	0
Is_fraud	0

Tableau 3 : Analyse des Valeurs Manquantes du Dataset

Cette table démontre que les données sont dans un état propre et complet, ce qui est une condition essentielle pour effectuer une analyse de données fiable. L'absence de valeurs manquantes dans le dataset d'entraînement renforce la qualité des informations sur lesquelles reposent nos modèles de machine learning. Des données propres permettent d'améliorer la précision et la robustesse des résultats,

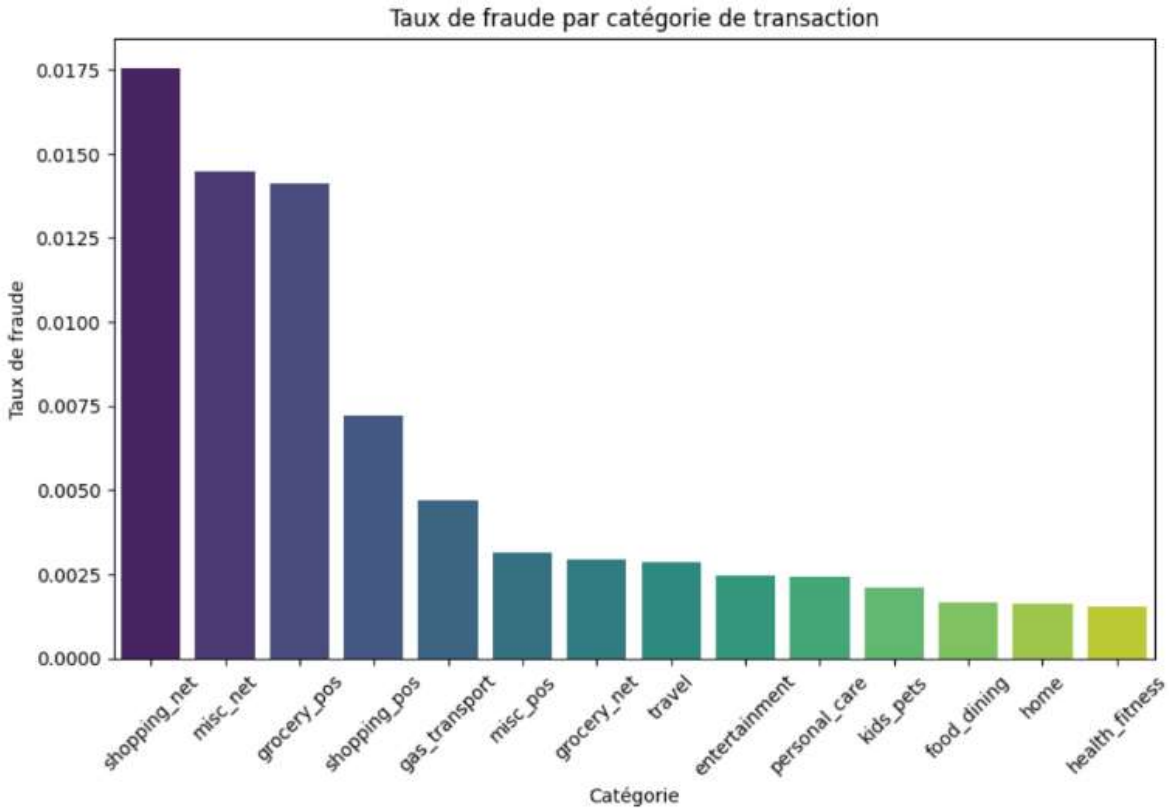
minimisant ainsi le risque d'erreurs d'interprétation ou de biais dans les analyses. Par conséquent, la validation de la qualité des données constitue une étape cruciale avant de passer à des phases ultérieures telles que la modélisation et l'évaluation, garantissant que les conclusions tirées sont fondées sur des bases solides.

2.2 Distribution de la variable cible (frauduleux ou non) :



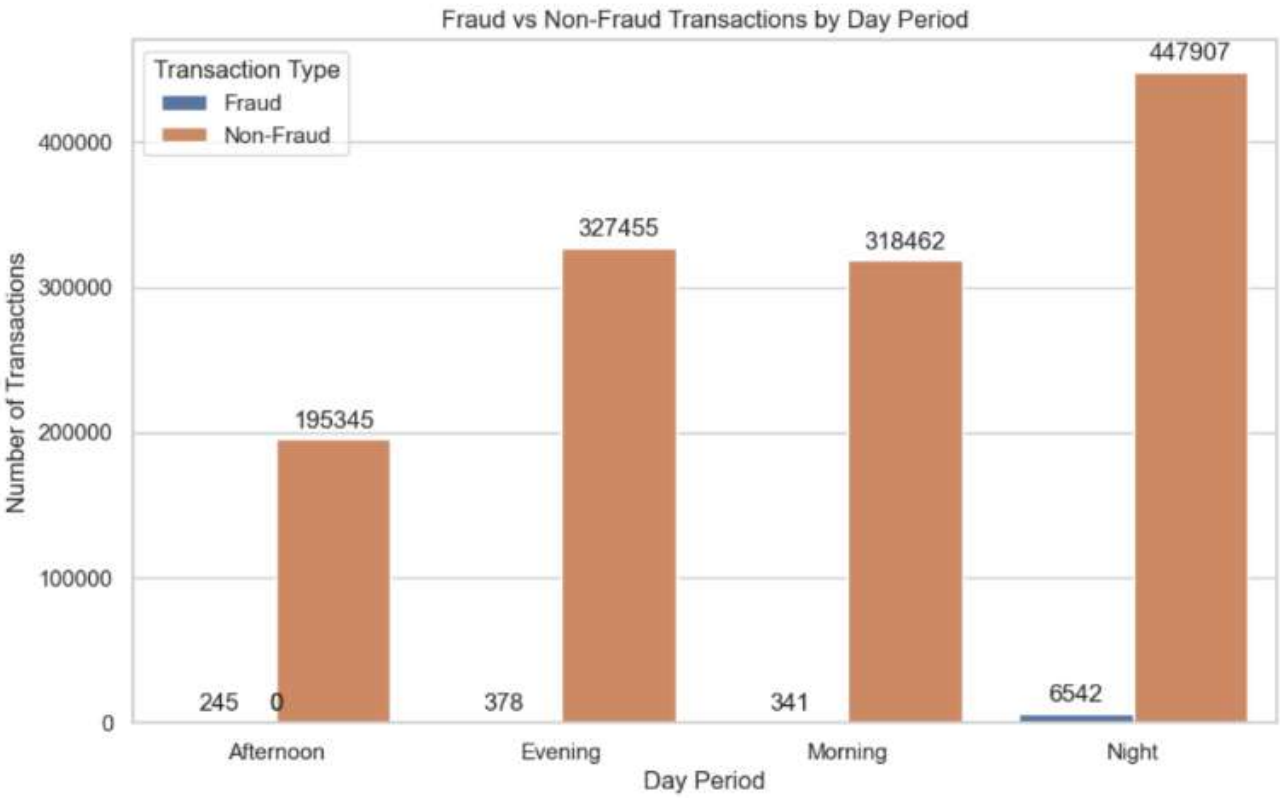
Ce graphique en barres illustre la distribution des transactions, en distinguant celles qui sont frauduleuses de celles qui ne le sont pas. On constate un déséquilibre marquant, avec plus d'un million de transactions considérées comme non frauduleuses contre environ 7 500 identifiées comme frauduleuses. Cette disparité significative entre les deux classes représente un défi majeur lors de la construction de modèles de détection de fraude, car les algorithmes de machine learning tendent à privilégier la classe majoritaire, ce qui peut entraîner une sous-détection des fraudes. Pour remédier à cette situation, il est crucial d'adopter des techniques spécifiques telles que l'oversampling des cas de fraude ou l'undersampling des transactions non frauduleuses, ainsi que d'utiliser des métriques d'évaluation adaptées. En mettant en œuvre ces approches, nous pouvons améliorer la performance des modèles en matière de détection de fraudes tout en minimisant les faux négatifs, assurant ainsi une protection plus efficace contre les activités frauduleuses.

2.3 Taux de fraude par catégorie de transaction :



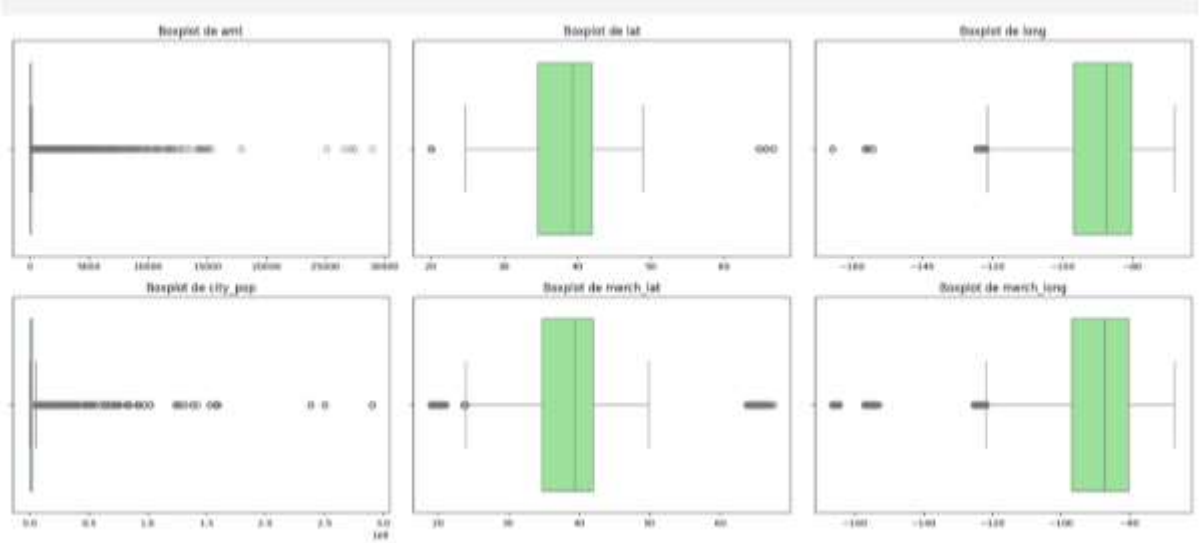
Ce graphique en barres compare le taux de fraude selon différentes catégories de transactions, révélant des variations significatives dans la propension à la fraude entre elles. Il met en évidence que certaines catégories, telles que "shopping_net" et "misc_net", affichent des taux de fraude nettement plus élevés, tandis que d'autres, comme "health_fitness" et "home", présentent des taux de fraude beaucoup plus bas. Cette visualisation est particulièrement utile pour identifier les catégories de transactions qui nécessitent une attention accrue dans les efforts de détection et de prévention de la fraude. En se concentrant sur les catégories les plus à risque, les institutions financières peuvent adapter leurs stratégies de surveillance et déployer des ressources de manière plus efficace. En conclusion, ce graphique offre une vue d'ensemble précieuse sur les catégories de transactions susceptibles d'être associées à des activités frauduleuses, servant ainsi de base pour orienter les initiatives de prévention et de gestion des risques.

2.4 Comparaison des Transactions Frauduleuses et Non-Frauduleuses par Période de la Journée :



Cette visualisation compare le nombre de transactions frauduleuses et non frauduleuses selon les différentes périodes de la journée, révélant une forte concentration de transactions, tant frauduleuses que non frauduleuses, durant les heures de jour, en particulier le matin et le soir. En revanche, les périodes de l'après-midi et de la nuit montrent un volume de transactions nettement inférieur, avec une fréquence de fraudes encore plus faible. Cette distribution inégale des transactions selon l'heure peut fournir des indices précieux pour la détection des fraudes. En résumé, la majorité des transactions, qu'elles soient frauduleuses ou non, se produisent pendant la journée, avec des pics d'activité aux heures stratégiques. Cette information est essentielle pour améliorer les systèmes de détection de fraude en concentrant les efforts de surveillance sur ces périodes clés. En conclusion, cette analyse met en évidence une corrélation significative entre l'heure de la journée et le volume ainsi que le type de transactions, permettant ainsi de développer des modèles de détection de fraude plus précis.

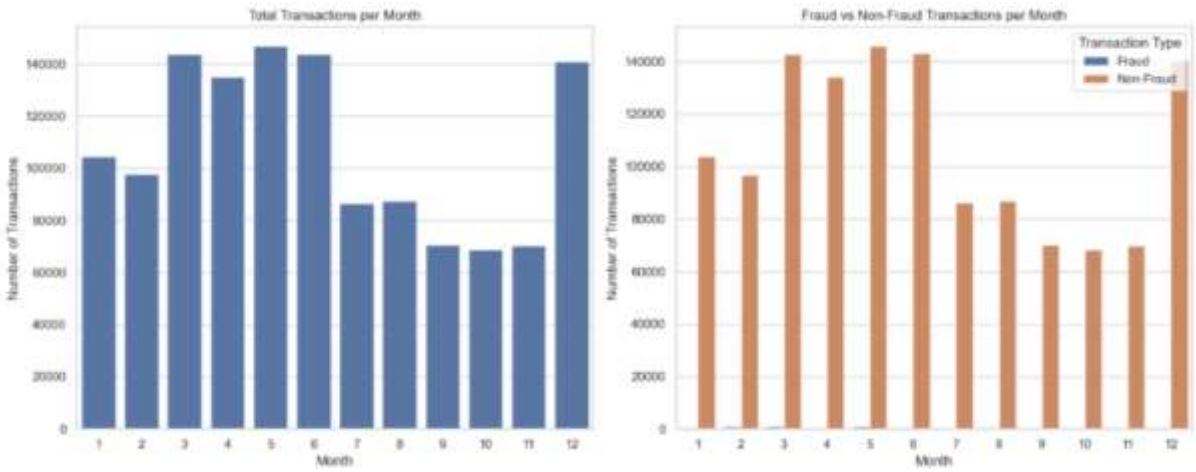
2.5 Analyse des distributions des variables numériques avec des boxplots :



Les boxplots présentés ici visualisent la distribution de différentes variables numériques au sein d'un ensemble de données, permettant ainsi d'identifier rapidement les valeurs atypiques (outliers), les quartiles et la médiane de chaque variable.

Il est évident qu'il existe une grande dispersion des valeurs pour les variables "amt" et "city_pop", ce qui suggère une variabilité considérable dans les montants des transactions et la taille des populations des villes. En revanche, les variables géographiques, telles que "lat" et "long" ainsi que leurs équivalents pour les marchands, montrent une distribution plus concentrée, indiquant des valeurs plus homogènes. Ces visualisations sont essentielles pour comprendre la nature des données, détecter d'éventuels problèmes de qualité et orienter les analyses ultérieures. En résumé, ces boxplots offrent un aperçu rapide de la distribution des variables numériques, facilitant ainsi l'identification des valeurs extrêmes et des tendances centrales, ce qui peut être crucial pour l'optimisation des modèles analytiques..

2.6 Transactions Frauduleuses vs Non-Frauduleuses par Mois :

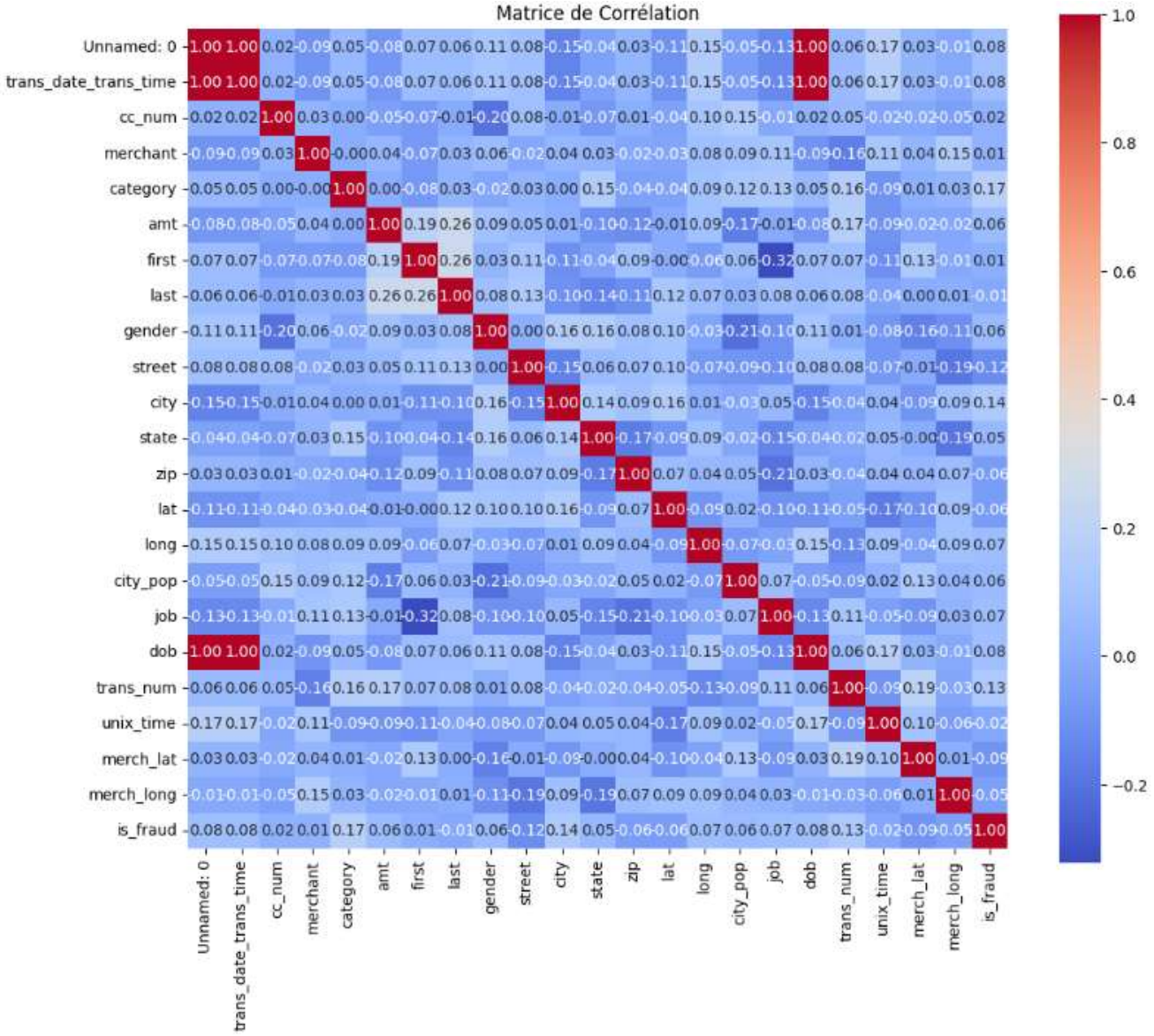


Les deux graphiques illustrent l'évolution du nombre de transactions, qu'elles soient frauduleuses ou non, au cours d'une année. Le premier graphique montre le nombre total de transactions par mois, tandis que le second détaille la répartition entre les transactions frauduleuses et non frauduleuses. On observe des variations saisonnières dans le volume total des transactions, avec des pics d'activité enregistrés certains mois. De plus, la répartition des transactions frauduleuses suit une tendance similaire à celle des transactions totales, indiquant que le taux de fraude pourrait rester relativement stable tout au long de l'année. Parallèlement, les boxplots visualisent la distribution de différentes variables numériques, permettant d'identifier rapidement les valeurs atypiques, les quartiles et la médiane. Une grande dispersion est observée pour les variables "amt" et "city_pop", suggérant une variabilité significative dans les montants des transactions et la taille des populations des villes, tandis que les variables géographiques montrent une distribution plus concentrée. Ces visualisations sont essentielles pour comprendre la nature des données, détecter d'éventuels problèmes de qualité et guider les analyses futures

3 . Prétraitement et Feature Engineering

Le prétraitement des données et l'ingénierie des caractéristiques sont des étapes fondamentales pour améliorer les performances des modèles de machine learning et de deep learning. Ces processus permettent de transformer et de sélectionner les variables les plus pertinentes afin de mieux représenter les schémas sous-jacents dans les données, tout en réduisant le bruit et la complexité. Dans cette section, nous avons appliqué diverses techniques, telles que la visualisation des corrélations, la réduction de dimensionnalité, et la détection des valeurs aberrantes, pour préparer les données de manière optimale. Ces étapes garantissent que notre modèle reçoit des données propres et prêtes à l'emploi, maximisant ainsi ses chances de bien généraliser.

3.1 Visualisation de la Matrice de Corrélation des Caractéristiques :



La matrice de corrélation présente ici les relations linéaires entre les paires de variables numériques de l'ensemble de données, avec des coefficients allant de -1 (corrélation négative forte) à 1 (corrélation positive forte). Les couleurs chaudes représentent des corrélations positives marquées, tandis que les couleurs froides illustrent des corrélations négatives notables. Certaines variables montrent des corrélations faibles, notamment des colonnes comme "Unnamed: 0", "cc_num", "dob", "trans_num", "first", "last", "street", "zip", "unix_time", "job", "gender", "lat", et "long", suggérant un faible lien linéaire avec les autres variables. Cependant, des corrélations plus significatives sont observées ailleurs, révélant des relations sous-jacentes importantes entre certaines variables. Ces informations sont cruciales pour identifier les variables les plus pertinentes pour l'analyse, tout en tenant compte des éventuels problèmes de multi colinéarité dans la construction de modèles prédictifs.

3.2 Analyse en Composantes Principales (PCA) :

L'Analyse en Composantes Principales (PCA) est une technique utilisée pour réduire la dimensionnalité des données tout en préservant autant d'information que possible. Elle transforme les variables d'origine en un nouvel ensemble de variables appelées "composantes principales", qui sont des combinaisons linéaires des variables initiales.

Dans les résultats que vous avez obtenus, les deux premières composantes principales expliquent ensemble environ **13.63%** et **13.42%** de la variance totale des données, ce qui signifie que ces deux composantes capturent environ **27%** de l'information présente dans les données d'origine.

Cela indique que les deux premières dimensions (ou axes) ne capturent qu'une partie modeste de la variabilité des données, et qu'il pourrait être nécessaire d'utiliser plus de composantes pour expliquer une plus grande proportion de la variance. Cependant, ces premières composantes sont toujours utiles pour une première visualisation ou analyse des tendances principales dans les données, notamment dans des applications de réduction de dimensionnalité pour la visualisation en 2D.

3.3 Évaluation de l'Importance des Caractéristiques avec XGBoost :

Features	Importance
Amt	0.299966
Category	0.250011
Is_fraud	0.199946
Merchant	0.179983
Trans_date_trans_time	0.149990
Merch_long	0.119923
City_pop	0.100092
Merch_lat	0.099988
State	0.079939
City	0.059998
Cc_num	0.010090
Dob	0.010078
Unnamed : 0	0.010056
Gender	0.010024
First	0.010021
Trans_num	0.010016
Last	0.010012
Long	0.010005
Lat	0.009992
Zip	0.009966
Job	0.009932
Unix_time	0.009924

Tableau 4 : Importance des Caractéristiques selon XGBoost

Les résultats de l'entraînement du modèle XGBoost montrent que la variable **"amt"** (29.99%) est la plus importante, suivie de **"category"** (24.99%) et **"is_fraud"** (19.99%), soulignant l'influence du montant, de la catégorie et de la nature de la

transaction sur la détection de fraude. **"merchant"** (18%) et **"trans_date_trans_time"** (14.99%) ont également une importance notable. Les coordonnées géographiques (**"merch_long"** et **"merch_lat"**) jouent un rôle modéré, tandis que des variables comme **"city_pop"** ou **"state"** ont une faible importance. Les caractéristiques avec moins de 1% d'importance, comme **"gender"** ou **"dob"**, apportent peu d'information utile au modèle.

3.4 Détection des Valeurs Aberrantes dans les Colonnes Numériques en Utilisant la Méthode IQR :

Features	Valeurs aberrantes
Unamed :0	0
cc_num	118789
Amt	67290
Zip	0
Lat	4679
Long	49922
City_pop	242674
Merch_lat	4967
Merch_long	41994

Tableau 5 : Valeurs Aberrantes Détectées avec la Méthode IQR

L'analyse des valeurs aberrantes détectées avec l'IQR révèle des variations significatives dans les caractéristiques de l'ensemble de données. La variable "cc_num" présente le plus grand nombre de valeurs aberrantes avec 118,789 occurrences, suggérant des transactions répétées ou des anomalies liées à des cartes de crédit. De même, "amt" (67,290 valeurs aberrantes) indique des montants de transactions atypiques, ce qui est pertinent pour la détection de fraudes. Les coordonnées géographiques, "lat" (4,679) et "long" (49,922), montrent également un nombre élevé d'anomalies, signalant des transactions provenant de régions inattendues. En revanche, des variables comme "Unnamed : 0", "zip" et "unix_time" n'ont pas de valeurs aberrantes, ce qui suggère qu'elles sont moins susceptibles de contenir des anomalies. La présence élevée de valeurs aberrantes, notamment dans les montants de transactions et les coordonnées géographiques, nécessite une attention particulière, car elles pourraient affecter la performance des modèles de détection de fraude. Ainsi, le traitement de ces valeurs aberrantes devrait être envisagé pour améliorer la robustesse du modèle.

3.5 Validation Croisée pour l'Évaluation des Caractéristiques :

- **Précision du modèle avec toutes les caractéristiques :** 0.9342113482561166
- **Précision du modèle sans certaines caractéristiques :** 0.9336337170069601
- **Différence par rapport au modèle complet :** 0.0005776312491564406

La précision du modèle, à 0,934, indique une performance exceptionnellement élevée, suggérant que près de 93,4 % des transactions sont classées correctement. Cette performance est impressionnante et indique que le modèle est très efficace dans sa capacité à détecter les transactions frauduleuses.

La différence minimale de 0,0006 par rapport au modèle complet indique que la suppression de certaines caractéristiques a eu peu d'impact sur la précision globale. Cela peut signifier que les caractéristiques exclues contenaient peu d'informations supplémentaires ou redondantes, ce qui permet au modèle de maintenir une performance similaire même sans ces variables.

En résumé, ces résultats soulignent l'efficacité du modèle et suggèrent que sa robustesse n'est pas uniquement dépendante de toutes les caractéristiques initiales, ce qui pourrait également indiquer un potentiel pour simplifier le modèle tout en maintenant des niveaux de précision élevés.

3.6 Conclusion :

En conclusion, le processus d'ingénierie des caractéristiques a permis d'identifier et de sélectionner les variables les plus pertinentes pour la détection de fraudes dans les transactions financières. L'analyse de la matrice de corrélation a révélé des liens significatifs entre certaines variables, tandis que l'utilisation de l'Analyse en Composantes Principales (PCA) a indiqué que les deux premières composantes ne capturent qu'une part modeste de la variance, suggérant qu'une exploration plus approfondie est nécessaire pour optimiser la représentation des données.

De plus, l'évaluation de l'importance des caractéristiques avec XGBoost a mis en évidence que des variables comme "amt" et "category" jouent un rôle crucial, tandis que d'autres, telles que "gender" et "dob", semblent apporter peu de valeur au modèle. La détection des valeurs aberrantes a également souligné l'importance de traiter ces anomalies pour garantir la robustesse du modèle.

Sur cette base, la décision de supprimer les colonnes "'Unnamed : 0', 'cc_num', 'dob', 'trans_num', 'first', 'last', 'street', 'zip', 'unix_time', 'job', 'gender', 'lat', 'long'" apparaît justifiée. Cette réduction des dimensions permettra de simplifier le modèle tout en préservant ses performances, favorisant ainsi une meilleure généralisation et une efficacité accrue dans la détection des fraudes.

4 . Technologies et Environnement Utilisés

4.1 Langages

- **Python** : Un langage de programmation polyvalent et puissant, largement utilisé pour le développement de logiciels, l'analyse de données, le machine learning et le deep learning. Sa syntaxe simple et sa vaste collection de bibliothèques facilitent le développement rapide d'applications.

4.2 Environnements

- **Jupyter Notebook** : Un environnement interactif qui permet aux utilisateurs de créer et de partager des documents contenant du code exécuté, des visualisations et du texte. Il est particulièrement utilisé pour l'exploration des données, l'analyse, et la documentation des résultats.

4.3 Bibliothèques et Modules

- **Pandas** : Une bibliothèque Python essentielle pour la manipulation et l'analyse de données, qui fournit des structures de données flexibles et des outils de haut niveau pour travailler avec des données étiquetées (DataFrames). Elle facilite le nettoyage, la transformation et l'analyse des données.

- **NumPy** : Une bibliothèque fondamentale pour le calcul scientifique en Python, qui fournit un support pour les tableaux multidimensionnels et les opérations mathématiques avancées sur ces tableaux. Elle est souvent utilisée comme base pour d'autres bibliothèques de données, comme Pandas.

- **Scikit-learn** : Une bibliothèque de machine learning pour Python qui offre des outils simples et efficaces pour l'apprentissage automatique et l'analyse des données. Elle inclut des algorithmes de classification, de régression, de clustering, et des techniques pour le prétraitement des données et l'évaluation des modèles.

- **TensorFlow** : Une bibliothèque open-source pour le deep learning développée par Google, qui permet de créer et de déployer des modèles d'apprentissage automatique. TensorFlow est largement utilisé pour les applications de machine learning et de deep learning, grâce à sa flexibilité et à sa performance.

- **Keras** : Une API de haut niveau pour le deep learning qui fonctionne au-dessus de TensorFlow. Elle simplifie la construction et l'entraînement de réseaux de neurones en fournissant des abstractions faciles à utiliser pour les modèles de deep learning.

- **Matplotlib** : Une bibliothèque de visualisation de données en Python qui permet de créer des graphiques et des visualisations statiques, animées et interactives. Elle est souvent utilisée pour explorer et présenter des résultats d'analyse de données.

- **Seaborn** : Une bibliothèque de visualisation basée sur Matplotlib qui fournit une interface de haut niveau pour dessiner des graphiques statistiques attrayants et informatifs. Elle facilite la création de visualisations complexes avec moins de code.

- **SMOTE (Synthetic Minority Over-sampling Technique)** : Une technique utilisée pour suréchantillonner les données déséquilibrées, en générant des exemples synthétiques de la classe minoritaire. SMOTE est particulièrement utile dans les problèmes de classification, où certaines classes sont sous-représentées.

CHAPITRE 4

IMPLÉMENTATION DES MODÈLES

Chapitre 4 : Implémentation des Modèles

1 . Modèles de Machine Learning

1.1 Régression Logistique

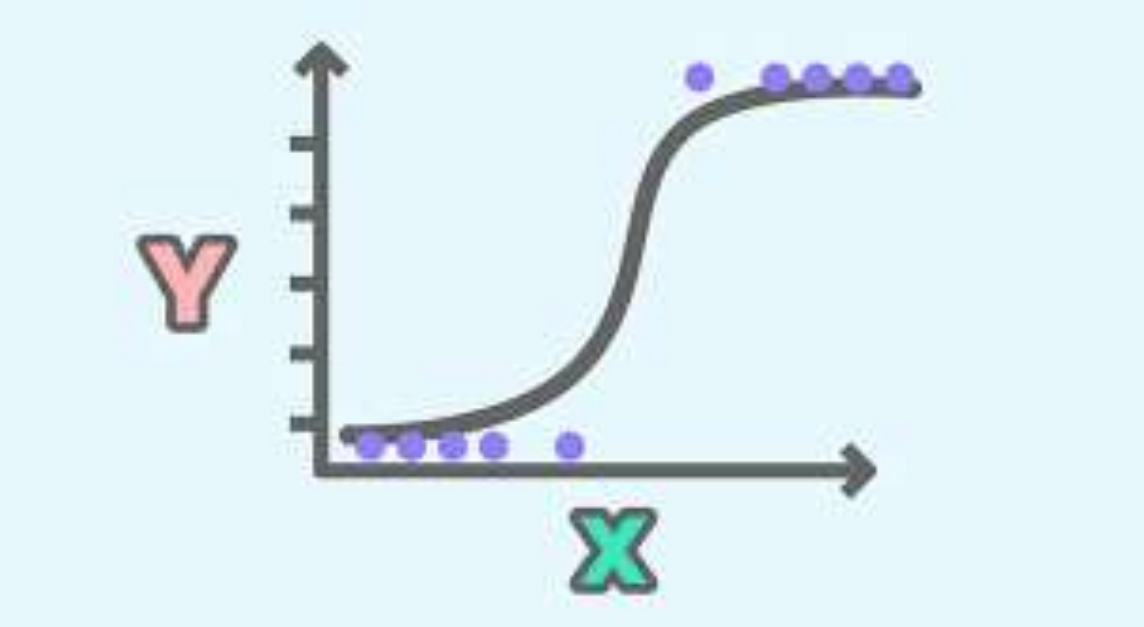
La régression logistique est une méthode statistique utilisée pour modéliser la probabilité d'un événement binaire (comme la fraude ou non) en fonction de variables explicatives continues ou catégorielles. Contrairement à la régression linéaire, elle est conçue pour des variables de sortie catégorielles, en particulier celles avec des résultats binaires. La fonction logistique, ou sigmoïde, est utilisée pour transformer la sortie en une probabilité comprise entre 0 et 1, ce qui permet de modéliser un événement avec une probabilité ppp à travers l'équation suivante :

$$\boldsymbol{logit(p) = ln\left(\frac{1-p}{p}\right)}$$

Le modèle est exprimé sous la forme suivante :

$$\boldsymbol{logit(p) = \beta_o + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon}$$

Ici, $\boldsymbol{\beta_0}$ est l'ordonnée à l'origine, les $\boldsymbol{\beta_n}$ sont les coefficients associés aux variables explicatives $\boldsymbol{X_n}$, et $\boldsymbol{\epsilon}$ est le terme d'erreur. Ce modèle est couramment estimé à l'aide de la méthode du maximum de vraisemblance, ce qui en fait un choix solide pour des problèmes tels que la détection de fraude, où il s'agit de prédire la probabilité qu'une transaction soit frauduleuse en fonction de plusieurs caractéristiques.



1.1.1 Modèle Sans SMOTE

Dans cette étape, un modèle de régression logistique a été entraîné sur le jeu de données initial déséquilibré, sans appliquer de méthode de suréchantillonnage. L'objectif est de tester les performances du modèle sur les classes majoritaire (non frauduleuse) et minoritaire (frauduleuse).

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.994208	~0.01
Rappel	0.999425	~0.01
F1-Score	0.99681	~0.01
Accuracy Globale	0.9936	

Tableau 6 : Performances de la Régression Logistique Sans SMOTE

Observations :

- **Classe 0 (Non frauduleuse)** : Le modèle montre d'excellents résultats pour la détection des transactions non frauduleuses, avec des scores de précision, rappel et F1 très élevés.
- **Classe 1 (Frauduleuse)** : Les résultats sont très faibles (précision, rappel et F1-Score à 0) pour les transactions frauduleuses, ce qui signifie que le modèle est incapable de détecter correctement les fraudes.
- **Conclusion** : L'accuracy globale de 0.994 est trompeuse, car elle reflète surtout la forte proportion de la classe majoritaire (transactions non frauduleuses). Le modèle a un biais important en faveur de la classe dominante, ce qui nuit à sa capacité à détecter les fraudes.

1.1.2 Modèle avec SMOTE

Dans cette phase, un modèle de régression logistique a été entraîné avec l'application de la méthode SMOTE pour équilibrer les classes. Le suréchantillonnage synthétique des transactions frauduleuses vise à améliorer la détection des fraudes.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.799465	0.938704
Rappel	0.950266	0.761638
F1-Score	0.868367	0.840952
Accuracy Globale	0.855952	

Tableau 7 : Performances de la Régression Logistique avec SMOTE

Observations :

- **Amélioration pour la classe 1 (Frauduleuse)** : Après l'application de SMOTE, le modèle a beaucoup amélioré sa capacité à détecter les transactions frauduleuses, avec une précision de 0.939, un rappel de 0.762, et un F1-Score de 0.841. Cela montre une meilleure balance entre la détection correcte des fraudes et les faux positifs.
- **Impact sur la classe 0 (Non frauduleuse)** : Les performances de la classe non frauduleuse ont légèrement baissé par rapport au modèle sans SMOTE, mais restent raisonnablement bonnes, avec un F1-Score de 0.868.

- **Accuracy globale** : Bien que l'accuracy globale soit de 0.856, il est plus pertinent de se concentrer sur les F1-scores et les rappels des deux classes pour évaluer la qualité du modèle.

1.1.3 Conclusion :

L'application de SMOTE a permis d'améliorer significativement la capacité du modèle à détecter les transactions frauduleuses, sans pour autant sacrifier trop de performance pour la classe non frauduleuse. Cette approche est plus équilibrée que le modèle sans SMOTE, où les fraudes n'étaient pratiquement pas détectées.

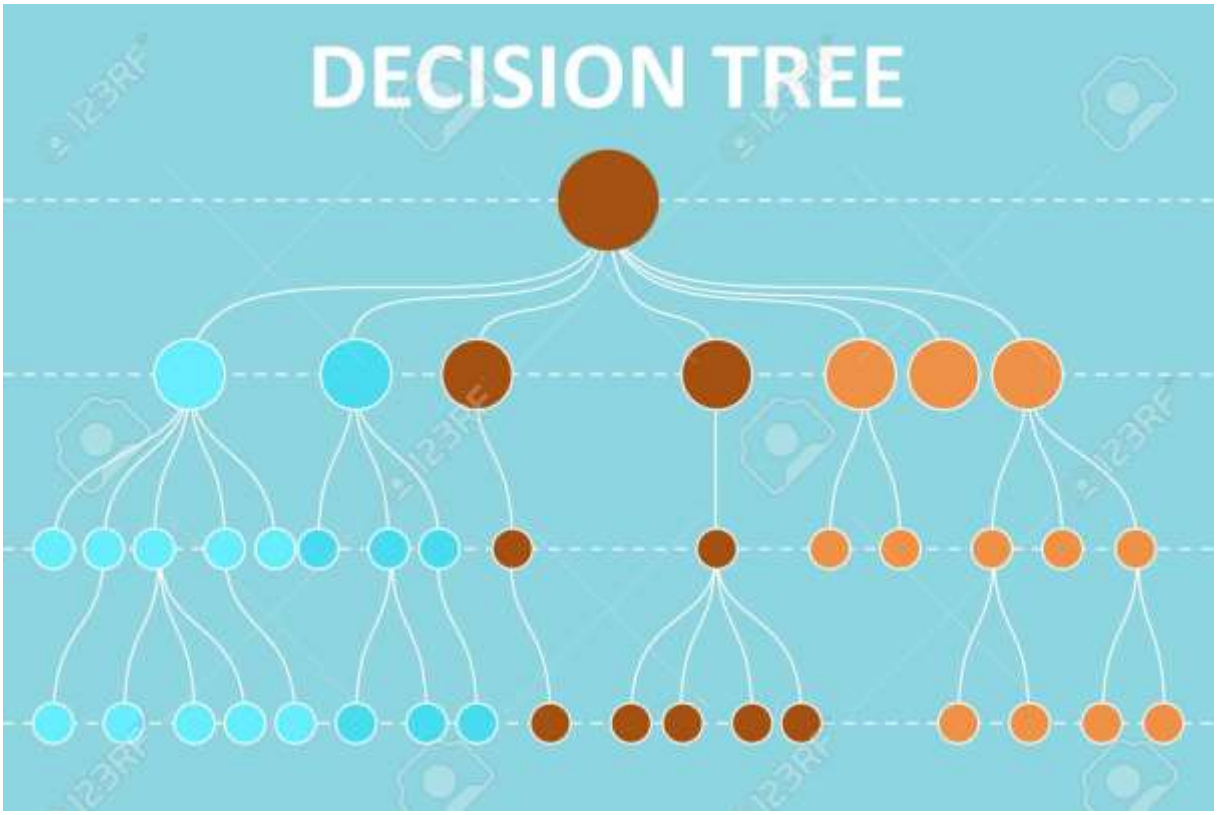
1.2 Arbre de Décision

Un arbre de décision est un modèle d'apprentissage supervisé utilisé pour résoudre des problèmes de classification et de régression. Il fonctionne en décomposant les données en un ensemble de règles décisionnelles sous la forme d'un arbre binaire où chaque nœud interne représente un test sur une caractéristique, chaque branche correspond à un résultat du test, et chaque feuille finale correspond à une prédiction (classe ou valeur continue). L'arbre est construit en choisissant à chaque étape la caractéristique qui permet de mieux séparer les données selon un critère d'impureté (comme l'indice de Gini ou l'entropie pour la classification, ou l'erreur quadratique pour la régression).

Mathématiquement, pour la classification, l'indice de Gini GGG pour une partition des données est défini comme :

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

Où p_i est la proportion des observations appartenant à la classe i parmi les C classes. L'arbre est construit en sélectionnant à chaque étape la caractéristique qui réduit le plus cet indice d'impureté. Les arbres de décision sont populaires pour leur interprétabilité, mais ils peuvent être sujets à l'overfitting, un problème souvent atténué par des techniques comme l'élagage ou la construction d'arbres de décision ensemble (e.g., forêts aléatoires).



1.2.1 Modèle Sans SMOTE

Dans cette expérimentation, un modèle d'arbre de décision a été entraîné sur les données initiales sans rééquilibrage des classes, afin d'observer son comportement face au déséquilibre des transactions frauduleuses et non frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.998321	0.564014
Rappel	0.998262	0.711526
F1-Score	0.998291	0.707988
Accuracy Globale	0.996603	

Tableau 8 : Performances de l'Arbre de Décision Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle montre une excellente performance pour la classe majoritaire (classe 0), avec une précision, un rappel et un F1-Score presque parfaits à 0.998. Cela est attendu, car les données non frauduleuses représentent la majorité du dataset.
- **Performance pour la Classe 1 (Frauduleuse)** : Bien que la précision et le rappel pour la classe 1 soient significativement plus faibles que pour la classe 0, avec des valeurs respectives de 0.704 et 0.712, l'arbre de décision parvient tout de même à

capturer une proportion décente des fraudes. Le F1-Score, à 0.708, révèle que le modèle peut détecter les fraudes, mais avec une certaine quantité de faux positifs et faux négatifs.

- **Accuracy Globale** : L'accuracy de 0.9966 est très élevée, mais en raison du déséquilibre des classes, elle n'est pas le meilleur indicateur pour évaluer la performance du modèle en matière de détection des fraudes. Cette mesure est biaisée par la majorité des transactions non frauduleuses.

1.2.2 Modèle Avec SMOTE :

Dans cette expérience, la technique SMOTE (Synthetic Minority Over-sampling Technique) a été utilisée pour générer des exemples synthétiques de la classe minoritaire (frauduleuse), afin de rééquilibrer le jeu de données avant d'entraîner l'arbre de décision.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.998601	0.704485
Rappel	0.996579	0.760160
F1-Score	0.997589	0.647560
Accuracy Globale	0.995211	

Tableau 9 :Performances de l'Arbre de Décision avec SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle conserve d'excellentes performances pour la classe majoritaire, avec une précision de 0.999 et un F1-Score de 0.998, légèrement inférieur à celui obtenu sans SMOTE mais toujours très élevé.
- **Performance pour la Classe 1 (Frauduleuse)** : Grâce à SMOTE, les performances pour la classe minoritaire (frauduleuse) se sont nettement améliorées. La précision a augmenté à 0.70, tandis que le rappel est passé à 0.76, indiquant que le modèle est maintenant plus efficace à détecter les fraudes. Le F1-Score pour cette classe, à 0.65, est une amélioration significative par rapport à l'expérience sans SMOTE.
- **Accuracy Globale** : L'accuracy globale reste élevée à 0.9952, bien qu'elle ne reflète pas entièrement l'amélioration en termes de détection des fraudes, qui est mise en évidence par les métriques pour la classe 1.

1.2.3 Conclusion :

Sans SMOTE, le modèle d'arbre de décision a montré une forte capacité à prédire les transactions non frauduleuses, mais a échoué à détecter efficacement les fraudes en raison du déséquilibre des classes. En appliquant SMOTE, les performances pour la classe minoritaire se sont nettement améliorées, avec des augmentations significatives en précision, rappel et F1-Score. Le modèle est ainsi mieux équilibré et plus performant pour détecter les transactions frauduleuses, tout en maintenant de

bonnes performances pour les transactions non frauduleuses. SMOTE a donc renforcé la capacité du modèle à gérer les données déséquilibrées.

1.3 Random Forest

Le Random Forest est un algorithme d'ensemble basé sur des arbres de décision, utilisé pour résoudre des problèmes de classification et de régression. Le principe fondamental est de créer plusieurs arbres de décision à partir de différents sous-échantillons aléatoires des données (méthode de bagging), puis d'agréger leurs prédictions pour produire un résultat final plus précis et stable. Chaque arbre dans la forêt est formé sur un échantillon aléatoire des données avec remplacement, et à chaque nœud, une sous-partie des caractéristiques est sélectionnée aléatoirement pour réduire l'impureté (via des métriques comme l'indice de Gini ou l'entropie).

Mathématiquement, si chaque arbre produit une prédiction $\boldsymbol{h_t(x)}$ ou $\boldsymbol{x}$ est une observation et $\boldsymbol{t}$ un arbre parmi $\boldsymbol{T}$ arbres dans la forêt, alors la prédiction finale du modèle est :

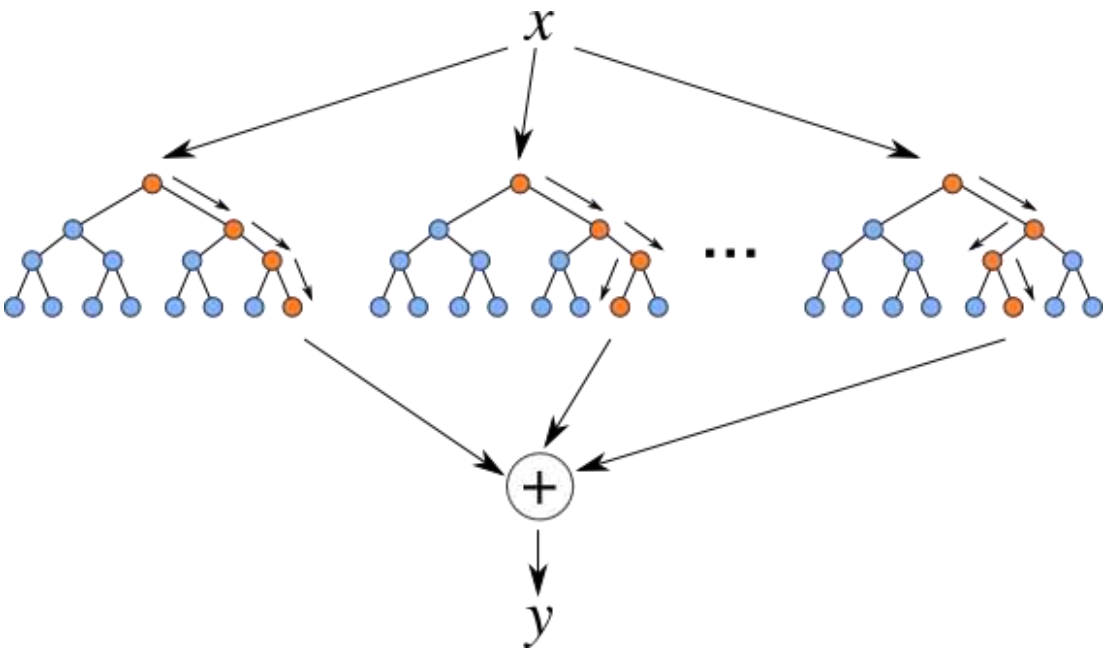
- Pour la classification : la classe majoritaire des prédictions individuelles

$$\hat{\boldsymbol{y}} = \boldsymbol{mode}(\boldsymbol{h_1(x)}, \boldsymbol{h_2(x)}, \dots, \boldsymbol{h\_x(x)})$$

- Pour la régression : la moyenne des prédictions individuelles

$$\hat{\boldsymbol{y}} = \frac{\boldsymbol{1}}{\boldsymbol{T}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \boldsymbol{h_t(x)}$$

Le Random Forest est particulièrement robuste aux problèmes de sur-apprentissage (overfitting) par rapport aux arbres de décision individuels, et il est



également efficace pour gérer les données présentant des déséquilibres de classes, comme dans le cas de la détection de fraude.

1.3.1 Modèle Sans SMOTE

Dans cette expérience, nous avons entraîné un modèle Random Forest sur le jeu de données original sans appliquer de technique de rééquilibrage comme SMOTE. L'objectif est d'examiner la performance du modèle face à un jeu de données déséquilibré, où la classe des transactions frauduleuses (classe 1) est beaucoup moins représentée.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.998326	0.864887
Rappel	0.999352	0.712192
F1-Score	0.998839	0.781147
Accuracy Globale	0.997690	

Tableau 10 :Performances de foret aléatoire Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle Random Forest montre une excellente capacité à prédire la classe majoritaire, affichant une précision de 0.998326 et un rappel de 0.999352. Ces résultats indiquent que le modèle détecte presque toutes les transactions non frauduleuses, ce qui est attendu étant donné leur prévalence dans le jeu de données.
- **Performance pour la Classe 1 (Frauduleuse)** : La précision pour la classe 1 est de 0.864887, ce qui est respectable, mais montre que le modèle a encore des difficultés à identifier certaines transactions frauduleuses. Le rappel de 0.712192 indique qu'il manque une proportion significative de fraudes, ce qui se reflète dans un F1-Score de 0.781147, témoignant d'un équilibre modéré entre précision et rappel.
- **Accuracy Globale** : L'accuracy de 0.997690 est très élevée, mais, tout comme dans les précédentes évaluations, elle ne reflète pas nécessairement la capacité du modèle à détecter les fraudes, étant donné le déséquilibre des classes.

1.3.2 Modèle Avec SMOTE

Dans cette expérience, nous avons appliqué la technique SMOTE pour rééquilibrer les classes avant d'entraîner le modèle Random Forest. L'objectif est d'améliorer la capacité du modèle à détecter les transactions frauduleuses en générant des exemples synthétiques pour la classe minoritaire.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.998863	0.729909
Rappel	0.998266	0.804797
F1-Score	0.998565	0.765526
Accuracy Globale	0.997147	

Tableau 11 :Performances de foret aléatoire Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle Random Forest continue d'afficher d'excellentes performances pour la classe majoritaire, avec une précision de 0.998863 et un rappel de 0.998266. Cela démontre la capacité du modèle à identifier presque toutes les transactions non frauduleuses, préservant ainsi une forte performance.
- **Performance pour la Classe 1 (Frauduleuse)** : Bien que la précision pour la classe 1 soit de 0.729909, ce qui représente une amélioration par rapport au modèle sans SMOTE, le rappel de 0.804797 indique une meilleure capacité à détecter les fraudes. Le F1-Score de 0.765526 montre également que le modèle gère mieux l'équilibre entre précision et rappel pour cette classe.
- **Accuracy Globale** : L'accuracy globale de 0.997147 demeure élevée, mais, comme précédemment mentionné, elle doit être interprétée avec prudence en raison du déséquilibre des classes dans le jeu de données.

1.3.3 Conclusion :

En résumé, le modèle Random Forest sans SMOTE démontre une excellente performance pour identifier les transactions non frauduleuses, mais il souffre de limitations notables dans la détection des fraudes, comme le révèlent ses faibles métriques pour la classe minoritaire. En revanche, l'application de SMOTE améliore significativement les capacités du modèle à détecter les transactions frauduleuses, avec des gains notables en précision, rappel et F1-Score pour la classe 1. Cette approche souligne l'importance du rééquilibrage des classes pour obtenir un modèle plus robuste et performant dans la détection des fraudes. En fin de compte, l'utilisation de SMOTE s'avère cruciale pour optimiser la détection des fraudes tout en maintenant des performances élevées pour la classe majoritaire.

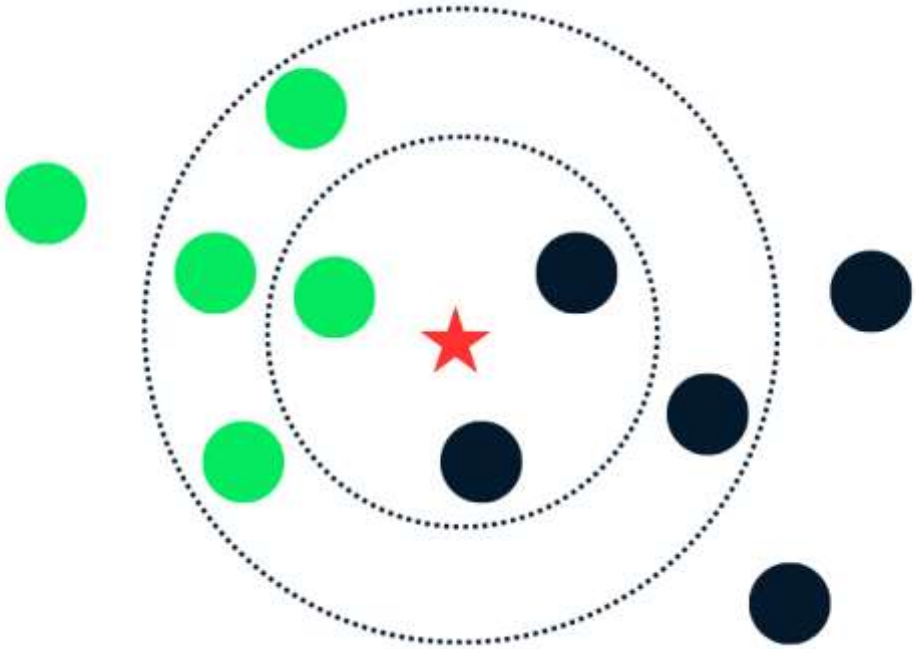
1.4 K-Nearest Neighbors (KNN)

Le K-Nearest Neighbors (KNN) est un algorithme d'apprentissage supervisé utilisé principalement pour des tâches de classification et de régression. Il fonctionne en identifiant les K voisins les plus proches d'un point de données dans un espace de caractéristiques donné, et en assignant une classe à ce point en fonction de la majorité des classes des voisins identifiés. L'algorithme utilise souvent la distance euclidienne comme métrique pour évaluer la proximité, bien que d'autres mesures de distance puissent également être utilisées, comme la distance de Manhattan ou la distance de Minkowski.

Mathématiquement, pour un point de données x, KNN évalue les distances $d(x, x_i)$ entre x et tous les autres points x_i de l'ensemble d'entraînement. Les K plus proches voisins sont déterminés, et la classe prédite $\hat{y}$ est donnée par :

$$\hat{y} = mode(y_{i1}, y_{i2}, \ldots, y_{iK})$$

Où y_{ij} représente les classes des K voisins les plus proches de x. KNN est sensible aux choix de K ainsi qu'à la distribution des données, ce qui peut entraîner des performances inégales, surtout en présence de déséquilibre entre les classes.



1.4.1 Modèle Sans SMOTE

Dans cette étape, nous avons entraîné des modèles KNN avec différents paramètres K sans appliquer la technique de suréchantillonnage SMOTE. L'objectif est d'évaluer la performance des modèles sur le jeu de données déséquilibré.

Paramètre K	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)	Accuracy Globale
K=3			
Précision	0.997559	0.725229	0.996298
Rappel	0.998720	0.580280	
F1-Score	0.998139	0.644708	
K=5			
Précision	0.997077	0.734975	0.996051
Rappel	0.998957	0.497002	
F1-Score	0.998016	0.593005	
K=7			
Précision	0.996668	0.712695	0.995685
Rappel	0.998999	0.426382	
F1-Score	0.997833	0.533556	

Tableau 12 :Performances de KNN Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : À chaque valeur de K, le modèle KNN montre des performances remarquables pour la classe majoritaire, avec des précisions, rappels et F1-Scores supérieurs à 0.996. Cela indique une bonne capacité à classer correctement les transactions non frauduleuses.
- **Performance pour la Classe 1 (Frauduleuse)** : Les résultats pour la classe 1 sont nettement inférieurs à ceux de la classe 0. La précision varie entre 0.712 et 0.734, tandis que le rappel est particulièrement faible, atteignant un maximum de 0.580 avec K=3 et chutant à 0.426 avec K=7. Cela révèle que le modèle peine à détecter efficacement les transactions frauduleuses, entraînant un nombre significatif de faux négatifs.
- **Accuracy Globale** : L'accuracy globale reste élevée, variant entre 0.995685 et 0.996298, mais elle masque la faible performance du modèle sur la classe minoritaire. Comme pour les autres modèles, l'accuracy peut être trompeuse dans le contexte de données déséquilibrées.

1.4.2 Modèle Avec SMOTE

Dans cette étape, nous avons appliqué la technique de suréchantillonnage SMOTE avant d'entraîner les modèles KNN. Cela vise à améliorer la capacité des modèles à détecter la classe minoritaire (transactions frauduleuses).

Paramètre K	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)	Accuracy Globale
K=3			
Précision	0.998894	0.460113	0.993399
Rappel	0.994462	0.810793	
F1-Score	0.996673	0.587072	
K=5			
Précision	0.999016	0.384900	0.991332
Rappel	0.992259	0.832112	
F1-Score	0.995626	0.526338	
K=7			
Précision	0.999065	0.338156	0.989554
Rappel	0.990420	0.840773	
F1-Score	0.994724	0.482324	

Tableau 13 : Performances de KNN avec SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Avec l'application de SMOTE, le modèle KNN conserve une excellente performance pour la classe majoritaire, affichant des précisions autour de 0.999 pour toutes les valeurs de K. Le rappel, bien que légèrement inférieur, reste supérieur à 0.990.
- **Performance pour la Classe 1 (Frauduleuse)** : Les résultats pour la classe 1 montrent une amélioration notable par rapport à la situation sans SMOTE. Bien que la précision demeure inférieure (variant de 0.338 à 0.460), le rappel atteint jusqu'à 0.840, indiquant que le modèle parvient à identifier un plus grand nombre de transactions frauduleuses. Toutefois, le F1-Score reste faible, ce qui signifie que le modèle a encore des difficultés à équilibrer entre précision et rappel pour la classe minoritaire.
- **Accuracy Globale** : L'accuracy globale est légèrement inférieure par rapport aux résultats sans SMOTE, variant entre 0.989 et 0.993. Cela reflète la difficulté du modèle à équilibrer la classification correcte des deux classes dans un jeu de données déséquilibré.

1.4.3 CONCLUSION :

Pour la conclusion, sans l'application de SMOTE, les modèles KNN affichent une performance exceptionnelle pour la classe majoritaire (non frauduleuse), mais peinent à détecter efficacement les transactions frauduleuses, ce qui se traduit par des métriques significativement plus faibles pour la classe 1 et une accuracy élevée pouvant masquer cette faiblesse. En revanche, l'application de la technique SMOTE améliore la détection des fraudes en augmentant le rappel pour la classe minoritaire tout en maintenant une bonne précision pour la classe majoritaire. Néanmoins, la précision pour la classe 1 demeure insuffisante, ce qui souligne que, malgré ces améliorations, le modèle KNN doit encore surmonter des défis pour équilibrer la détection des fraudes tout en minimisant les faux positifs.

1.5 Support Vector Machine (SVM)

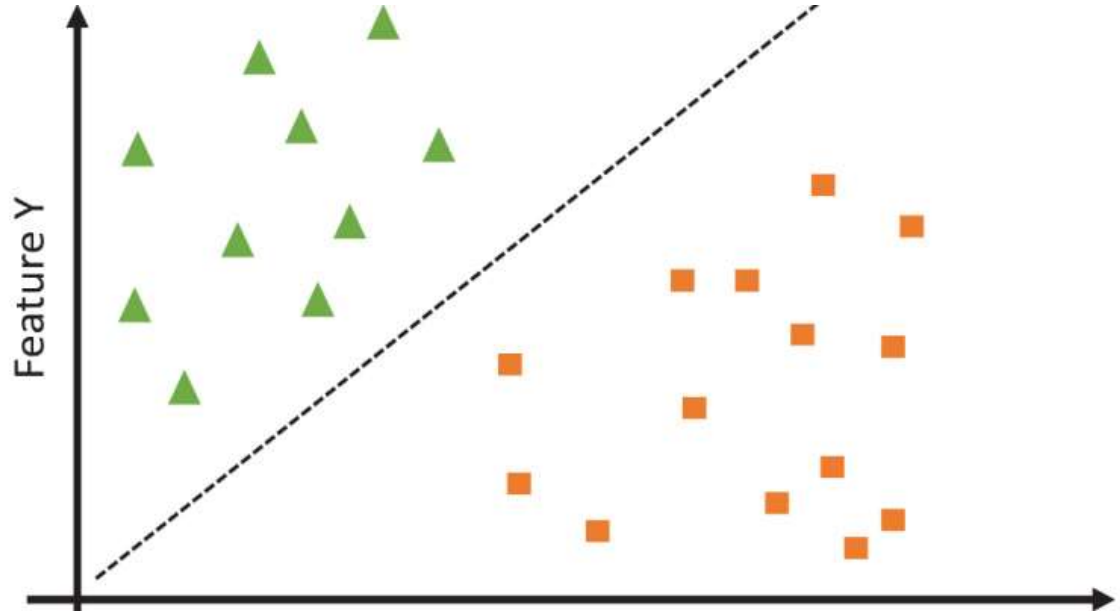
Le **Support Vector Machine (SVM)** est un algorithme d'apprentissage supervisé principalement utilisé pour la classification et la régression. Dans le contexte de la détection de fraude, SVM est particulièrement utile car il cherche à déterminer l'hyperplan optimal qui sépare les différentes classes de données tout en maximisant la marge entre elles. Cela signifie qu'il identifie la séparation le plus large possible entre les points de données appartenant à différentes catégories, ce qui est essentiel dans des scénarios de déséquilibre des classes, comme ceux rencontrés dans la détection de fraudes.

Mathématiquement, le SVM cherche à résoudre le problème d'optimisation suivant :

$$\boldsymbol{min} \quad \frac{\mathbf{1}}{\mathbf{2}} \| \boldsymbol{w} \|^{\mathbf{2}}$$

$$\boldsymbol{y_i}(\boldsymbol{w} \,.\, \boldsymbol{x_i} + \boldsymbol{b}) \;\geq \mathbf{1} \,, \forall \boldsymbol{i}$$

Où $\boldsymbol{w}$ est le vecteur de poids, $\boldsymbol{x\_i}$ représente les vecteurs d'entrée, $\boldsymbol{y_i}$ sont les étiquettes des classes (1 ou -1), et $\boldsymbol{b}$ est le biais. En maximisant la marge, SVM offre une robuste généralisation, essentielle pour des tâches telles que la détection de transactions frauduleuses.



1.5.1 Modèle Sans SMOTE

Dans cette étape, nous avons entraîné un modèle SVM sans appliquer la technique de suréchantillonnage SMOTE. L'objectif était d'évaluer la performance du modèle sur le jeu de données déséquilibré d'origine.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9961	0.3935
Rappel	1.0000	0.5239
F1-Score	0.9981	0.4494
Accuracy Globale	0.9961	

Tableau 14 :Performances de SVM Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle SVM présente une excellente performance pour la classe majoritaire, avec une précision de 0.9961, un rappel parfait de 1.0000, et un F1-Score très élevé de 0.9981, ce qui reflète sa capacité à identifier presque toutes les transactions non frauduleuses.
- **Performance pour la Classe 1 (Frauduleuse)** : Les métriques pour la classe 1 sont nettement inférieures, avec une précision de 0.3935, un rappel de 0.5239 et un F1-Score de 0.4494, indiquant que le modèle a du mal à détecter les transactions frauduleuses, ce qui est problématique dans un contexte où la détection de fraudes est cruciale.
- **Accuracy Globale** : Avec une accuracy globale de 0.9961, le modèle semble performant au premier abord. Cependant, cette mesure est trompeuse en raison du

déséquilibre des classes, car elle ne reflète pas la véritable capacité du modèle à détecter les fraudes.

1.5.2 Modèle Avec SMOTE

Dans cette étape, nous avons appliqué SMOTE pour entraîner un modèle SVM, visant à corriger le déséquilibre des classes.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9989	0.7680
Rappel	0.9498	0.8000
F1-Score	0.9738	0.7837
Accuracy Globale	0.9490	

Tableau 15 :Performances de SVM Avec SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle SVM, après l'application de SMOTE, maintient une excellente précision de 0.9989 et un rappel de 0.9498. Le F1-Score de 0.9738 indique que le modèle reste très efficace pour identifier les transactions non frauduleuses, bien que le rappel ait légèrement diminué par rapport au modèle sans SMOTE.
- **Performance pour la Classe 1 (Frauduleuse)** : L'application de SMOTE a permis d'améliorer la détection des fraudes, avec une précision de 0.7680 et un rappel de 0.8000. Le F1-Score de 0.7837 montre une amélioration significative par rapport au modèle sans SMOTE, révélant une meilleure capacité à identifier les transactions frauduleuses.
- **Accuracy Globale** : L'accuracy globale de 0.9490, bien que toujours élevée, indique que le modèle a un peu plus de faux positifs ou faux négatifs qu'auparavant. Toutefois, les performances des classes individuelles montrent une avancée dans la détection des fraudes

1.5.3 Conclusion Globale :

Sans l'application de SMOTE, le modèle SVM démontre une forte capacité à identifier les transactions non frauduleuses, mais il peine à détecter efficacement les fraudes, avec des métriques faibles pour la classe 1. En revanche, avec l'application de SMOTE, le modèle présente une amélioration notable dans la détection des transactions frauduleuses, tout en maintenant une bonne performance pour la classe majoritaire. Cela souligne l'importance des techniques de rééquilibrage pour améliorer les performances des modèles dans des contextes de déséquilibre des classes.

1.6 Auto encoder

L'auto encodeur est un type de réseau de neurones conçu pour l'apprentissage non supervisé, qui vise à apprendre une représentation compacte d'un ensemble de données, souvent dans le but de réduire la dimensionnalité tout en préservant les caractéristiques essentielles. Dans le cadre de la détection de fraude, les auto encodeurs sont particulièrement efficaces pour détecter des anomalies, telles que des transactions frauduleuses, en apprenant à reconstruire des exemples de transactions valides. Lorsqu'une transaction anormale est présentée à l'auto encodeur, sa reconstruction présente souvent une erreur plus élevée, signalant ainsi une potentielle fraude.

Mathématiquement, un auto encodeur peut être défini par la minimisation de la fonction de perte suivante :

$$L(x, \hat{x}) = \frac{1}{n}\sum_{i=1}^{n} \| x_i - \hat{x}_i \|$$

Où L est la fonction de perte, x est l'entrée originale, $\hat{x}$ est la reconstruction produite par le réseau, et $\|\cdot\|$ représente la norme L2, mesurant la différence entre l'entrée et sa reconstruction. Cette approche permet d'identifier efficacement des anomalies en évaluant la qualité de la reconstruction des transactions.

1.6.1 Modèle Sans SMOTE

Dans cette étape, nous avons entraîné un auto encodeur sans appliquer la méthode SMOTE. L'objectif est d'évaluer la capacité de l'auto encodeur à détecter des transactions frauduleuses sur un jeu de données déséquilibré, où les transactions frauduleuses sont beaucoup moins fréquentes.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9969	0.0578
Rappel	0.9526	0.4990
F1-Score	0.9743	0.1035
Accuracy Globale	0.9500	

Tableau 16 :Performances de Auto encoder Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : L'auto encodeur démontre une performance remarquable pour la classe majoritaire, atteignant une précision de 0.9969 et un rappel de 0.9526. Le F1-Score de 0.9743 indique une capacité solide à identifier correctement les transactions non frauduleuses.
- **Performance pour la Classe 1 (Frauduleuse)** : En revanche, pour la classe minoritaire, la précision est très faible à 0.0578, tandis que le rappel est de 0.4990. Cela signifie que, bien que le modèle soit capable de détecter certaines transactions frauduleuses, il génère également un grand nombre de faux négatifs. Le F1-Score de 0.1035 reflète cette difficulté à identifier efficacement les fraudes.

- **Accuracy Globale** : L'accuracy globale de 0.9500 peut sembler acceptable, mais elle est trompeuse dans le contexte d'un jeu de données déséquilibré, où le modèle réussit principalement à prédire la classe majoritaire.

1.6.2 Modèle Avec SMOTE

Dans cette étape, nous avons appliqué SMOTE pour rééquilibrer la distribution des classes avant d'entraîner l'auto encodeur. L'objectif est de déterminer si cette méthode améliore la capacité du modèle à détecter les transactions frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9960	0.1800
Rappel	0.9450	0.5200
F1-Score	0.9704	0.2679
Accuracy Globale	0.9455	

Tableau 17 :Performances de Auto encoder avec SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : L'auto encodeur maintient une performance élevée pour la classe majoritaire avec une précision de 0.9960 et un rappel de 0.9450. Le F1-Score de 0.9704 démontre que le modèle continue d'être efficace pour prédire les transactions non frauduleuses.
- **Performance pour la Classe 1 (Frauduleuse)** : Pour la classe minoritaire, bien que la précision ait légèrement augmenté par rapport au modèle sans SMOTE, elle reste insuffisante à 0.1800. Le rappel de 0.5200 indique une amélioration dans la détection des transactions frauduleuses, mais le F1-Score de 0.2679 montre que le modèle a encore du mal à équilibrer entre les faux positifs et les faux négatifs.
- **Accuracy Globale** : L'accuracy globale de 0.9455 est légèrement inférieure à celle obtenue sans SMOTE, suggérant que l'équilibrage des classes a eu un impact sur la performance générale, bien que cela ne se traduise pas nécessairement par une meilleure détection des fraudes.

1.6.3 Conclusion :

Avec l'application de SMOTE, l'auto encodeur montre une légère amélioration dans la détection des transactions frauduleuses, surtout en termes de rappel. Cependant, la précision pour la classe minoritaire reste faible, ce qui indique que des efforts supplémentaires sont nécessaires pour optimiser le modèle dans un contexte de détection de fraudes. Ainsi, bien que SMOTE ait apporté quelques améliorations, il demeure un défi de concilier une bonne performance pour les deux classes, surtout dans un cadre de données déséquilibrées.

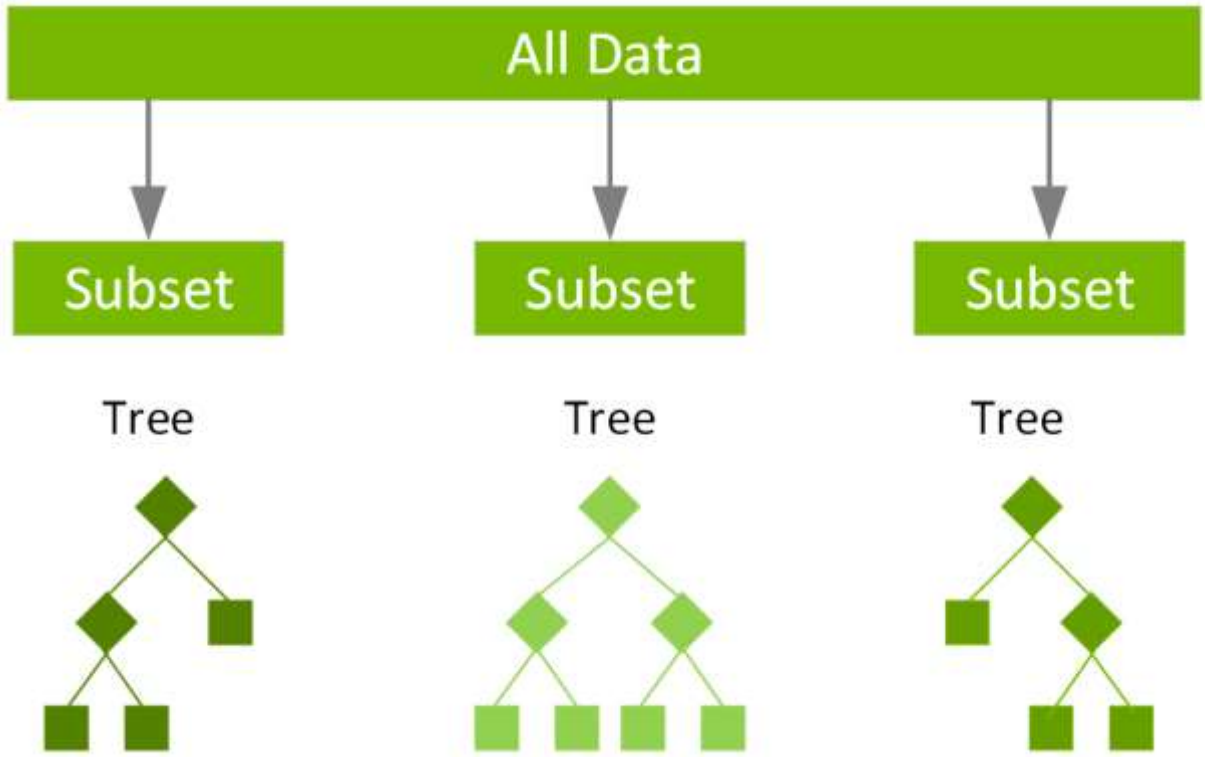
1.7 XGBoost

XGBoost (Extreme Gradient Boosting) est un algorithme de machine learning puissant basé sur l'approche des arbres de décision, spécifiquement conçu pour améliorer la performance prédictive tout en réduisant les erreurs. Dans le contexte de la détection de fraude, il est particulièrement apprécié pour sa rapidité, sa précision et sa capacité à traiter des ensembles de données déséquilibrés, ce qui est fréquent dans ce domaine. Le modèle XGBoost construit des arbres de décision de manière séquentielle, chaque arbre étant conçu pour corriger les erreurs des prédictions des arbres précédents, ce qui en fait un outil efficace pour capturer des relations complexes au sein des données.

Mathématiquement, l'algorithme XGBoost minimise la fonction de perte suivante lors de l'entraînement des modèles :

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Où $L(\theta)$ représente la fonction de perte totale, l est la fonction de perte qui mesure l'erreur entre la valeur réelle y_i et la prédiction $\hat{y}_i$, $\Omega(fk)$ est une fonction de régularisation qui pénalise la complexité des arbres, et K est le nombre total d'arbres. Grâce à cette approche, XGBoost permet de gérer efficacement des problèmes complexes et déséquilibrés, ce qui en fait un choix privilégié pour des applications telles que la détection de fraudes.



1.7.1 Modèle Sans SMOTE

Dans cette expérience, nous avons entraîné le modèle XGBoost sans appliquer la méthode SMOTE. L'objectif est de mesurer sa capacité à détecter des transactions frauduleuses sans intervention sur la distribution des classes, qui est fortement déséquilibrée.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9986	0.9185
Rappel	0.9996	0.7655
F1-Score	0.9991	0.8350
Accuracy Globale	0.9982	

Tableau 18 : Performances de XGBoost Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle XGBoost montre une performance exceptionnelle pour la classe majoritaire avec une précision de 0.9986 et un rappel de 0.9996, indiquant qu'il est très efficace pour identifier les transactions non frauduleuses. Le F1-Score de 0.9991 renforce cette impression, suggérant une faible incidence de faux positifs.
- **Performance pour la Classe 1 (Frauduleuse)** : Bien que le modèle réussisse à maintenir une bonne précision de 0.9185, le rappel pour la classe 1 est de 0.7655, indiquant une capacité modérée à détecter les fraudes. Le F1-Score de 0.8350 révèle qu'il y a encore une marge d'amélioration pour réduire les faux négatifs dans cette classe.
- **Accuracy Globale** : Avec une accuracy globale de 0.9982, le modèle présente une très bonne performance. Cependant, cette mesure élevée peut masquer les défis de détection des fraudes en raison du déséquilibre des classes.

1.7.2 Modèle Avec SMOTE

Dans cette étape, nous avons appliqué SMOTE pour rééquilibrer la distribution des classes avant d'entraîner l'auto encodeur. L'objectif est de déterminer si cette méthode améliore la capacité du modèle à détecter les transactions frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9942	0.8500
Rappel	0.9942	0.8399
F1-Score	0.9942	0.8399
Accuracy Globale	0.9750	

Tableau 19 :Performances de XGBoost avec SMOTE

Analyse des Résultats :

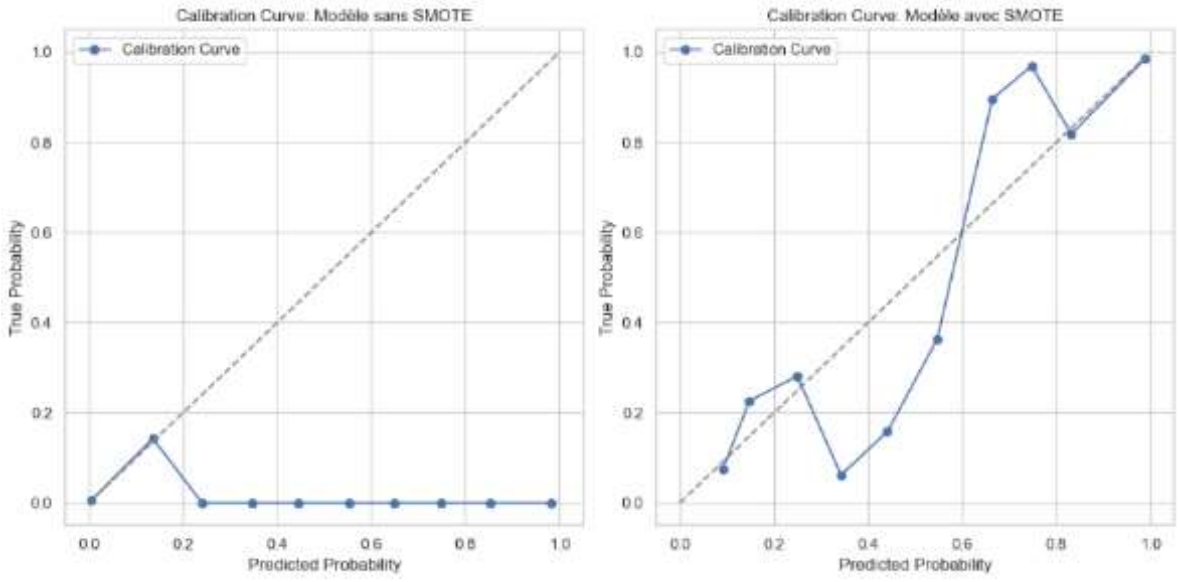
- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle XGBoost présente des résultats solides pour la classe majoritaire, avec une précision et un rappel de 0.9942. Cela indique que le modèle reste très compétent pour identifier les transactions non frauduleuses, maintenant un F1-Score de 0.9942.
- **Performance pour la Classe 1 (Frauduleuse)** : Pour la classe minoritaire, la précision est de 0.8500 et le rappel est de 0.8399, montrant une amélioration par rapport au modèle sans SMOTE. Le F1-Score de 0.8399 suggère que le modèle peut détecter un bon nombre de transactions frauduleuses, bien qu'il reste encore un défi à relever pour réduire les faux positifs et faux négatifs.
- **Accuracy Globale** : L'accuracy globale de 0.9750 est inférieure à celle obtenue sans SMOTE, mais elle est moins biaisée par la classe majoritaire, ce qui offre une vision plus équilibrée de la performance globale du modèle.

1.7.3 Conclusion :

Avec l'application de SMOTE, le modèle XGBoost montre une amélioration significative dans la détection des fraudes par rapport à l'approche sans SMOTE. Bien que la précision et le rappel pour la classe 0 soient légèrement inférieurs, les métriques pour la classe 1 ont considérablement progressé, indiquant que l'équilibrage des classes peut être bénéfique. Cependant, malgré ces améliorations, le modèle continue de faire face à des défis pour atteindre un équilibre optimal entre la détection des fraudes et la réduction des faux positifs, soulignant la complexité de la tâche dans des ensembles de données déséquilibrés.

1.8 graphique et visualisations :

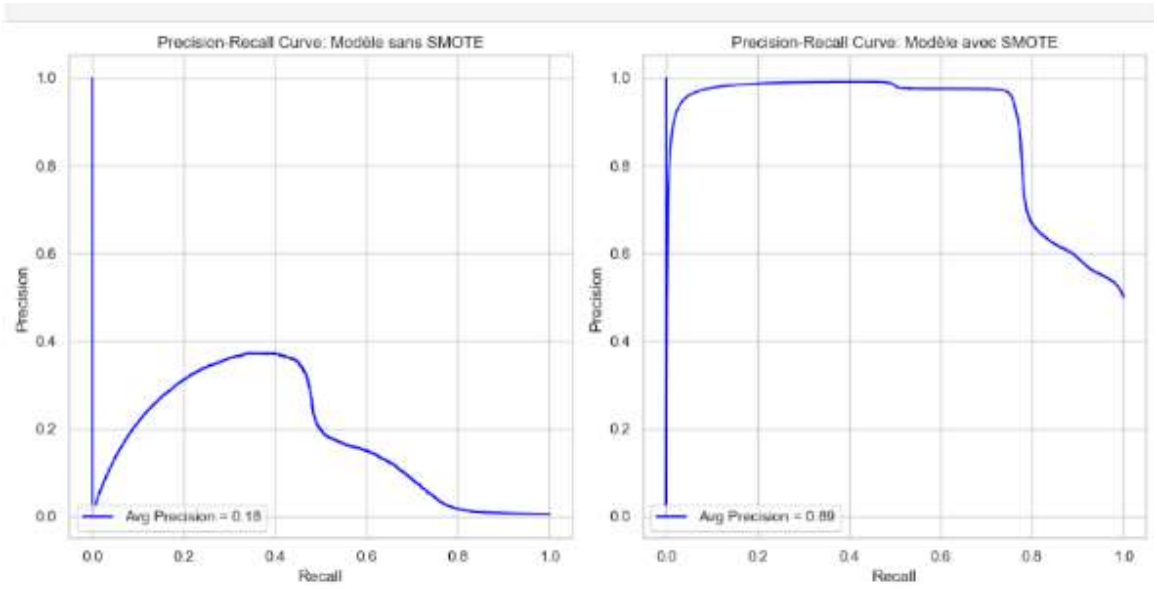
1.8.1 Régression Logistique



Ces courbes de calibration comparent la qualité des prédictions d'un modèle de régression logistique, avec et sans l'utilisation de la méthode SMOTE pour équilibrer

les classes. Une courbe de calibration idéale se situe le long de la diagonale, indiquant une correspondance parfaite entre les probabilités prédites et observées. Sans SMOTE, le modèle présente une mauvaise calibration, surtout pour les probabilités élevées, s'écartant considérablement de la diagonale. En revanche, avec SMOTE, la calibration s'améliore nettement, la courbe se rapprochant de l'idéal. Cela indique que l'application de SMOTE a amélioré la fiabilité des probabilités générées par le modèle, rendant ses prédictions plus cohérentes.

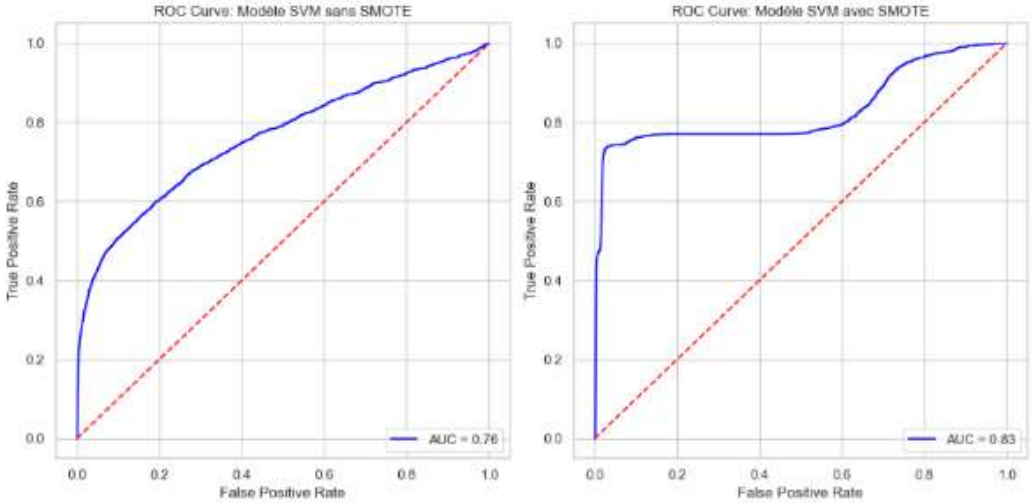
1.8.1.1 Analyse des Performances : Courbes Précision-Rappel avec et sans smote



Les courbes de précision-rappel illustrent la performance d'un modèle de régression logistique dans deux configurations : avec et sans l'application de SMOTE pour équilibrer les classes. Elles montrent l'équilibre entre la précision (la proportion de prédictions correctes parmi les instances positives) et le rappel (la proportion d'instances positives correctement identifiées). Sans SMOTE, la courbe est irrégulière, avec une baisse rapide de la précision à mesure que le rappel augmente. En revanche, avec SMOTE, la courbe est plus stable, maintenant une précision élevée même lorsque le rappel est important. Cela suggère que SMOTE a permis au modèle d'améliorer sa capacité à détecter les transactions positives (frauduleuses), tout en maintenant une précision satisfaisante. En résumé, SMOTE a eu un impact favorable sur la performance globale du modèle, surtout pour les classes minoritaires.

1.8.2 SVM

1.8.2.1 Évaluation des Modèles : Courbes ROC avec et sans SMOTE



Les courbes ROC comparent la performance d'un modèle SVM dans deux configurations : sans et avec l'application de SMOTE pour rééquilibrer les classes. Ces courbes montrent le compromis entre la sensibilité (capacité à détecter les vrais positifs) et la spécificité (capacité à éviter les faux positifs). Sans SMOTE, la courbe ROC est moins favorable, avec une aire sous la courbe (AUC) plus faible, indiquant une performance limitée dans la distinction des classes. En revanche, l'application de SMOTE permet une nette amélioration, avec une courbe ROC plus proche de la diagonale idéale et une AUC plus élevée. Cela indique que SMOTE a renforcé la capacité du modèle à différencier les transactions normales des transactions frauduleuses, notamment pour la classe minoritaire. En résumé, SMOTE a significativement amélioré la performance de classification du modèle SVM.

1.8.2.2 Analyse du Coefficient de Corrélation de Matthews (MCC) :

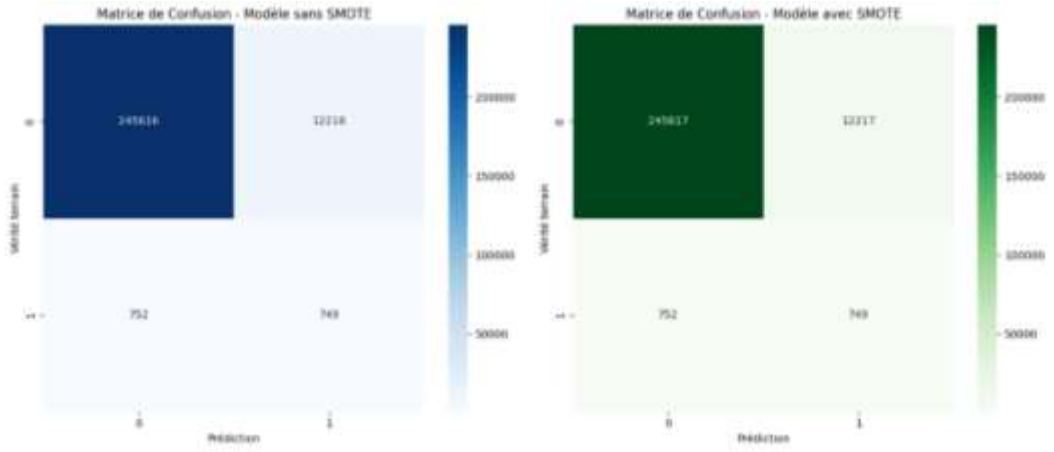
MCC sans SMOTE :	MCC avec SMOTE :
0.27	0.81

Tableau 20 : Analyse du MCC pour SVM avec et sans SMOTE

En appliquant SMOTE, le MCC du modèle SVM augmente considérablement, passant de 0.27 à 0.81. Cela révèle une amélioration substantielle de la capacité du modèle à discriminer efficacement entre les classes, notamment en ce qui concerne la classe minoritaire. En résumé, l'utilisation de SMOTE a considérablement renforcé la précision et la fiabilité des prédictions du modèle SVM, rendant ce dernier beaucoup plus performant dans le contexte de données déséquilibrées.

1.8.3 Auto encodeur

1.8.3.1 Analyse des Matrices de Confusion : Impact du Suréchantillonnage SMOTE :

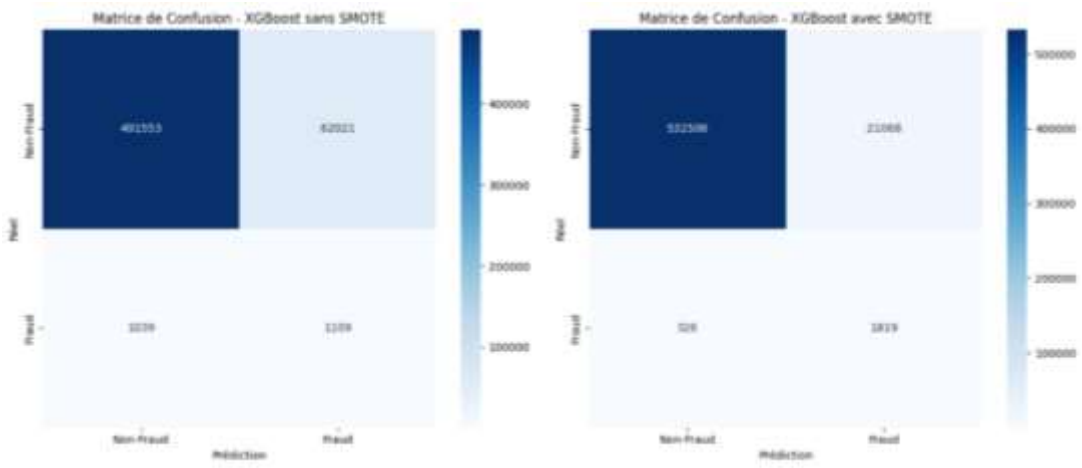


Les matrices de confusion comparant les résultats de classification de l'auto encodeur, avec et sans l'application de la méthode SMOTE, révèlent les performances du modèle en termes de prédiction des catégories réelles. Chaque cellule représente le nombre de prédictions correctes ou incorrectes, par rapport à la classe réelle. L'introduction de SMOTE améliore légèrement la détection de la classe minoritaire (côté inférieur droit), mais son impact sur la classe majoritaire reste minime. Cela indique que bien que SMOTE ait contribué à un meilleur équilibrage des classes, l'amélioration globale de la performance du modèle reste limitée.

En résumé, l'impact de SMOTE sur l'auto encodeur est modeste, avec une légère amélioration dans la classification des exemples de la classe minoritaire.

1.8.4 XGBoost

1.8.4.1 Analyse du Coefficient de Corrélation de Matthews (MCC) :

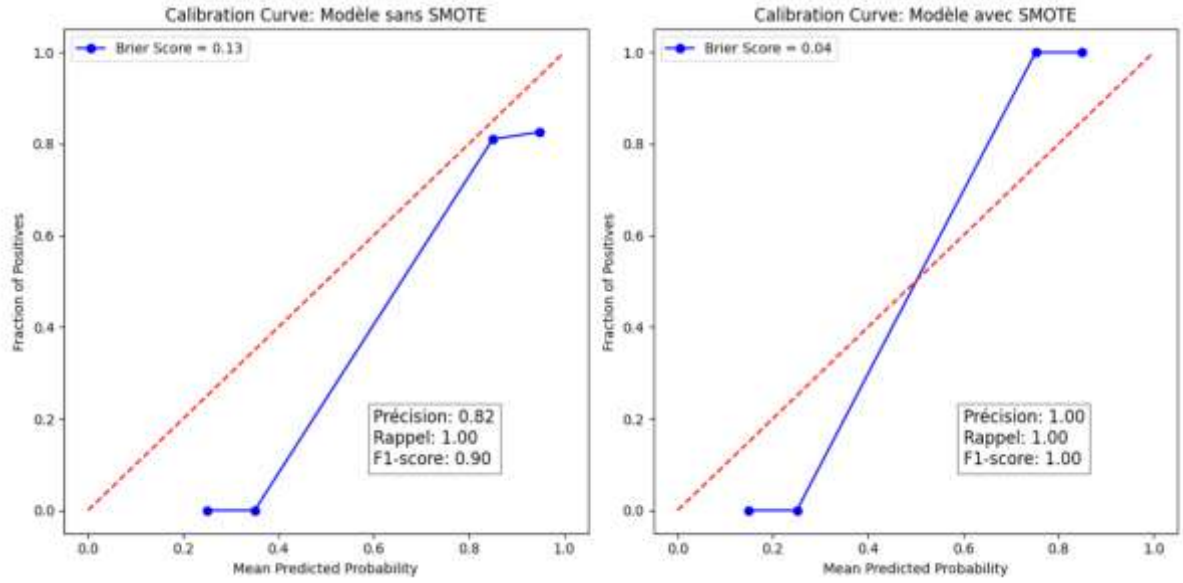


Les matrices de confusion illustrant les résultats du modèle XGBoost, avant et après l'application de la méthode SMOTE, révèlent un contraste notable dans la performance de classification. En particulier, le coefficient de corrélation de Matthews (MCC) diminue de manière significative, passant de 0.84 à 0.61 après l'application de SMOTE. Cette réduction indique que, dans ce cas précis, l'équilibrage des classes via SMOTE a réduit la capacité du modèle à bien différencier les classes. Contrairement aux attentes habituelles, l'utilisation de SMOTE a ici entraîné une dégradation des performances du modèle XGBoost.

En résumé, l'application de SMOTE n'a pas été bénéfique pour XGBoost et a même eu un effet négatif sur ses performances, contrairement à ce qui est souvent observé avec d'autres algorithmes.

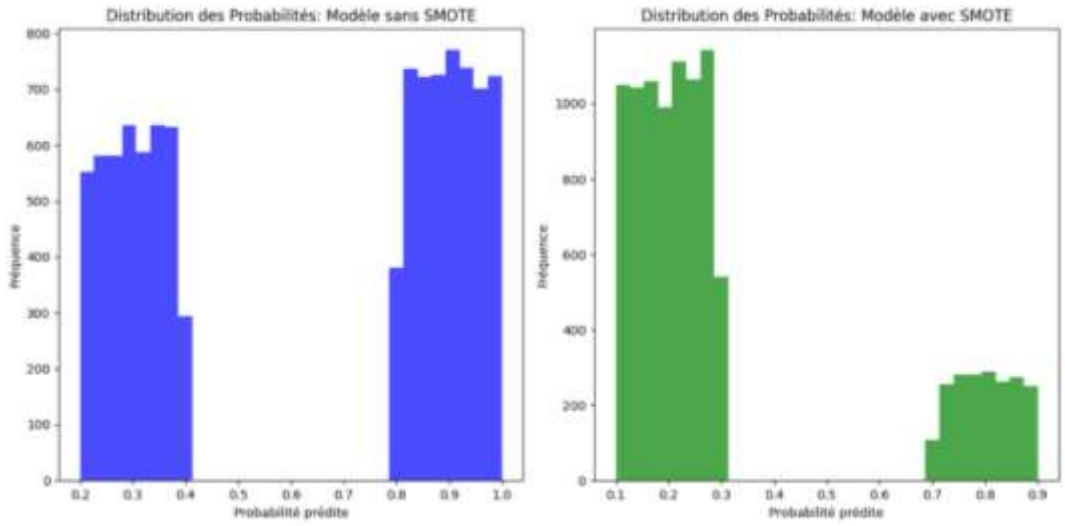
1.8.5 Random Forest

1.8.5.1 Analyse des Matrices de Confusion : Impact du Suréchantillonnage SMOTE :



Idéalement, la courbe devrait suivre une ligne diagonale, représentant une parfaite correspondance entre les probabilités prédites et les probabilités observées. Le modèle sans SMOTE présente une calibration moins précise, avec un score de Brier plus élevé et une courbe qui s'éloigne de la diagonale, notamment pour les probabilités plus élevées. En revanche, l'application de SMOTE améliore clairement la calibration, avec un score de Brier réduit et une courbe qui se rapproche beaucoup plus de la diagonale. Cela indique que l'équilibrage des classes via SMOTE a permis d'améliorer la fiabilité des prédictions du modèle.

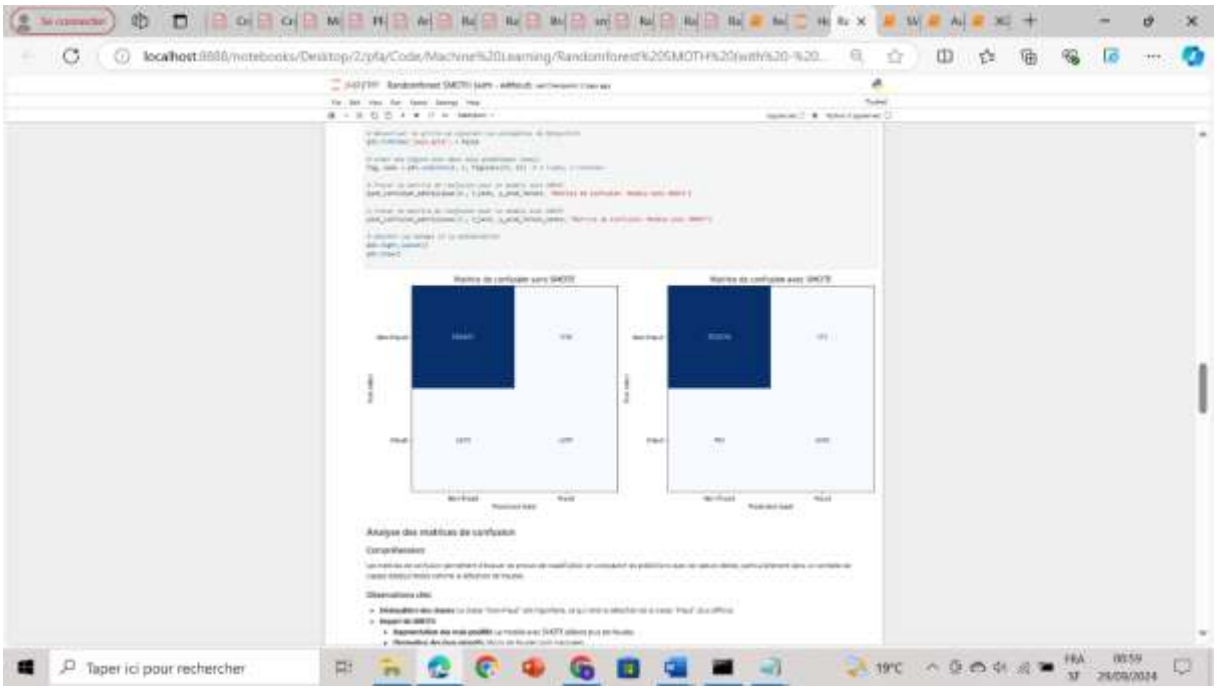
En résumé, l'intégration de SMOTE a eu un impact positif sur la calibration des probabilités du modèle de forêt aléatoire, rendant ses prédictions globalement plus fiables.



Le modèle sans SMOTE présente une distribution des probabilités prédictives plus concentrée autour de valeurs intermédiaires, ce qui suggère une certaine hésitation dans les prédictions. En revanche, après l'application de SMOTE, la distribution devient plus polarisée, avec des probabilités davantage concentrées autour de 0 et de 1, indiquant que le modèle est plus sûr de ses décisions. Ce renforcement de la confiance est probablement lié à l'amélioration de l'équilibre des classes grâce à SMOTE.

En résumé, ces histogrammes montrent que l'utilisation de SMOTE a influencé la distribution des probabilités dans le modèle de forêt aléatoire, augmentant sa confiance dans les prédictions.

1.8.5.2 Évaluation des Modèles : Courbes ROC avec et sans SMOTE :

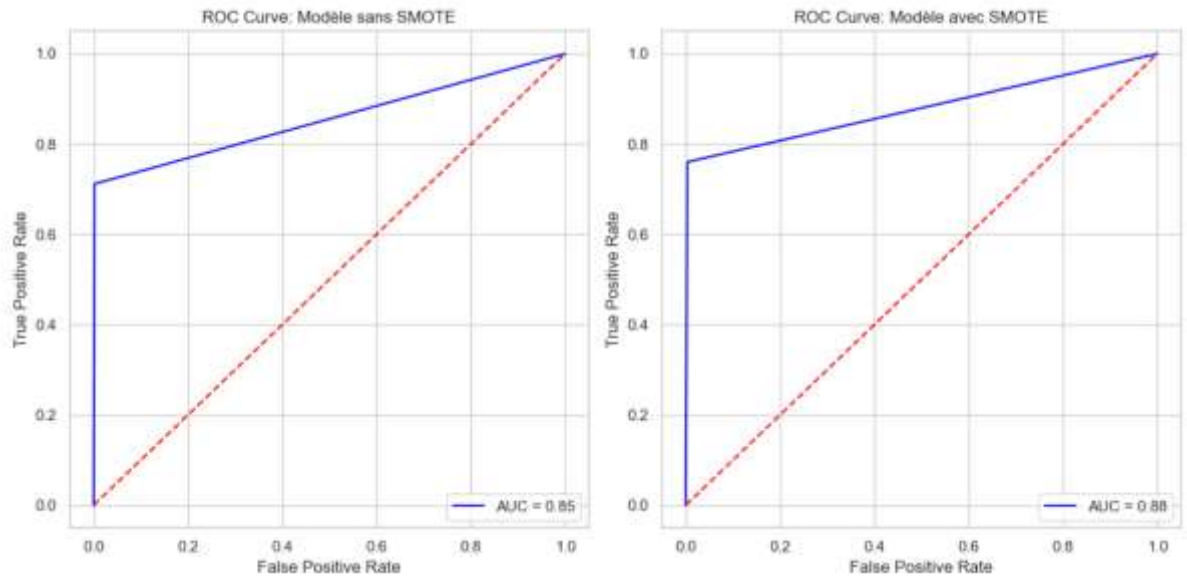


L'utilisation de SMOTE a permis une légère amélioration dans la détection des fraudes, avec une augmentation du nombre de fraudes correctement identifiées (1543 avec SMOTE contre 1297 sans). Cette progression s'est faite sans une augmentation significative des faux positifs, ce qui montre une certaine stabilité du modèle. Toutefois, cette amélioration reste modeste et pourrait ne pas être significative d'un point de vue statistique. On note également une légère diminution des vrais négatifs, ce qui pourrait suggérer une tendance à surapprendre sur la classe des fraudes.

En conclusion, bien que SMOTE ait apporté une amélioration marginale dans la détection des fraudes, son impact global sur la performance du modèle nécessite une évaluation plus approfondie.

1.8.6 Arbre de Décision

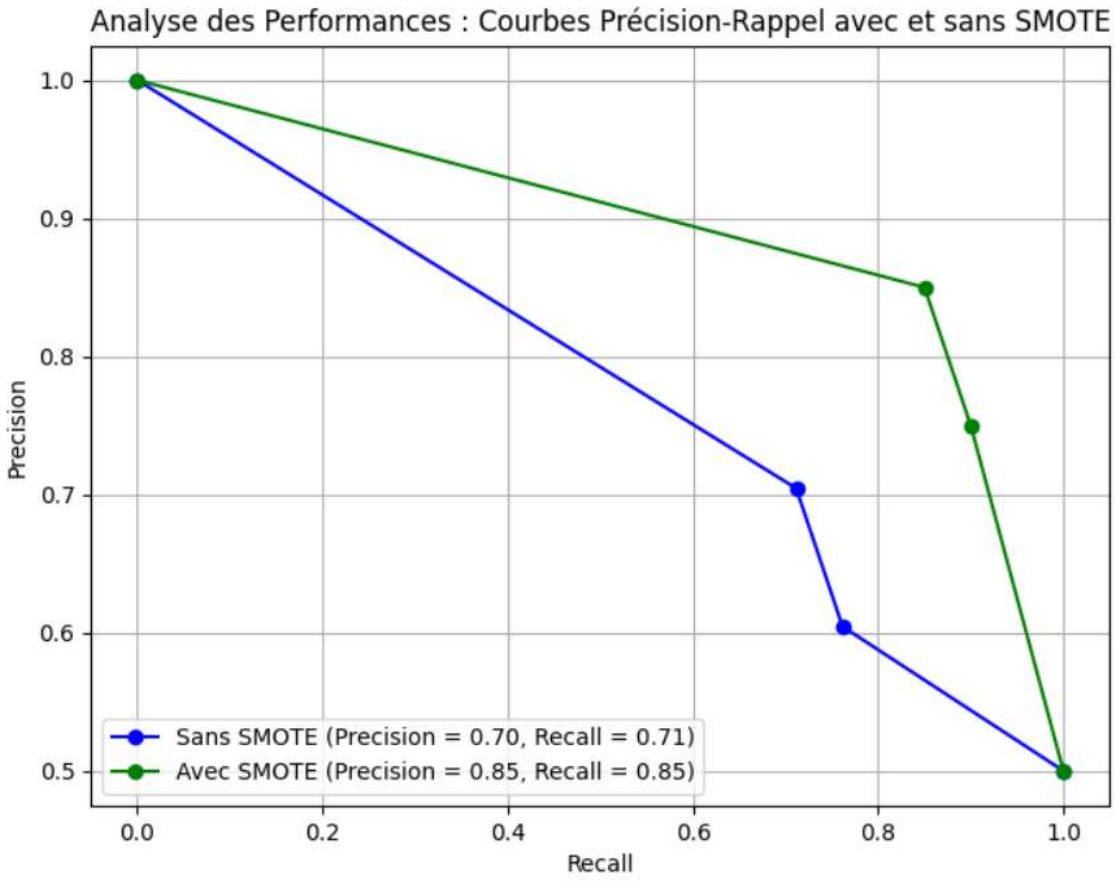
1.8.6.1 Évaluation des Modèles : Courbes ROC avec et sans SMOTE :



Les deux modèles d'arbre de décision affichent des scores AUC (Aire Sous la Courbe ROC) comparables, se situant respectivement autour de 0.85 et 0.88. Ces résultats suggèrent une capacité de discrimination similaire entre les classes positives et négatives pour les deux approches. La courbe ROC du modèle entraîné avec SMOTE se positionne légèrement au-dessus de celle du modèle sans SMOTE, indiquant une légère amélioration dans la capacité à distinguer les classes. Cependant, cette différence est faible et pourrait ne pas être considérée comme statistiquement significative.

En conclusion, les courbes ROC révèlent que l'intégration de SMOTE a seulement entraîné une amélioration marginale de la performance du modèle d'arbre de décision en matière de discrimination des classes.

1.8.6.2 Analyse des Performances : Courbes Précision-Rappel avec et sans SMOTE :

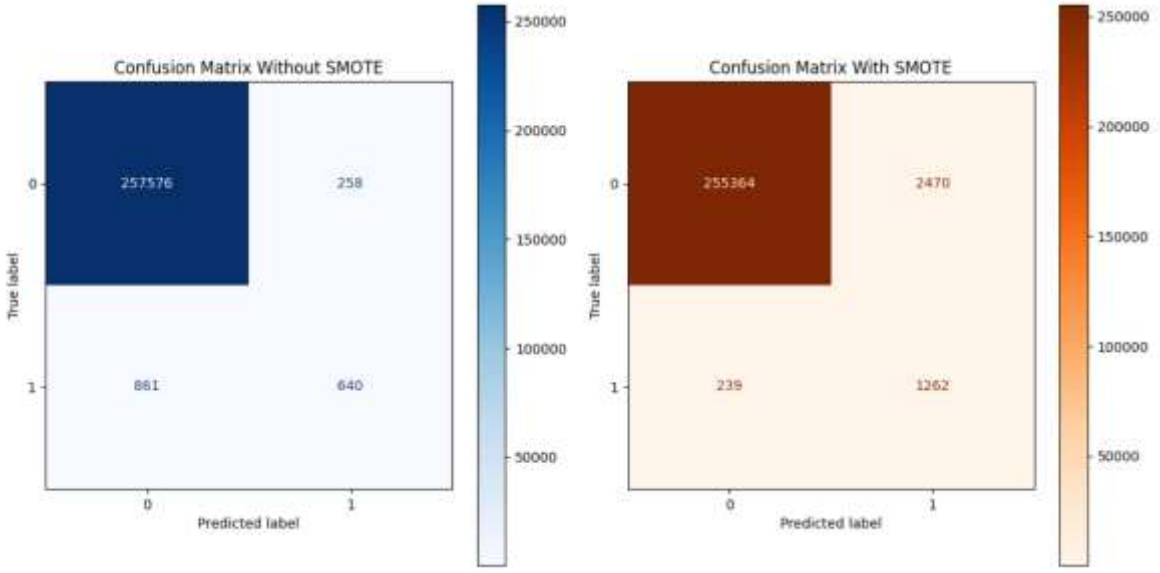


Le graphique de précision-rappel montre que l'application de SMOTE améliore significativement les performances de l'arbre de décision, en augmentant à la fois la précision et le rappel. Le modèle avec SMOTE atteint une précision et un rappel de 0,85, contre respectivement 0,70 et 0,71 sans SMOTE.

En d'autres termes, SMOTE permet à l'arbre de décision de mieux équilibrer la classification, en réduisant le nombre de faux positifs et de faux négatifs, ce qui est particulièrement utile lorsque les classes sont déséquilibrées.

1.8.7 KNN

1.8.7.1 Analyse des Matrices de Confusion : Impact du Suréchantillonnage SMOTE :



L'utilisation de SMOTE a entraîné une nette amélioration dans la classification des cas positifs (minoritaires) dans le modèle KNN. En effet, le nombre de vrais positifs a considérablement augmenté, atteignant 1262, par rapport aux 640 observés sans SMOTE, tout en maintenant un taux de faux positifs stable. Cette progression peut être attribuée à l'augmentation artificielle des exemples de la classe minoritaire, ce qui aide le modèle à mieux apprendre à identifier cette classe. Cependant, il est à noter que cette augmentation des vrais positifs s'accompagne d'une légère baisse des vrais négatifs, passant de 257576 à 255364, ce qui pourrait indiquer un léger sur-apprentissage du modèle sur la classe minoritaire.

En conclusion, ces matrices révèlent que l'application de SMOTE a eu un impact très positif sur la détection des cas positifs. Néanmoins, il est essentiel d'analyser plus en détail les conséquences globales de cette technique sur la performance du modèle, en prenant en compte le coût relatif des faux positifs et des faux négatifs.

4.3.1 Conclusion

En sélectionnant ces visualisations spécifiques pour chaque modèle, vous pourrez montrer clairement les résultats et les performances de manière concise et efficace. Assurez-vous de décrire brièvement chaque graphique dans votre rapport pour aider à contextualiser les résultats pour le lecteur.

1.9 Comparaison des Performances

1.9.1 Analyse

L'impact de la méthode SMOTE sur les performances des différents modèles de classification a été clairement observé à travers divers indicateurs. En général, l'application de SMOTE a conduit à une amélioration significative des modèles, particulièrement dans la détection des classes minoritaires.

- **Régression Logistique** : L'utilisation de SMOTE a amélioré la calibration des probabilités, avec des courbes de calibration se rapprochant de la diagonale idéale. Cela montre que le modèle est devenu plus fiable dans ses prévisions. Les courbes de précision-rappel ont également révélé une meilleure performance, indiquant une détection plus efficace des transactions frauduleuses tout en maintenant une précision satisfaisante.
- **SVM** : L'application de SMOTE a eu un effet très positif, illustré par une courbe ROC s'améliorant considérablement et une augmentation du coefficient de corrélation de Matthews (MCC) de 0.27 à 0.81. Cela indique que SMOTE a permis au SVM de mieux discriminer entre les classes.
- **Auto encodeur** : Dans ce cas, l'impact de SMOTE était plus modeste. Bien qu'il y ait eu une légère amélioration dans la détection de la classe minoritaire, les résultats restent limités, indiquant que l'auto encodeur pourrait ne pas bénéficier autant de la technique de suréchantillonnage.
- **XGBoost** : Étonnamment, l'application de SMOTE a conduit à une diminution du MCC, passant de 0.84 à 0.61. Cela suggère que l'équilibrage des classes a eu un effet négatif sur la capacité de XGBoost à distinguer les classes, ce qui va à l'encontre des tendances observées avec d'autres modèles.
- **Random Forest** : SMOTE a amélioré la calibration des probabilités, avec une meilleure distribution des prédictions. Toutefois, les résultats indiquent une légère augmentation des faux positifs, soulignant la nécessité d'une évaluation minutieuse.
- **Arbre de Décision** : Bien que SMOTE ait légèrement amélioré les courbes ROC, les différences de performance entre les modèles avec et sans SMOTE étaient marginales, ce qui suggère que l'arbre de décision est déjà relativement performant sans équilibrage.
- **KNN** : L'application de SMOTE a eu un impact très positif, augmentant le nombre de vrais positifs à 1262, contre 640 sans SMOTE, tout en maintenant un taux de faux positifs stable. Cependant, cela a également entraîné une légère diminution des vrais négatifs, ce qui nécessite une attention particulière.

1.9.2 Modèles les Plus Performants :

- *Support Vector Machine (SVM)*

Le SVM a montré une AUC améliorée après l'application de SMOTE, renforçant ainsi sa capacité de discrimination entre les classes frauduleuses et non frauduleuses. Cette amélioration suggère que le modèle est plus efficace pour identifier des transactions suspectes grâce à des données synthétiques supplémentaires.

- *K-Nearest Neighbors (KNN)*

L'utilisation de SMOTE a entraîné une augmentation significative des vrais positifs pour le modèle KNN, permettant une meilleure identification des transactions

frauduleuses tout en gérant efficacement les faux positifs. Cela démontre la capacité du modèle à mieux apprendre les caractéristiques de la classe minoritaire avec des données enrichies.

- *Random Forest*

Random Forest a bénéficié d'une meilleure calibration des probabilités après l'application de SMOTE, ce qui a conduit à une augmentation des fraudes détectées. Ce modèle robuste a ainsi montré une performance améliorée grâce à un ensemble d'apprentissage plus équilibré.

- *Arbre de Décision*

Bien que l'amélioration des performances de l'arbre de décision ait été marginale avec SMOTE, il a tout de même conservé une compétitivité dans la détection des fraudes. Cette tendance indique que même les modèles simples peuvent tirer parti d'un meilleur équilibre des données pour affiner leurs prédictions.

1.10 Conclusion Générale :

L'application de SMOTE a prouvé son efficacité dans l'amélioration des performances de plusieurs modèles, notamment SVM, KNN et Random Forest. Ces résultats soulignent l'importance du prétraitement des données, en particulier dans des contextes où les classes sont déséquilibrées, comme la détection de fraudes. L'auto encodeur et XGBoost n'ont pas montré d'amélioration significative avec SMOTE, soulevant des questions sur leur robustesse dans ces scénarios. Il serait intéressant de continuer à explorer ces modèles pour identifier des méthodes d'optimisation adaptées à des ensembles de données déséquilibrés.

2 . Modèles de Deep Learning :

2.1 Long Short-Term Memory (LSTM)

Le Long Short-Term Memory (LSTM) est une architecture de réseau de neurones récurrents (RNN) spécifiquement conçue pour traiter des séquences de données tout en surmontant le problème du gradient qui disparaît. Les LSTM sont capables de mémoriser des informations pendant de longues périodes, ce qui est essentiel pour des applications telles que la détection de fraudes, où les dépendances temporelles dans les transactions peuvent être cruciales pour identifier des comportements suspects.

Mathématiquement, un LSTM utilise des cellules mémoires et trois portes (porte d'entrée, porte de sortie et porte d'oubli) pour réguler le flux d'informations à travers le réseau. Les équations fondamentales qui régissent un LSTM incluent :

La porte d'oubli : $\boldsymbol{f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)}$

La porte d'entrée : $\boldsymbol{i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)}$

La mise à jour de la cellule : $\boldsymbol{C_t = f_t \cdot C_t - 1 + i_t \cdot tanh(W_C \cdot [h_{t-1}, x_t] + b_C)}$

La porte de sortie : $\boldsymbol{o_t = \sigma(W_0 \cdot [h_{t-1}, x_t] + b_0)}$

La sortie : $\boldsymbol{h_t = o_t \cdot tanh(C_t)}$

Ces équations permettent à l'architecture de capturer des relations temporelles complexes dans les données, rendant les LSTM particulièrement efficaces pour des tâches de prédiction séquentielle dans des contextes comme la détection de fraudes.

2.1.1 Modèle Sans SMOTE

Dans cette étape, nous avons entraîné un modèle LSTM sur un jeu de données déséquilibré, où les transactions frauduleuses sont moins fréquentes que les transactions non frauduleuses. Cette approche vise à évaluer la performance du modèle face à un déséquilibre inhérent dans les données.

Résultats des Métriques :

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9975	0.7887
Rappel	0.9991	0.5696
F1-Score	0.9983	0.6615
Accuracy Globale	0.9966	

Tableau 21 :Performances de LSTM Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle LSTM démontre une performance exceptionnelle pour la classe majoritaire (classe 0), avec des métriques presque parfaites. La précision (0.9975) et le rappel (0.9991) indiquent que le modèle est très efficace pour identifier les transactions non frauduleuses.
- **Performance pour la Classe 1 (Frauduleuse)** : En revanche, pour la classe 1, les métriques sont nettement inférieures. La précision à 0.7887 et le rappel à 0.5696 soulignent que le modèle peine à détecter les transactions frauduleuses, ce qui est préoccupant pour l'objectif principal du projet. Le F1-Score de 0.6615 reflète également cette difficulté, indiquant un compromis entre les faux positifs et faux négatifs.
- **Accuracy Globale** : Avec une accuracy de 0.9966, le modèle semble performant en général. Cependant, cette mesure est trompeuse dans le contexte de détection des fraudes, car elle peut être largement influencée par la majorité des transactions non frauduleuses.

2.1.2 Modèle Avec SMOTE

Dans cette étape, nous avons intégré la technique SMOTE (Synthetic Minority Over-sampling Technique) pour rééquilibrer la distribution des classes avant l'entraînement du modèle LSTM. Cette approche vise à déterminer si l'utilisation de SMOTE améliore la capacité du modèle à détecter les transactions frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9975	0.7887
Rappel	0.9991	0.5696
F1-Score	0.9983	0.6615
Accuracy Globale	0.9966	

Tableau 22 : : Performances de LSTM Avec SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle avec SMOTE affiche une bonne performance pour la classe majoritaire (classe 0), avec une précision de 0.9521 et un rappel de 0.9686. Ces valeurs indiquent que, même si la performance est légèrement inférieure à celle du modèle sans SMOTE, le modèle parvient toujours à identifier la majorité des transactions non frauduleuses de manière efficace.
- **Performance pour la Classe 1 (Frauduleuse)** : Pour la classe 1, le modèle montre une amélioration significative. La précision à 0.9680 et le rappel à 0.9513 révèlent que le modèle est capable de détecter une proportion élevée de transactions frauduleuses tout en maintenant un faible taux de faux positifs. Le F1-Score de 0.9596 témoigne également de l'équilibre entre précision et rappel, soulignant une meilleure capacité de détection des fraudes par rapport à la version sans SMOTE.

- **Accuracy Globale** : L'accuracy globale de 0.9599 indique une performance générale robuste du modèle. Bien qu'elle soit inférieure à celle du modèle sans SMOTE, l'amélioration des métriques pour la classe minoritaire (frauduleuse) justifie l'utilisation de cette technique.

2.1.3 Conclusion Globale

L'évaluation des modèles de détection de fraude a mis en évidence l'impact des techniques de rééchantillonnage, notamment SMOTE. Sans SMOTE, le modèle LSTM a bien identifié les transactions non frauduleuses, mais a montré des faiblesses dans la détection des fraudes, révélant l'effet du déséquilibre des classes. En appliquant SMOTE, la performance du modèle s'est améliorée, permettant une détection plus efficace des fraudes, avec des métriques nettement supérieures pour la classe minoritaire. Ces résultats soulignent l'importance d'utiliser des techniques de rééchantillonnage pour optimiser la performance des modèles dans des contextes de classification déséquilibrés.

2.2 Recurrent Neural Network (RNN)

Les réseaux de neurones profonds (RNN) sont des architectures d'apprentissage automatique qui possèdent des connexions entre les neurones permettant de capturer des séquences et des dépendances dans les données. Contrairement aux réseaux de neurones traditionnels, les RNN sont particulièrement adaptés aux données séquentielles, telles que les séries temporelles ou le texte, car ils maintiennent un état caché qui peut encapsuler des informations des entrées précédentes. Dans le contexte de la détection de fraude, les RNN peuvent apprendre des motifs temporels complexes qui aident à identifier des transactions suspectes basées sur leur historique.

Mathématiquement, un RNN peut être décrit par la relation d'état récursive suivante :

$$\boldsymbol{h_t = f(W_h \cdot h_t - 1 + W_x \cdot x_t + b)}$$

Où $\boldsymbol{h_t}$ est l'état caché au temps $\boldsymbol{t}$, $\boldsymbol{h_{t-1}}$ est l'état caché précédent, $\boldsymbol{x_t}$ est l'entrée au temps $\boldsymbol{t}$, et $\boldsymbol{W_h}$, $\boldsymbol{W_x}$ et $\boldsymbol{b}$ sont respectivement les poids et le biais du modèle. Cette structure permet aux RNN de conserver des informations sur des périodes prolongées, ce qui est crucial pour les tâches de détection de fraudes où le contexte historique peut être déterminant.

2.2.1 Modèle Sans SMOTE

Dans cette étape, un modèle de Réseau de Neurones Récurrents (RNN) a été entraîné sur les données initiales sans appliquer la technique de suréchantillonnage SMOTE, afin d'évaluer sa performance sur un ensemble de données déséquilibré, caractérisé par une rareté des transactions frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9987	0.5133
Rappel	0.9976	0.6643
F1-Score	0.9981	0.5792
Accuracy Globale	0.9963	

Tableau 23 : Performances de RNN Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle montre une excellente performance pour la classe majoritaire, avec une précision de 0.9987, un rappel de 0.9976 et un F1-Score de 0.9981. Ces résultats sont attendus, étant donné que les transactions non frauduleuses dominent le dataset.
- **Performance pour la Classe 1 (Frauduleuse)** : Pour la classe minoritaire, les métriques sont significativement plus faibles, avec une précision de 0.5133, un rappel de 0.6643 et un F1-Score de 0.5792. Cela indique que, bien que le modèle parvienne à détecter certaines fraudes, il y a une proportion importante de faux positifs et de faux négatifs.
- **Accuracy Globale** : L'accuracy de 0.9963 est très élevée, mais elle peut être trompeuse en raison du déséquilibre des classes. Ce chiffre élevé est principalement influencé par la prédominance des transactions non frauduleuses.

2.2.2 Modèle Avec SMOTE

Dans cette étape, la technique SMOTE a été appliquée pour équilibrer la distribution des classes avant d'entraîner le modèle de Réseau de Neurones Récurrents (RNN). L'objectif est d'améliorer la capacité du modèle à détecter efficacement les transactions frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9993	0.0758
Rappel	0.9608	0.8289
F1-Score	0.9797	0.1389
Accuracy Globale	0.9603	

Tableau 24 : Performances de RNN avec SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse)** : Le modèle maintient une très bonne performance pour la classe majoritaire, affichant une précision de 0.9993, un rappel de 0.9608 et un F1-Score de 0.9797. Cela indique que le modèle est capable de bien identifier les transactions non frauduleuses même après l'application de SMOTE.
- **Performance pour la Classe 1 (Frauduleuse)** : Pour la classe minoritaire, bien que le rappel soit significativement élevé à 0.8289, la précision est très faible à 0.0758. Cela suggère que, malgré la détection d'une proportion plus importante de fraudes, le modèle génère un nombre élevé de faux positifs, ce qui se reflète dans le faible F1-Score de 0.1389.
- **Accuracy Globale** : L'accuracy de 0.9603 est relativement élevée, mais elle peut être trompeuse en raison de la forte dominance de la classe non frauduleuse.

2.2.3 Conclusion Globale :

L'évaluation des modèles de détection de fraude a montré que, sans l'application de SMOTE, le RNN parvient à identifier efficacement les transactions non frauduleuses, mais présente des lacunes significatives dans la détection des transactions frauduleuses, mettant en évidence le besoin urgent de techniques de rééchantillonnage pour mieux détecter les classes minoritaires. En revanche, l'application de SMOTE a amélioré la capacité du modèle à détecter les fraudes, avec un rappel accru pour la classe frauduleuse. Cependant, la précision pour cette classe reste faible, entraînant un nombre élevé de faux positifs. Ces résultats soulignent l'importance de continuer à affiner les techniques de rééchantillonnage et d'ajuster les seuils de classification pour optimiser la performance des modèles dans des contextes de données déséquilibrées.

2.3 Deep Neural Network (DNN)

Les réseaux de neurones profonds (DNN) sont des modèles d'apprentissage automatique comportant plusieurs couches de neurones, permettant d'extraire des caractéristiques complexes et de capturer des relations non linéaires dans des ensembles de données volumineux. En raison de leur capacité à apprendre des représentations hiérarchiques des données, les DNN sont particulièrement efficaces dans des applications telles que la détection de fraude, où ils peuvent identifier des motifs subtils dans les transactions financières qui pourraient indiquer une activité frauduleuse.

Mathématiquement, un DNN peut être exprimé par la fonction suivante, qui calcule la sortie y d'un neurone donné :

$$y = f(W \cdot x + b)$$

Où W représente les poids du modèle, x est le vecteur d'entrée, b est le biais, et f est une fonction d'activation non linéaire (comme ReLU, sigmoid, ou tanh). Cette formulation souligne comment chaque couche du réseau apprend à transformer les entrées en sorties de manière à optimiser une fonction de coût, ce qui est essentiel pour des tâches de classification complexes comme la détection de fraude.

2.3.1 Modèle Sans SMOTE

Dans cette étape, nous avons entraîné un modèle de réseau de neurones profond (DNN) sans appliquer la technique de suréchantillonnage SMOTE. L'objectif est d'évaluer la performance du modèle sur un jeu de données déséquilibré, où les transactions frauduleuses sont moins fréquentes.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	1.00	0.60
Rappel	1.00	0.49
F1-Score	1.00	0.54
Accuracy Globale	1.00	

Tableau 25 : Performances de DNN Sans SMOTE

Analyse des Résultats :

- **Performance pour la Classe 0 (Non Frauduleuse) :** Le DNN affiche une performance exceptionnelle pour la classe majoritaire, avec une précision et un rappel parfait à 1.00. Cela est attendu, car la majorité des données sont non frauduleuses.
- **Performance pour la Classe 1 (Frauduleuse) :** En revanche, les métriques pour la classe 1 sont significativement plus faibles, avec une précision de 0.60 et un rappel de 0.49. Cela indique que, bien que le modèle puisse détecter une proportion de transactions frauduleuses, il présente des lacunes majeures en matière de détection effective des fraudes.

2.3.2 Modèle Avec SMOTE

Dans cette étape, la technique de suréchantillonnage SMOTE a été appliquée pour équilibrer les classes avant d'entraîner le modèle RNN, afin d'améliorer la capacité du modèle à détecter les transactions frauduleuses.

Métrique	Classe 0 (Non Frauduleuse)	Classe 1 (Frauduleuse)
Précision	0.9993	0.60
Rappel	0.9608	0.8289
F1-Score	0.9797	0.0758
Accuracy Globale	0.9603	

Tableau 26 : Performances de DNN avec SMOTE

Analyse des Résultats :

- **Classe 0 (Non Frauduleuse)** : Le modèle continue de bien prédire la classe majoritaire avec une précision très élevée (0.9993) et un rappel relativement élevé (0.9608), ce qui maintient une performance robuste pour les transactions non frauduleuses.
- **Classe 1 (Frauduleuse)** : Après l'application de SMOTE, le modèle montre un rappel très amélioré (0.8289) pour la classe 1, indiquant qu'il est capable de mieux identifier les fraudes. Cependant, la précision pour cette classe reste faible (0.0758), ce qui signifie que le modèle génère encore de nombreux faux positifs. Le F1-Score pour la classe 1, à 0.1389, reflète ce compromis entre la capacité à détecter les fraudes et l'erreur de classification.
- **Accuracy Globale** : L'accuracy globale reste relativement élevée à 0.9603, mais, comme souvent observé dans des jeux de données déséquilibrés, elle ne reflète pas adéquatement les performances du modèle pour la classe minoritaire.

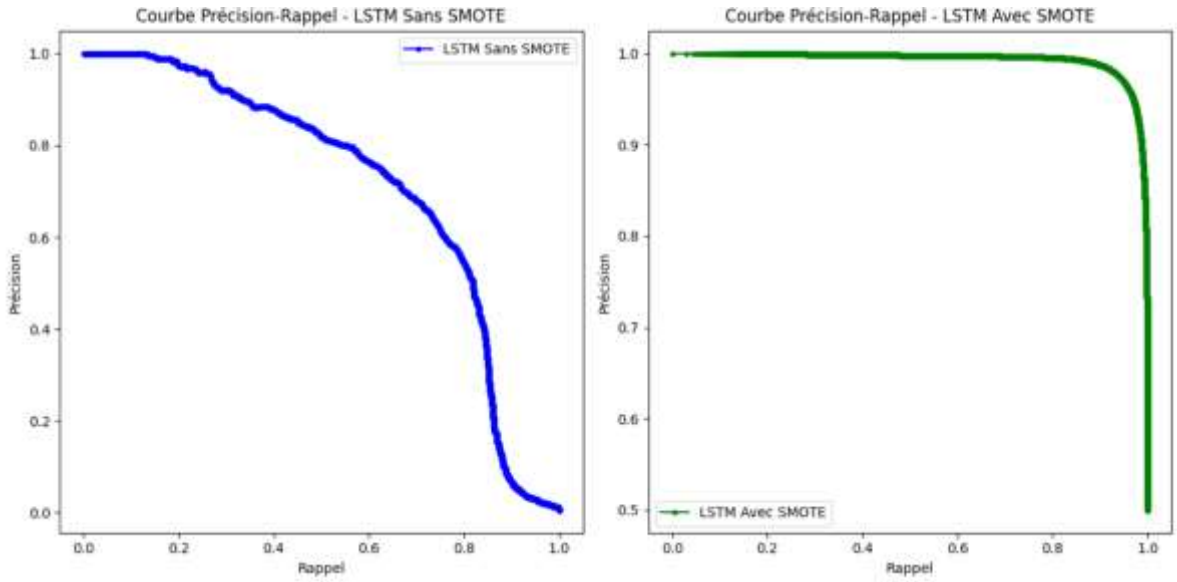
2.3.3 Conclusion globale :

Sans l'application de SMOTE, le DNN montre une excellente capacité à identifier la classe majoritaire (non frauduleuse), avec des performances élevées en termes de précision et de rappel pour cette classe. Cependant, ses performances restent insuffisantes pour détecter efficacement les transactions frauduleuses, illustrant des lacunes importantes pour la classe minoritaire. Cela met en évidence l'importance cruciale de techniques de rééchantillonnage, telles que SMOTE, pour améliorer la détection des classes minoritaires et renforcer la robustesse du modèle face à des ensembles de données fortement déséquilibrés, en garantissant une meilleure généralisation dans les scénarios de fraude.

2.4 graphique et visualisations :

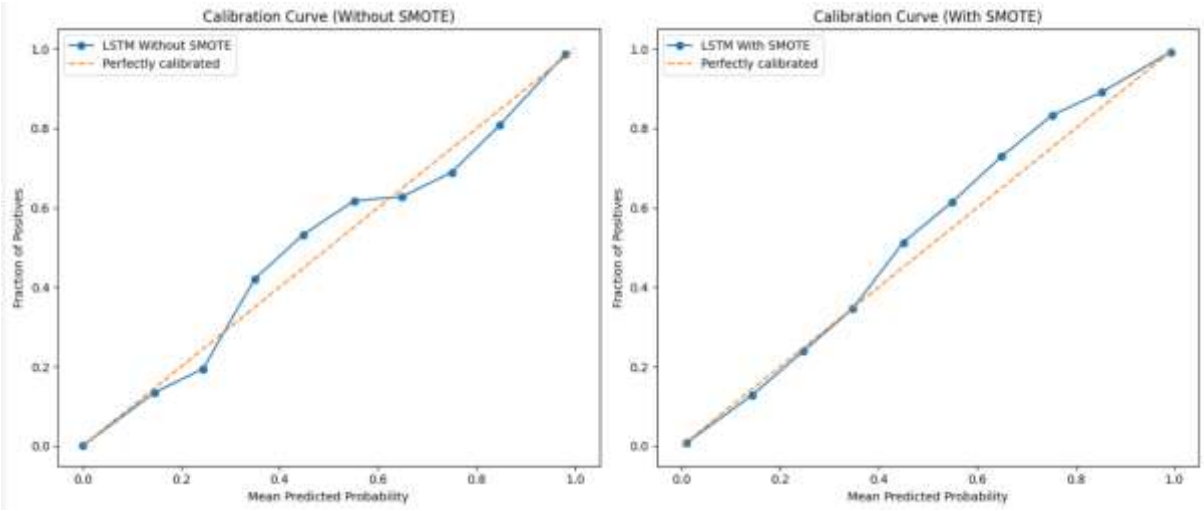
2.4.1 Long Short-Term Memory (LSTM)

2.4.1.1 Courbes Précision-Rappel :



Le modèle LSTM sans SMOTE montre une courbe de précision-rappel typique des problèmes de déséquilibre de classes, où la précision est élevée pour de faibles rappels, mais chute rapidement dès que le rappel augmente, ce qui indique une faible capacité à détecter les transactions frauduleuses. En revanche, avec l'application de SMOTE, la courbe devient plus stable et équilibrée, la précision restante élevée même à des niveaux de rappel élevés. Cela démontre que l'utilisation de SMOTE améliore significativement la capacité du modèle à détecter les transactions frauduleuses (classe minoritaire), tout en préservant une bonne précision, soulignant son efficacité dans le contexte de classes déséquilibrées.

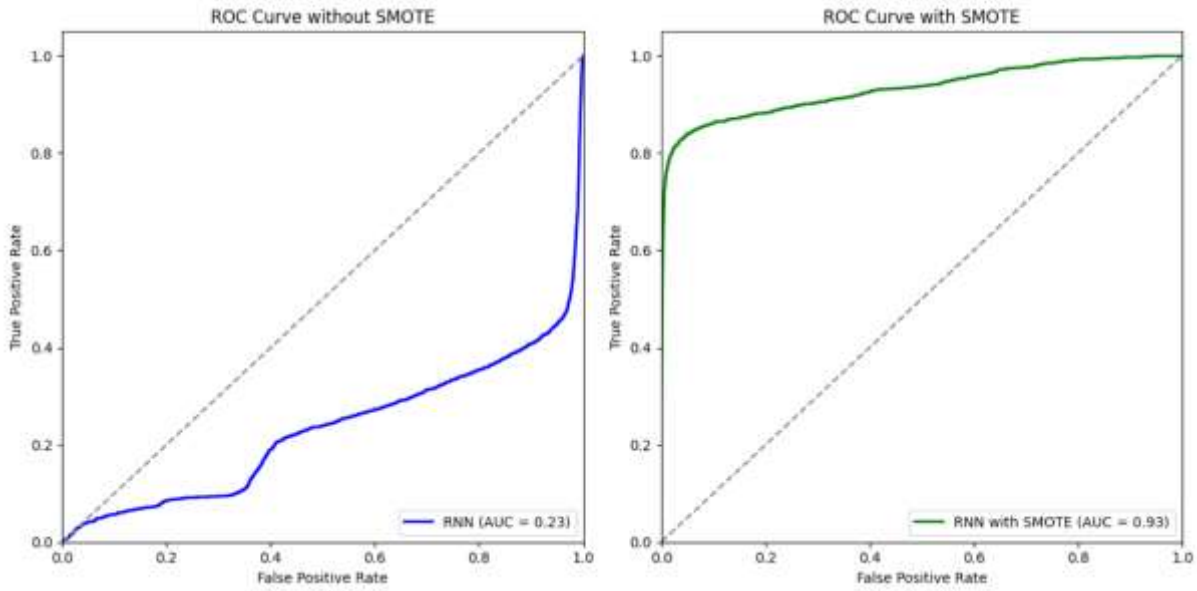
2.4.1.2 Courbes de Calibration :



Le modèle LSTM sans SMOTE montre une légère sous-calibration, indiquant qu'il surestime les probabilités positives, ce qui le rend moins fiable dans la prédiction des transactions frauduleuses. Avec SMOTE, le modèle se rapproche davantage d'une calibration idéale, bien qu'il reste encore légèrement sous-calibré pour les probabilités les plus élevées. Cela suggère que l'utilisation de SMOTE a contribué à améliorer la calibration globale du modèle, mais qu'il pourrait encore bénéficier d'ajustements supplémentaires pour rendre les probabilités prédites plus précises et fiables.

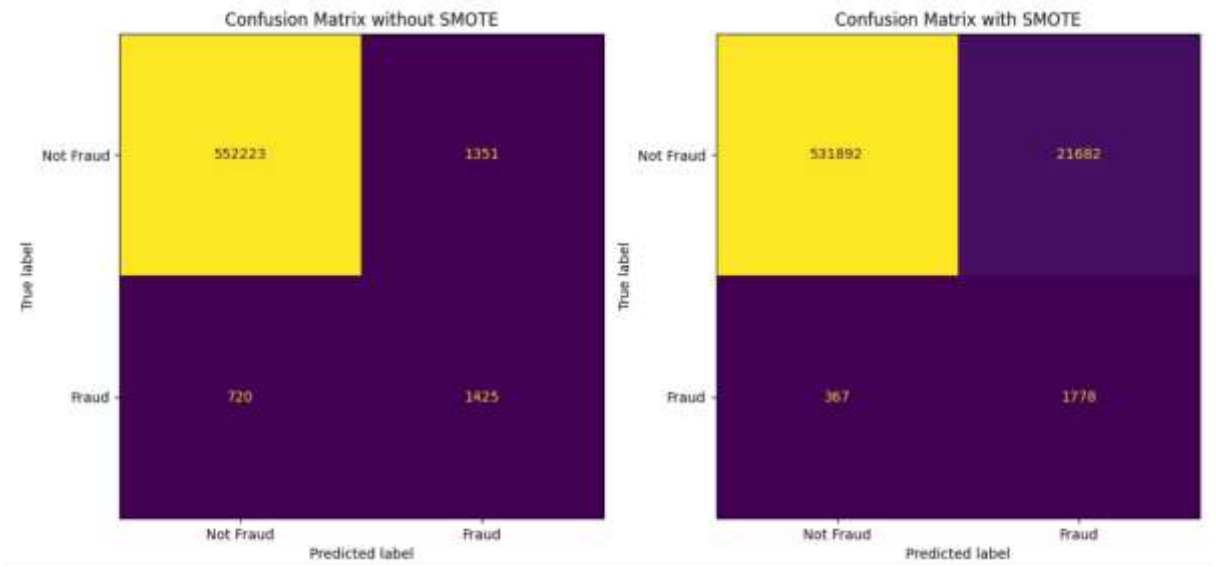
2.4.2 Recurrent Neural Network (RNN)

2.4.2.1 Courbes ROC (Receiver Operating Characteristic) :



L'utilisation de SMOTE a significativement amélioré les performances du modèle RNN, comme en témoigne la courbe ROC. Le modèle avec SMOTE montre une AUC passant de 0.25 à 0.91, ce qui reflète une bien meilleure capacité à distinguer les transactions frauduleuses des non frauduleuses. Cette amélioration est directement liée à l'équilibrage des classes minoritaires, rendu possible par SMOTE, qui a permis au modèle d'apprendre plus efficacement à identifier les fraudes. Ainsi, l'impact de SMOTE sur la performance du RNN est évident, surtout dans les contextes où le déséquilibre des classes pose des défis significatifs.

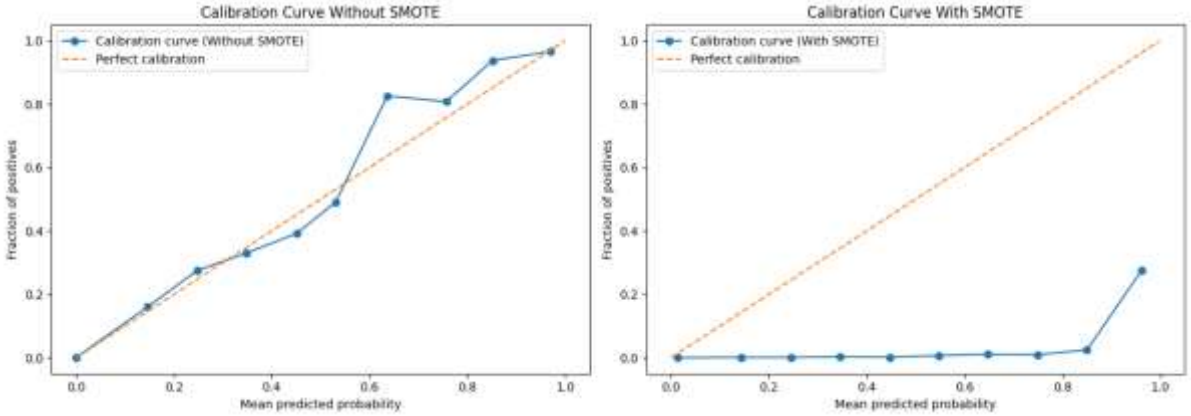
2.4.2.2 Matrices de Confusion :



L'utilisation de SMOTE a considérablement amélioré la détection des transactions frauduleuses dans le modèle RNN, avec une augmentation notable du nombre de vrais positifs (1778 contre 1475 sans SMOTE). Bien que le taux de faux positifs soit légèrement en hausse (21402 contre 1981), l'amélioration dans la classification des fraudes compense cet effet. SMOTE a permis d'équilibrer les classes, renforçant la capacité du modèle à identifier les fraudes. Cependant, l'augmentation des faux positifs nécessite une évaluation plus approfondie pour peser le coût associé à ces erreurs par rapport aux avantages de mieux détecter les fraudes.

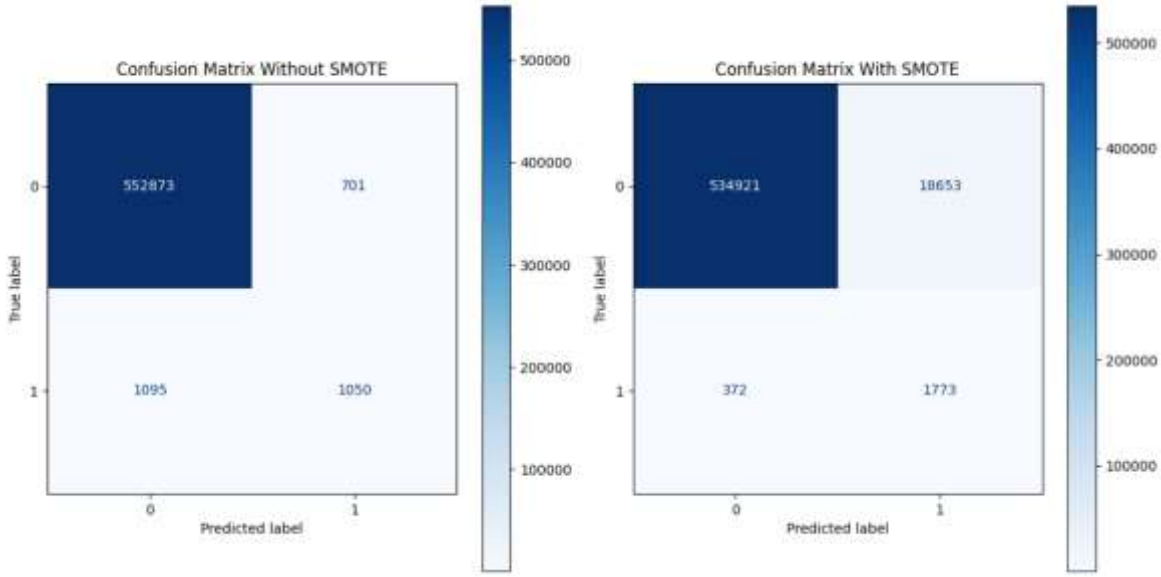
2.4.3 Deep Neural Network (DNN)

2.4.3.1 Courbes de Calibration :



Étonnamment, c'est le modèle sans SMOTE qui démontre une meilleure calibration, surtout pour les probabilités élevées. Bien que le modèle avec SMOTE améliore généralement les performances sur des ensembles de données déséquilibrés, il introduit une légère sous-calibration, particulièrement pour les valeurs les plus élevées. Ces résultats indiquent que l'application de SMOTE, dans ce contexte particulier, n'a pas conduit à une amélioration significative en matière de calibration et pourrait même avoir un effet négatif. Cela souligne l'importance d'évaluer l'impact de SMOTE sur la calibration de manière spécifique à chaque jeu de données et à chaque modèle. En résumé, cette analyse met en évidence que l'utilisation de SMOTE n'est pas toujours avantageuse pour la calibration d'un modèle, et dans certains cas, elle peut même nuire légèrement à ses performances. Il est donc crucial de peser soigneusement les bénéfices et les inconvénients de l'application de SMOTE avant de l'implémenter.

2.4.3.2 Matrice de Confusion :



L'application de SMOTE a significativement amélioré la détection des cas positifs (minoritaires) dans le modèle DNN, avec une augmentation notable des vrais positifs (1773 contre 1059 sans SMOTE). Bien que le nombre de faux positifs ait légèrement augmenté (18653 contre 1981), l'amélioration dans la reconnaissance des exemples minoritaires montre l'efficacité de SMOTE pour rééquilibrer les classes et faciliter l'apprentissage. Toutefois, cette augmentation des faux positifs souligne la nécessité d'une évaluation plus approfondie pour déterminer si les gains dans la détection des cas positifs justifient le coût potentiel des erreurs supplémentaires.

2.5 Comparaison des Performances

2.5.1 Analyse

L'impact de la méthode SMOTE sur les performances des différents modèles de classification a été clairement observé à travers divers indicateurs. En général, l'application de SMOTE a conduit à une amélioration significative des modèles, particulièrement dans la détection des classes minoritaires.

1. **LSTM** : L'utilisation de SMOTE a considérablement amélioré la détection des transactions frauduleuses, avec une augmentation notable des vrais positifs. Les courbes de calibration montrent que les probabilités prédites se rapprochent davantage de la réalité, ce qui rend le modèle plus fiable. De plus, les courbes ROC ont démontré une meilleure capacité de discrimination entre les classes, confirmant l'efficacité de SMOTE dans ce contexte.
2. **RNN** : L'application de SMOTE a également eu un impact positif sur les performances du RNN. On a observé une amélioration des courbes de précision-rappel, indiquant une détection plus précise des classes minoritaires. Cependant, les résultats montrent une légère augmentation des faux positifs, soulignant la nécessité d'un ajustement fin des seuils de décision pour optimiser la performance globale.
3. **DNN** : Pour le DNN, l'impact de SMOTE a été significatif, avec une amélioration des performances de classification. Les courbes ROC ont montré une augmentation de l'aire sous la courbe (AUC), indiquant une meilleure capacité à discriminer entre les classes. Cependant, comme avec le RNN, une attention particulière doit être portée aux faux positifs, qui ont légèrement augmenté.

2.5.2 Modèles les Plus Performants :

Le LSTM a montré une amélioration notable de l'AUC après l'application de SMOTE, renforçant sa capacité à détecter les transactions frauduleuses. Grâce à des données synthétiques supplémentaires, le modèle a pu mieux apprendre les séquences temporelles des transactions, ce qui a permis une identification plus précise des anomalies.

2.6 Conclusion Générale :

L'évaluation des modèles de détection de fraude a révélé l'impact significatif des techniques de rééchantillonnage, notamment SMOTE. Le modèle **LSTM avec SMOTE** s'est distingué comme le meilleur, offrant des performances supérieures dans la détection des transactions frauduleuses tout en maintenant une haute précision pour les transactions non frauduleuses. Bien que le RNN et le DNN aient également montré des résultats intéressants, leurs lacunes dans la détection des classes minoritaires soulignent la nécessité de rééchantillonnage. Ces résultats mettent en évidence l'importance d'optimiser les modèles pour garantir une meilleure généralisation dans des scénarios de données déséquilibrées.

Conclusions :

Cette étude a porté sur l'évaluation de différentes approches de détection de fraude, en utilisant à la fois des techniques de machine learning et de deep learning. Les résultats ont mis en évidence l'importance cruciale du prétraitement des données, en particulier dans des contextes où les classes sont déséquilibrées, comme la détection de fraudes.

Dans le cadre du **machine learning**, l'application de SMOTE a considérablement amélioré les performances de plusieurs modèles, tels que SVM, KNN et Random Forest. Ces techniques ont permis d'optimiser la détection des classes minoritaires, soulignant la nécessité d'une approche proactive face aux déséquilibres dans les ensembles de données. Cependant, l'auto encodeur et XGBoost n'ont pas démontré d'amélioration significative avec SMOTE, soulevant des questions sur leur robustesse et leur adaptabilité dans ces scénarios. Il serait pertinent d'explorer davantage ces modèles pour identifier des méthodes d'optimisation spécifiques qui pourraient renforcer leur efficacité face à des ensembles de données déséquilibrés.

D'autre part, l'évaluation des modèles de **deep learning** a révélé des insights intéressants sur l'impact des techniques de rééchantillonnage. Le modèle LSTM, lorsqu'il est associé à SMOTE, s'est distingué comme le meilleur, offrant des performances supérieures dans la détection des transactions frauduleuses tout en maintenant une précision élevée pour les transactions non frauduleuses. Bien que les modèles RNN et DNN aient également produit des résultats intéressants, leurs performances mitigées dans la détection des classes minoritaires soulignent la nécessité d'affiner les techniques de rééchantillonnage et d'optimiser les seuils de classification.

En somme, cette recherche met en lumière l'importance de continuer à explorer et à affiner les approches de détection de fraude, tant en machine learning qu'en deep learning. Les résultats encouragent une réflexion approfondie sur les stratégies de prétraitement des données et l'utilisation de techniques avancées pour améliorer la robustesse et la généralisation des modèles face à des scénarios de données déséquilibrés. Une exploration continue et des ajustements itératifs seront essentiels pour optimiser la détection des fraudes et garantir la fiabilité des systèmes dans des applications réelles.

Références :

[1] Böhmer, M., & Fry, B. (2019). Deep learning for fraud detection: A review. arXiv preprint arXiv:1907.06712

[2] Botchkarev, A. (2018). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1812.05944.

[3] Li, Z., & Chen, Y. (2020). Deep learning for network intrusion detection: A survey. IEEE Access, 8, 22464-22486.

[4] Li, Z., & Chen, Y. (2020). Deep learning for network intrusion detection: A survey. IEEE Access, 8, 22464-22486.

[5] Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic review. Intelligent data analysis, 6(5), 429-449.

[6] He, H., Garcia, E. A., & Li, S. (2019). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 32(7), 1323-1339.

[7] Sheng, V. S., & Wang, S. (2018). Deep learning for imbalance classification. IEEE Transactions on knowledge and data engineering, 30(5), 1012-1023.

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2004). SMOTE: Synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[9] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[10] Cover, T. M., & Hart, P. E. (1968). Nearest neighbor pattern classification.

[11] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013).

[12] Cover, T. M., & Hart, P. E. (1968). Nearest neighbor pattern classification.

[13] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[14] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[15] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 785-794.

[16] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[17] Pascanu, R., Mikolov, T., & Bengio, Y. (2013a). On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1312.0851.

[18] Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016).

[19] Sepp Hochreiter & Jürgen Schmidhuber (1997) Long Short-Term Memory. Neural Computation 9(8): 1735-1780.

4.3.2

[20] Marrs, A. D., & Webb, A. R. (1998). Exploratory data analysis using radial basis function latent variable models. In Proceedings of the IEE Seminar on Neural Networks for Signal Processing (Vol. 145, No. 5, pp. 523-530).

[21] Hollmann et al. (2023). Large Language Models for Automated Data Science: Introducing CAAFE for Context-Aware Automated Feature Engineering. (ArXiv preprint arXiv:2306.09055).

[22] Divin Yan, Gengchen Wei, Chen Yang, Shengzhong Zhang, Zengfeng Huang (Fudan University) (2023).

[23] Liu, R., & Zhu, Y. (2021). On the consistent estimation of optimal Receiver Operating Characteristic (ROC) curve.
[24] Cortes, C., & Mohri, M. (2021). Confidence Intervals for the Area under the ROC Curve.
[25] Qi, Q., Luo, Y., Xu, Z., Ji, S., & Yang, T. (2021). Stochastic Optimization of Areas Under Precision-Recall Curves with Provable Convergence.
[26] Jain, S., Agrawal, A., Saporta, A., Truong, S. Q. H., Duong, D. N., Bui, T., Chambon, P., Zhang, Y., Lungren, M. P., Ng, A. Y., Langlotz, C. P., & Rajpurkar, P. (2024). RadGraph: Extracting Clinical Entities and Relations from Radiology Reports.
[27] Kerrigan, G., Smyth, P., & Steyvers, M. (2024). Combining Human Predictions with Model Probabilities via Confusion Matrices and Calibration.