UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Master 2 Pro - Traitement de l'Information et Data Science en Entreprise
(TIDE)

Travaux d'atelier

Sujet : Système de détection de fraude basé sur l'apprentissage automatique

Préparé par : Davidson ADRIEN

Tuteur pédagogique : Joseph RYNKIEWICZ

Promotion 2024 - 2025

Octobre 2025

Table of Contents

Sigles et acronymes

ADF	Augmented Dickey Fuller
ARIMA	Auto-Regressive Integrated Moving Average
LR	Logistic Regression
MAE	Mean Absolute Error
PR AUC	Precision/Recall Area Under the Curve
RF	Random Forest
RMSE	Root Mean Squared Error

Figures et tableaux

Résumé

Ce projet a pour objectif de développer un système de détection automatique de fraudes à partir de données de transactions financières, dans le cadre d'un atelier de formation en Data Science à l'Université Paris 1 Panthéon-Sorbonne. L'analyse exploratoire a débuté par une étude des séries temporelles du nombre de fraudes, révélant des composantes saisonnières et une tendance globale stable, validées par des tests statistiques tels que le test d'ADF. Un modèles ARIMA a été testé afin de prédire l'évolution des fraudes dans le temps.

Par la suite, l'analyse univariée, bivariée et multivariée a permis d'identifier des patterns anormaux pour certaines variables (notamment les balances et les montants) selon le statut de fraude. Des transformations ont été appliquées et deux nouvelles variables ont été créées pour mieux capter les différences entre montants et soldes.

Deux modèles de classification ont été développés : une régression logistique pénalisée et une Random Forest, cette dernière ayant fourni les meilleurs résultats, avec un recall de 96 %, une précision de 79 %, une log loss très faible (0.00096) et un Cohen's Kappa de 0.86.

Enfin, le modèle Random Forest a été intégré dans une application Streamlit permettant une prédiction en temps réel des transactions, avec des logiques métiers complémentaires pour renforcer l'interprétation des résultats et l'aide à la décision.

Mots-clés: Détection de fraude, Analyse de série temporelle, Classification Random Forest, Déséquilibre de classe, Déploiement application Streamlit

Abstract

This project aims to develop an automated fraud detection system based on financial transaction data, as part of a data science workshop at Université Paris 1 Panthéon-Sorbonne. The exploratory analysis began with a time series analysis of fraudulent transactions, revealing seasonal patterns and a relatively stable trend, confirmed through statistical tests such as the ADF test. Models like ARIMA were used to forecast future fraud volumes over time.

Further exploratory analysis (univariate, bivariate, and multivariate) helped identify abnormal behaviors in variables such as balances and transaction amounts, especially under fraudulent conditions. New features were engineered to capture discrepancies between amounts and account balances.

Two classification models were developed: a penalized logistic regression and a Random Forest, the latter showing the best performance, with a recall of 96%, precision of 79%, a very low log loss (0.00096), and a Cohen's Kappa of 0.86.

Finally, the model was deployed in a Streamlit application, enabling real-time fraud prediction. Additional business logic was implemented to support decision-making and enhance interpretability of the model outputs.

Key words : Fraud detection, Time series analysis, Random Forest Classification, Imbalance class, Streamlit App Deployment

Introduction

La fraude financière constitue aujourd'hui l'un des défis majeurs dans le domaine bancaire et des services de paiement. Elle se manifeste par toute tentative délibérée de tromper un système à des fins de gain illégal, notamment à travers des transactions non autorisées, des détournements de fonds ou des manipulations de données. Dans un environnement numérique en constante évolution, où les volumes de transactions explosent et les méthodes frauduleuses se complexifient, la détection précoce et automatisée de ces activités frauduleuses est devenue un impératif stratégique pour les institutions financières.

Ce travail s'inscrit dans le cadre d'un atelier académique de la formation "Traitement de l'Information et Data Science en Entreprise" à l'Université Paris 1 Panthéon-Sorbonne. Il vise à développer un système de détection de fraude basé sur des techniques d'apprentissage automatique, en mobilisant des données de transactions bancaires simulées. L'objectif est double : d'une part, explorer les schémas caractéristiques des opérations frauduleuses ; d'autre part, concevoir un modèle prédictif performant capable d'identifier en temps réel les transactions suspectes.

Dans ce contexte, plusieurs défis se posent, notamment le déséquilibre très prononcé entre les classes (les fraudes représentant une infime minorité), le choix des variables pertinentes, la définition d'un seuil de décision optimal, ainsi que l'interprétation des résultats dans un cadre métier réaliste. Ce rapport présente l'ensemble de la démarche adoptée, depuis l'analyse exploratoire des données jusqu'à la mise en place d'un modèle opérationnel, intégrable dans une application de type web.

Problématique

La détection de fraude dans les systèmes de paiement électronique repose souvent sur des méthodes traditionnelles, telles que des règles heuristiques ou des seuils fixes, qui peinent à s'adapter à l'évolution rapide des techniques frauduleuses. De plus, dans les bases de données transactionnelles, les cas de fraude sont extrêmement rares (parfois moins de 0.1 %), ce qui rend l'apprentissage automatique d'un modèle fiable d'autant plus complexe. Face à cet important déséquilibre de classes, comment construire un modèle capable non seulement de détecter un maximum de fraudes (rappel élevé), mais aussi de limiter les fausses alertes (précision acceptable) afin d'éviter des coûts opérationnels inutiles ?

Objectifs

Ce projet vise à concevoir une solution de détection de fraudes à partir de données simulées mais réalistes, en suivant une démarche complète de data science. Les objectifs spécifiques sont :

1. Analyser le jeu de données pour comprendre la distribution des variables, détecter les anomalies et identifier les comportements typiques des transactions frauduleuses.
2. Créer de nouvelles variables explicatives pertinentes pour améliorer la capacité de détection, notamment en intégrant des différences de soldes avant et après transaction.

3. Développer et comparer plusieurs modèles prédictifs (logistic regression, random forest, etc.), en tenant compte des contraintes spécifiques liées à l'apprentissage sur données déséquilibrées.
4. Évaluer les modèles avec des métriques adaptées, comme la courbe précision-rappel, la perte logarithmique (log-loss) et la statistique de Cohen's Kappa.
5. Proposer un seuil de décision optimisé, en fonction des priorités métiers (maximiser la détection, limiter les faux positifs, ou trouver un compromis).
6. Mettre en place une démonstration applicative, via une interface développée avec Streamlit, permettant de tester le modèle sur de nouvelles transactions en temps réel.

Présentation du jeu de données

Le jeu de données utilisé dans le cadre de cet atelier provient de Kaggle, une plateforme bien connue pour l'hébergement de jeux de données publics et de compétitions en science des données. Ce jeu contient des transactions bancaires afin de modéliser et détecter des fraudes financières dans un environnement réaliste mais synthétique.

Il contient 6 362 620 lignes et 11 colonnes, sans aucune valeur manquante. Chaque ligne représente une transaction financière, caractérisée par plusieurs variables :

- le type d'opération (type) tel que *PAYMENT*, *TRANSFER*, *CASH_OUT*, *CASH IN, DEBIT.*
- les montants engagés (amount)
- les soldes avant et après transaction pour les comptes émetteurs (oldbalanceOrg, newbalanceOrg) et récepteurs (oldbalanceDest, newbalanceDest)
- les identifiants anonymisés des clients (nameOrig, nameDest)
- un pas temporel (step)
- deux variables cibles : isFraud indiquant si la transaction est effectivement frauduleuse, et isFlaggedFraud indiquant si elle a été signalée comme potentiellement suspecte.

Ce dataset constitue un excellent support pédagogique pour expérimenter des techniques de détection de fraude par l'analyse de données et le machine learning.

Table 1. Aperçu du jeu de données

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

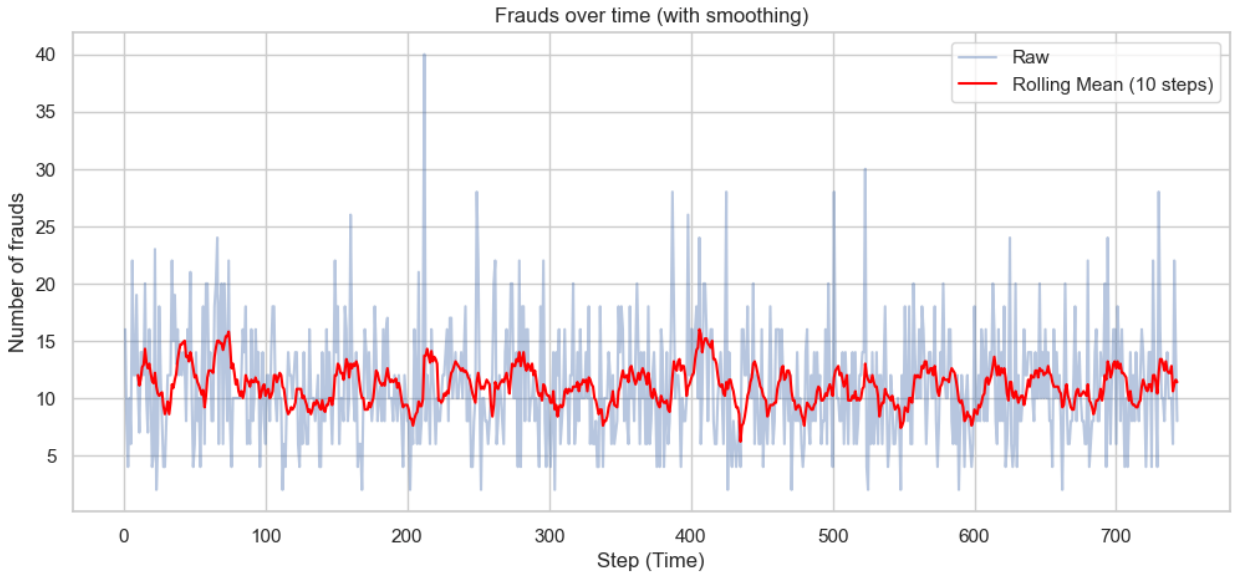
Partie 1 - Analyse de la variable cible “isFraud”

Evolution temporelle des fraudes

La variable “isFraud” est notre variable d’intérêt principale, indiquant si une transaction est frauduleuse (1) ou légitime (0). Sur l’ensemble des 6,3 millions de transactions, seules 8 213 transactions sont marquées comme frauduleuses, ce qui représente environ 0,13 % du total. Cela confirme un déséquilibre marqué dans les classes, une caractéristique fréquente dans les problématiques de détection de fraude, et qui devra être prise en compte dans la modélisation (notamment lors de l’évaluation des performances).

Nous allons d’abord tenter d’analyser le comportement temporel de cette variable, principalement le nombre de fraudes détectées, et tenter de modéliser leur fréquence au cours du temps.

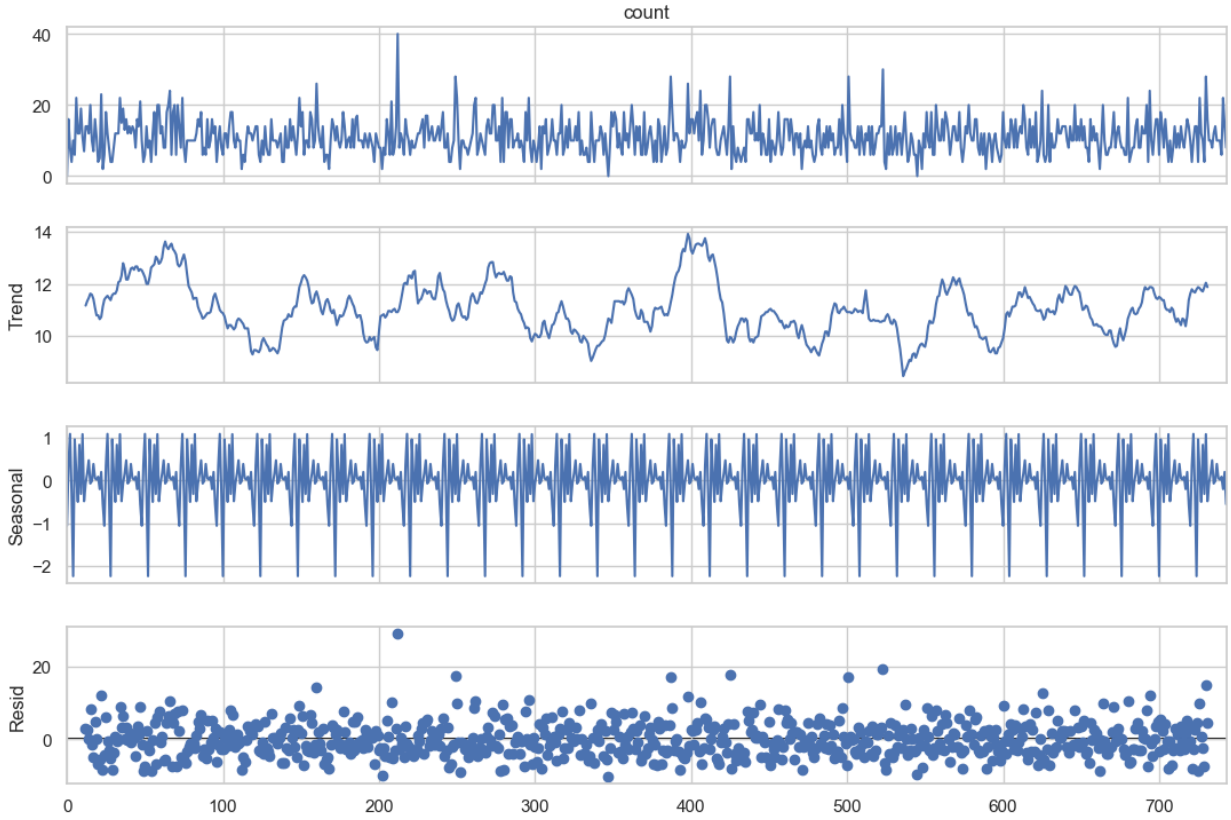
Figure 1. Evolution temporelle des fraudes



Le graphique ci-dessus représente le nombre de fraudes détectées au fil du temps (variable step, assimilable à des unités de temps régulières, variant de 1 à 743). La courbe bleue montre la série brute, avec une forte variabilité et des pics ponctuels atteignant jusqu’à 40 fraudes sur une unité de temps. Pour mieux visualiser les tendances sous-jacentes, une moyenne glissante (rolling mean) sur 10 pas de temps a été ajoutée en rouge. Cette courbe lissée révèle une relative stabilité du volume de fraudes dans le temps, oscillant autour de 10 à 13 fraudes par unité. Aucune tendance marquée à la hausse ou à la baisse n’est visible à ce stade, mais une légère cyclicité peut être soupçonnée, ce qui sera exploré plus finement dans l’analyse de la saisonnalité.

Décomposition de la série temporelle des fraudes

Figure 2. Décomposition de la série temporelle des fraudes



L'analyse de la série temporelle des fraudes montre une tendance globalement stable avec quelques fluctuations modérées au fil du temps. La composante tendance oscille autour de 10 à 13 fraudes par unité de temps sans évolution clairement croissante ou décroissante, suggérant une activité frauduleuse relativement constante. La composante saisonnière présente un motif récurrent et marqué, indiquant que certaines périodes sont systématiquement plus sujettes aux fraudes – probablement en lien avec des cycles d'activité utilisateur ou des moments stratégiques choisis par les fraudeurs.

Les résidus, quant à eux, sont globalement bien répartis autour de zéro, avec quelques valeurs aberrantes pouvant correspondre à des pics de fraudes inattendus. Enfin, le graphique d'autocorrélation montre une absence de corrélation significative entre les pas de temps, ce qui suggère que le nombre de fraudes à un moment donné n'est pas dépendant des périodes précédentes. Ces observations mettent en évidence un comportement frauduleux régulier mais non autocorrélé, ce qui peut guider les choix de modélisation futurs.

Stationnarité et choix du modèle

Afin d'évaluer la stationnarité de la série temporelle du nombre de fraudes, nous avons effectué le test d'Augmented Dickey-Fuller (ADF). Le résultat a donné une p-value égale à 0.0, ce qui nous conduit à rejeter l'hypothèse nulle de non-stationnarité. Nous pouvons donc considérer que notre série est stationnaire, ce qui justifie l'utilisation d'un modèle ARIMA sans différenciation (d = 0).

À partir de ce constat, nous avons utilisé l'algorithme auto_arima pour sélectionner automatiquement les paramètres optimaux du modèle. Le meilleur modèle retenu est ARIMA(2,0,0), ce qui indique une dépendance aux deux périodes précédentes sans différenciation ni moyenne mobile. Ce modèle a ensuite été utilisé pour effectuer une prévision du nombre de fraudes sur les 30 prochaines unités de temps.

Évaluation des performances du modèle

L'ajustement du modèle ARIMA a été évalué à l'aide de deux métriques classiques en séries temporelles : la MAE (Mean Absolute Error) et la RMSE (Root Mean Squared Error). Le modèle obtient une MAE de 3,93, ce qui signifie qu'en moyenne, l'erreur de prédiction est d'environ quatre fraudes par unité de temps. Le RMSE, légèrement plus élevé à 5,03, indique que certaines erreurs de prédiction sont plus importantes, cette métrique étant plus sensible aux écarts extrêmes. Ces résultats traduisent une capacité modérée du modèle à capturer la dynamique temporelle du nombre de fraudes, bien que la forte variabilité de la série et l'absence de variables explicatives puissent limiter la précision du modèle.

Conclusion partielle

L'analyse temporelle des transactions frauduleuses a permis de mieux comprendre la dynamique de leur apparition au fil du temps. Bien que les fraudes apparaissent de manière apparemment aléatoire, la décomposition saisonnière a mis en évidence une légère composante cyclique régulière, ainsi qu'une tendance globale relativement stable, sans variation majeure sur le long terme.

L'autocorrélation a confirmé l'absence de dépendance temporelle significative entre les périodes successives, ce qui suggère une quasi-stationnarité du processus de génération des fraudes. Ce constat a été appuyé par un test ADF (Augmented Dickey-Fuller) concluant à la stationnarité de la série.

Une modélisation ARIMA a été mise en œuvre pour prévoir l'évolution des fraudes à court terme. Le modèle sélectionné (ARIMA(2,0,0)) a permis de générer une prévision relativement constante du nombre de fraudes pour les prochaines étapes, autour de 11 cas par unité de temps, reflétant une forte inertie du phénomène. Cette stabilité pourrait s'expliquer par un système de détection ou de comportement frauduleux assez homogène sur la période étudiée.

Bien que le pouvoir prédictif du modèle soit limité par la faible variabilité temporelle, cette approche permet d'anticiper les volumes attendus et peut s'intégrer dans une stratégie de surveillance en temps réel. Pour aller plus loin, une approche de deep learning (ex. LSTM) pourrait être envisagée pour capter d'éventuelles dynamiques non linéaires.

Partie 2 – Analyse exploratoire des variables exogènes

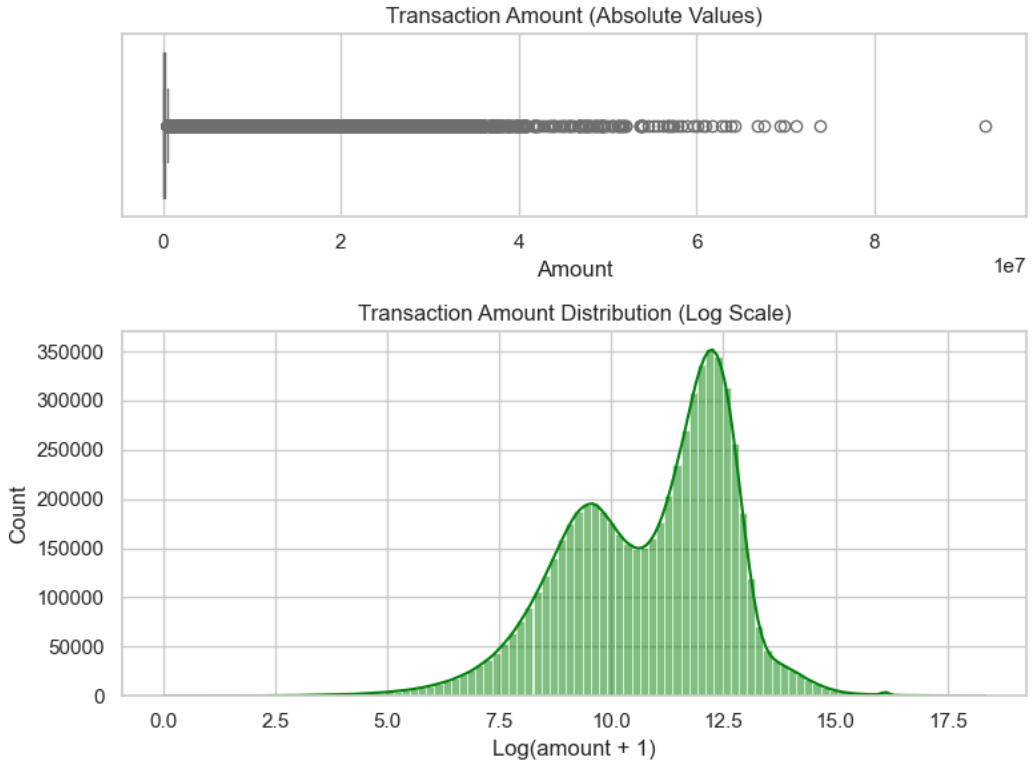
Après l'étude de la variable cible "isFraud" et de son évolution temporelle, nous poursuivons avec une analyse exploratoire des autres variables de la base. L'objectif est d'identifier les relations potentielles entre les caractéristiques des transactions (montant, type, soldes avant/après, etc.) et la survenue d'une fraude. Cette étape est essentielle pour mieux comprendre les mécanismes sous-jacents aux fraudes et orienter les choix de modélisation pour prédire si une nouvelle transaction est une fraude ou non, avec une certaine probabilité.

Analyse univariée des variables

L'analyse exploratoire des principales variables du jeu de données révèle des dynamiques très asymétriques et une forte hétérogénéité dans les comportements financiers.

- **Type de transaction** :
  Les opérations CASH_OUT et PAYMENT dominent en volume, dépassant chacune les **2 millions** de transactions. Les TRANSFER (transferts vers un autre compte) sont moins nombreuses. Les CASH_IN (dépôts sur le compte) sont aussi assez courantes, tandis que DEBIT reste marginal.

Figure 3. Distribution de la variable 'amount' – Absolue et Log



- **Montants des transactions (amount)** :
  Les montants bruts sont très dispersés avec de nombreux outliers au-delà de plusieurs millions. Pour atténuer cette asymétrie, une transformation logarithmique (log(amount + 1)) a été appliquée. On observe alors deux pics :
  - Autour de 10.5 ≈ 36 000 €
  - Et 12.5 ≈ 270 000 €

  Cela suggère deux grandes classes de transactions : des opérations standards et des transactions d'un montant particulièrement élevé, souvent suspectes.
- **Solde de l'émetteur avant transaction (oldbalanceOrg)** :
  Une grande part des clients ont un solde proche de 0. Toutefois, après transformation log, la densité s'étale autour de :
  - 10.5 ≈ 36 000 €
  - Et un second pic vers 15 ≈ 3 270 0 000 €

  Cela reflète une population mixte entre comptes peu approvisionnés et comptes très actifs.
- **Solde de l'émetteur après transaction (newbalanceOrig)** :
  La concentration autour de **0** est encore plus marquée ici. Cela correspond à des retraits totaux, souvent observés dans les CASH_OUT. La transformation log révèle une structure similaire à celle du solde initial, bien que légèrement plus comprimée.
- **Solde du destinataire avant transaction (oldbalanceDest)** :
  Là encore, une majorité de destinataires commencent avec un solde nul. Le pic principal en log apparaît vers :
  - log(13.5) ≈ 730 000 €
- **Solde du destinataire après transaction (newbalanceDest)** :
  Distribution similaire à celle de oldbalanceDest avec un léger décalage vers la droite. Cela indique des dépôts importants, typiquement après des TRANSFER ou des CASH_OUT.

Une forte présence de valeurs aberrantes est remarquée dans chacune de ces distributions.

Test de normalité

Avant de mettre en place des modèles statistiques, il est pertinent de vérifier si les principales variables quantitatives suivent une loi normale afin de déterminer la méthode de standardisation la plus appropriée. Pour cela, nous avons appliqué le test de Kolmogorov-Smirnov (la taille de notre table étant très grande > 5000) qui compare la distribution empirique des données à une loi normale théorique.

Les résultats indiquent clairement qu'aucune des variables ne suit une distribution normale (valeurs de $p < 0.05$), notamment les montants des transactions et les soldes des comptes. Cette non-normalité indique le choix des transformations à faire et aussi le recours à des modèles non paramétriques ou robustes comme les forêts aléatoires pour mieux capturer les schémas complexes de la fraude.

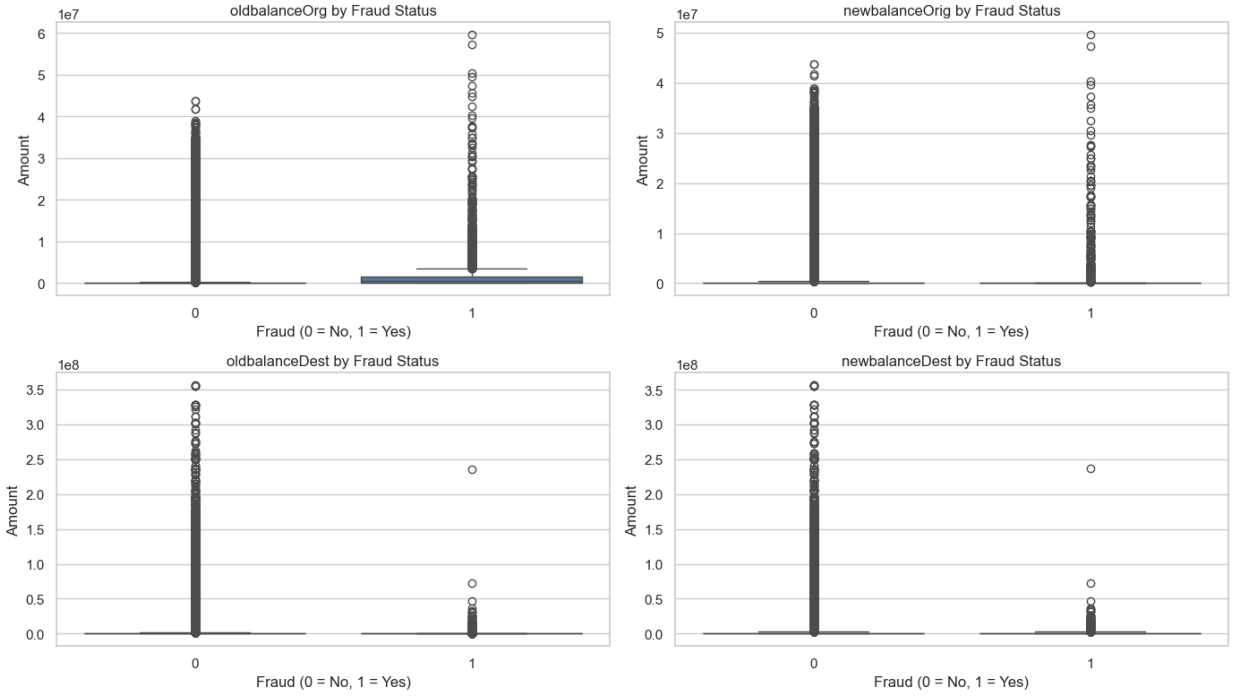
Analyse bivariée des variables

L'analyse bivariée met en évidence des différences notables entre les groupes frauduleux et non frauduleux. Le premier graphique montre clairement que seuls les types de transactions TRANSFER et CASH_OUT sont associés à des fraudes, avec des taux respectifs de 0,77 % et 0,18 %, tandis que les autres types de transactions (CASH_IN, DEBIT, PAYMENT) ne présentent aucun cas de fraude. Cette observation permet déjà de cibler les types d'opérations à surveiller en priorité. Un test d'association de chi-deux a permis d'établir une relation statistiquement significative entre les types de transaction et le statut de fraude.

Les boxplots des soldes avant et après transaction (chez l'émetteur et le bénéficiaire) montrent une dispersion importante des montants, mais on observe que les fraudes sont souvent associées à des soldes initiaux plus élevés, en particulier chez les émetteurs (oldbalanceOrg). Cela laisse penser que les fraudeurs ciblent en priorité des comptes bien alimentés.

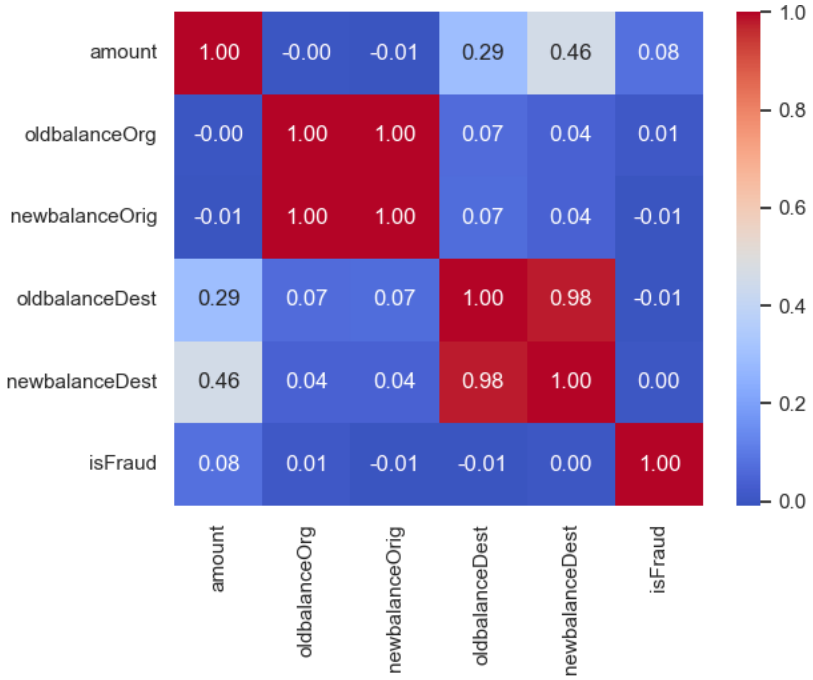
Le test non paramétrique de Kruskal-Wallis, effectué pour chaque variable numérique selon le statut de fraude, confirme que les médianes sont statistiquement différentes entre les groupes frauduleux et non frauduleux, ce qui valide la pertinence de ces variables dans la construction des modèles prédictifs.

Figure 4. Distribution par statut de fraude des soldes avant et après transaction



Analyse de corrélation

Figure 5. Matrice de corrélation des variables exogènes



Plusieurs variables présentent des corrélations fortes entre elles, ce qui pourrait entraîner de la multicolinéarité et nuire à la stabilité ou à l'interprétabilité de certains modèles. Pour limiter cet effet dans le cadre de la régression logistique, une régularisation de type L1 (Lasso) ou L2 (Ridge) sera appliquée. Ce procédé pénalise les informations redondantes et contribue à la stabilisation du modèle. En revanche, les modèles basés sur les arbres de décision (comme la forêt aléatoire) sont généralement robustes face à la multicolinéarité, car ils ne s'appuient pas sur des coefficients comme les modèles linéaires.

Création de variables

Pour affiner l'analyse et détecter d'éventuelles incohérences ou fraudes, deux nouvelles variables ont été créées : **balanceDiffOrig** (différence entre oldbalanceOrg et newbalanceOrg) et **balanceDiffDest** (différence entre newbalanceDest et oldbalanceDest). Ces variables permettent de vérifier si les montants transférés correspondent bien aux variations de soldes observées chez l'émetteur et le destinataire. Elles servent notamment de points de comparaison avec la variable amount, et facilitent ainsi la détection d'anomalies ou de comportements suspects dans les transactions.

Conclusion partielle

L’analyse exploratoire des données a permis de mieux comprendre la structure du jeu de données et d’identifier les premiers signaux associés aux fraudes. La variable cible isFraud est fortement déséquilibrée, avec un très faible taux global de fraude. Les types de transaction TRANSFER et CASH_OUT sont les seuls concernés par des fraudes, ce qui oriente l’analyse vers ces deux catégories. Les tests de normalité ont révélé que les variables numériques ne suivent pas une loi normale, ce qui justifie l’utilisation de tests non paramétriques comme celui de Kruskal-Wallis pour comparer les distributions selon le statut de fraude. Ce test a confirmé des différences significatives entre les groupes. De plus, l’analyse de corrélation a mis en évidence de fortes redondances entre certaines variables de balance, justifiant l’utilisation de techniques comme la régularisation dans les modèles linéaires pour limiter les effets de la multicolinéarité. Ces explorations préliminaires ont permis de dégager des pistes solides pour la modélisation prédictive.

Partie 3 – Modélisation prédictive

Régression logistique

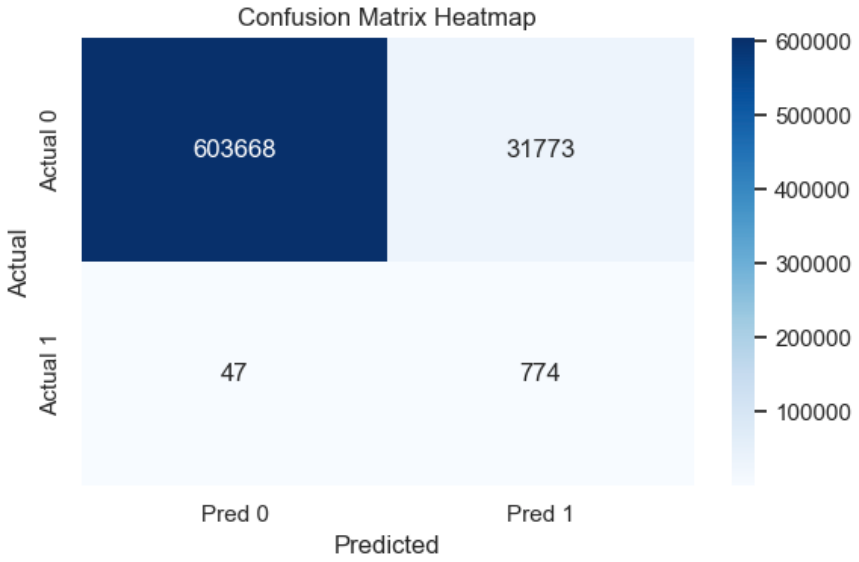
Nous entamons la phase de modélisation par la régression logistique, une méthode de classification binaire couramment utilisée pour estimer la probabilité d'un événement. Dans notre cas, l'objectif est de prédire la survenue d'une transaction frauduleuse (isFraud = 1) à partir des différentes caractéristiques disponibles. Ce modèle présente l'avantage d'être interprétable et relativement rapide à mettre en œuvre. Compte tenu de la présence d'un fort déséquilibre entre les classes, des ajustements sont nécessaires pour garantir des performances fiables. De plus, une régularisation (L1) sera appliquée afin d'atténuer les effets de la multicolinéarité détectée au cours de l'analyse exploratoire. Cette première approche servira de référence pour évaluer la capacité prédictive du modèle sur la détection de fraudes.

Vu la taille de notre base, le modèle a été entrainé sur 90% des données disponibles et testé sur les 10% restants.

Résultats du modèle

Table 2. Rapport de classification du modèle et matrice de confusion - LR

	precision	recall	f1-score	support
0	1.00	0.95	0.97	635441
1	0.02	0.94	0.05	821
accuracy			0.95	636262
macro avg	0.51	0.95	0.51	636262
weighted avg	1.00	0.95	0.97	636262



Les résultats de la régression logistique montrent une forte déséquilibre entre les classes, ce qui se reflète dans les indicateurs de performance.

Sur le rapport de classification :

- La **précision** (ou *precision*) pour la classe 1 (fraudes) est très faible (0.02), ce qui signifie que très peu de prédictions positives sont correctes. En d'autres termes, le modèle émet beaucoup de *faux positifs*.
- Le **rappel** (ou *recall*) pour la classe 1 est élevé (0.94), ce qui indique que le modèle réussit à détecter la majorité des fraudes existantes (*vrais positifs*), mais au prix d'un grand nombre de prédictions incorrectes.
- Le **f1-score**, qui est une moyenne harmonique entre précision et rappel, reste faible (0.05) pour la classe 1, ce qui montre une faible capacité du modèle à équilibrer entre détection correcte et exactitude.

La **matrice de confusion** confirme ces observations :

- Sur 821 fraudes réelles, 774 ont été correctement détectées, mais 47 ont été manquées.
- Cependant, 31 773 transactions normales ont été faussement identifiées comme des fraudes (faux positifs), ce qui est problématique en pratique.

Concernant les **autres métriques globales** :

- L'**accuracy** globale est de 95 %, mais elle est trompeuse en raison du fort déséquilibre des classes.
- Le **Cohen's kappa** est très bas (0.044), ce qui signifie que l'accord entre les prédictions du modèle et les vraies valeurs n'est que légèrement supérieur à celui du hasard.

- La **log loss** est relativement bonne (0.14), ce qui indique que les probabilités prédites par le modèle ne sont pas trop éloignées de la vérité, mais cela ne compense pas sa mauvaise capacité de discrimination pour les classes minoritaires.

Le modèle a donc du mal à bien identifier les fraudes sans générer un grand nombre de faux positifs. Nous avons essayé de trouver un seuil pour optimiser le trade-off recall/precision, mais les résultats n'ont pas été concluants. Une nouvelle approche a donc été adoptée: l'utilisation du modèle non-linéaire Random Forest.

Random Forest

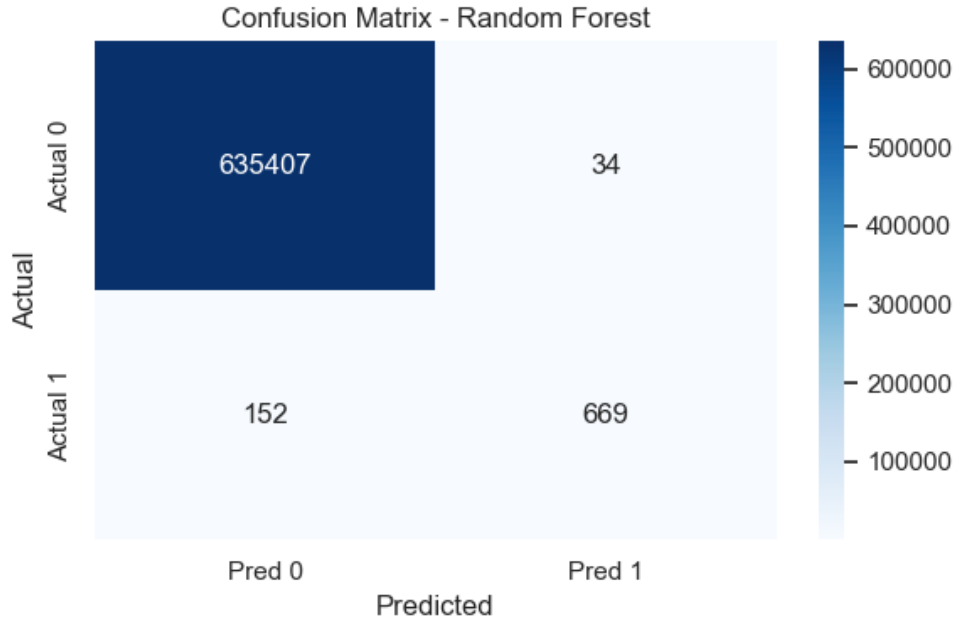
La Random Forest est un algorithme d'ensemble basé sur la combinaison de plusieurs arbres de décision entraînés sur des sous-échantillons des données. Elle est particulièrement adaptée dans le cadre de notre projet de détection de fraudes pour plusieurs raisons. D'abord, elle gère très bien les jeux de données déséquilibrés et non linéaires, en capturant des interactions complexes entre les variables sans nécessiter de transformation préalable.

Ensuite, contrairement à la régression logistique, la Random Forest n'est pas sensible à la multicolinéarité et peut fournir des mesures de l'importance des variables, utiles pour l'interprétation. Enfin, sa robustesse face aux valeurs extrêmes et aux données bruitées en fait un candidat sérieux pour la détection des fraudes, où les comportements anormaux sont rares mais critiques à identifier.

Résultats du modèle

Table 3. Rapport de classification du modèle et matrice de confusion - RF

	precision	recall	f1-score	support
0	1.00	1.00	1.00	635441
1	0.95	0.81	0.88	821
accuracy			1.00	636262
macro avg	0.98	0.91	0.94	636262
weighted avg	1.00	1.00	1.00	636262



Le modèle Random Forest que nous avons testé montre déjà des performances très solides dans la détection des fraudes. Il parvient à identifier 669 fraudes sur 821, soit un rappel de 81 %, ce qui signifie qu'il détecte une large majorité des cas frauduleux. Sa précision de 95 % sur la classe minoritaire confirme que les fraudes prédictes comme telles sont très souvent correctes. Avec une log loss de 0.0048, un Cohen's Kappa de 0.88, et une PR AUC de 0.96, ce modèle présente un bon équilibre entre la détection des fraudes et la limitation des faux positifs. Ces résultats soulignent l'intérêt des modèles d'arbres pour ce type de problématique, notamment grâce à leur capacité à gérer les données déséquilibrées et à capturer des interactions complexes entre les variables.

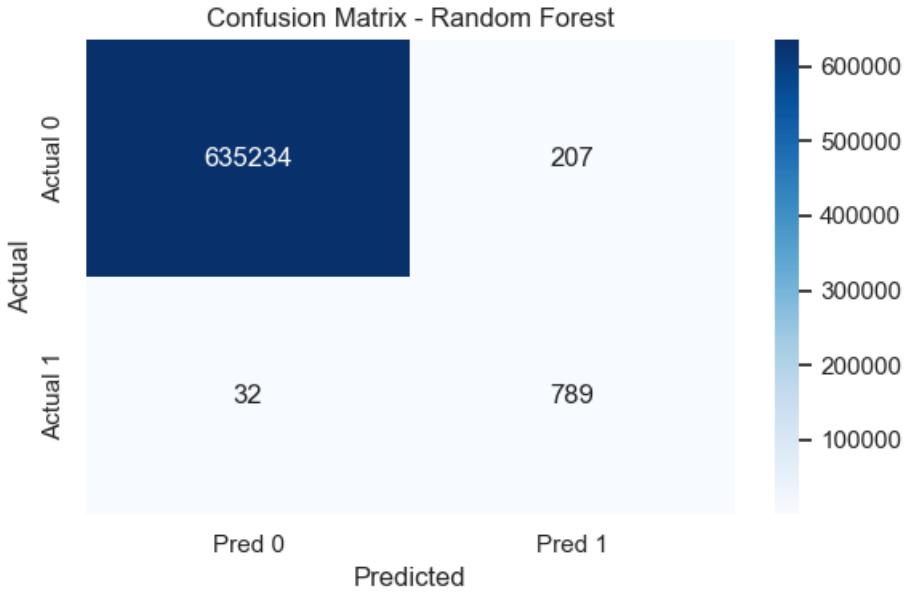
Fine tuning du seuil de décision

Afin de maximiser la détection des fraudes, nous avons affiné notre modèle Random Forest en réajustant ses paramètres, principalement le seuil de décision. L'objectif principal était d'augmenter le rappel (recall) sur la classe minoritaire représentant les fraudes, quitte à accepter une légère baisse de la précision. Cette stratégie s'inscrit dans une logique métier : il est souvent

préférable de signaler des transactions potentiellement frauduleuses, même si certaines s'avèrent être des faux positifs, plutôt que de manquer des fraudes réelles. Pour un seuil fixer à 0.12, les résultats sont encourageants avec un recall passé à 0.96, une précision toujours solide à 0.79, et un Cohen's Kappa de 0.86, indiquant une aussi bonne concordance entre prédictions et réalité. Le modèle est ainsi plus robuste pour les tâches de détection préventive.

Table 4. Rapport de classification du modèle et matrice de confusion – RF optimisé

	precision	recall	f1-score	support
0	1.00	1.00	1.00	635441
1	0.79	0.96	0.87	821
accuracy			1.00	636262
macro avg	0.90	0.98	0.93	636262
weighted avg	1.00	1.00	1.00	636262



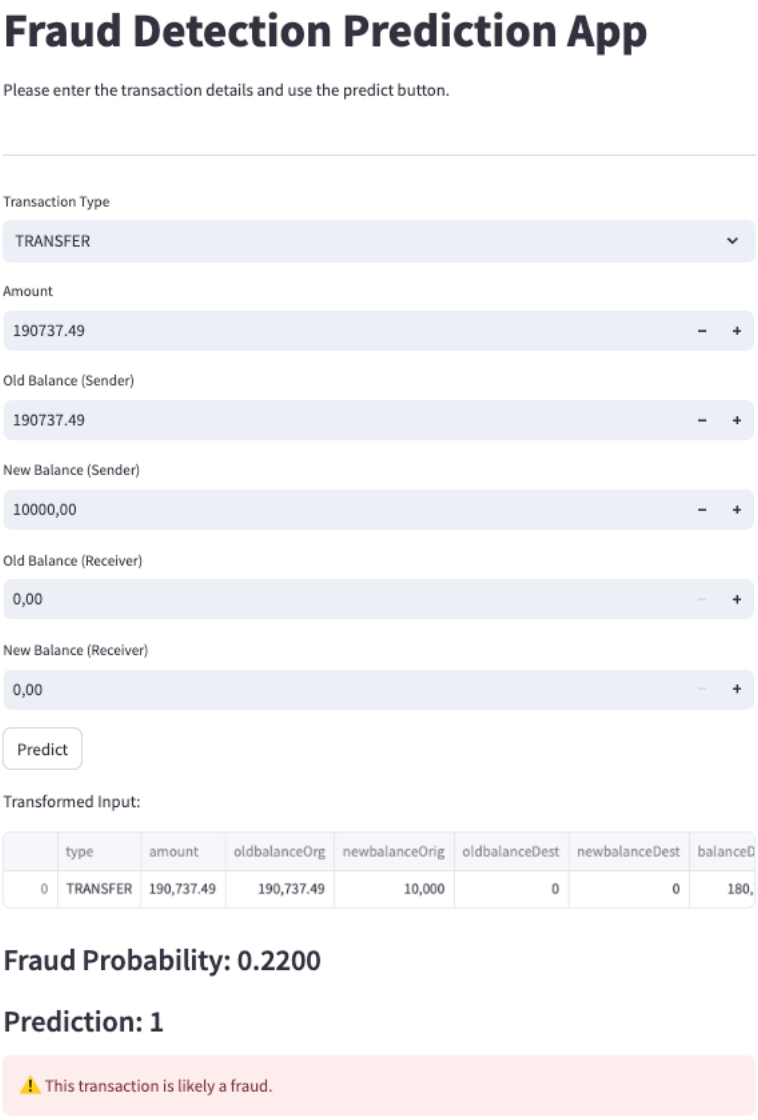
Partie 4 – Implémentation dans une application web

Dans un souci de mise en production et de valorisation concrète du modèle, nous avons développé une application interactive avec Streamlit. Cette interface permet de simuler des transactions et d'obtenir en temps réel une prédiction de leur caractère frauduleux ou non, sur la base du modèle Random Forest entraîné précédemment. L'utilisateur peut saisir les valeurs des différentes variables pertinentes (montant, balances avant/après, type de transaction, etc.) et visualiser immédiatement la décision du modèle.

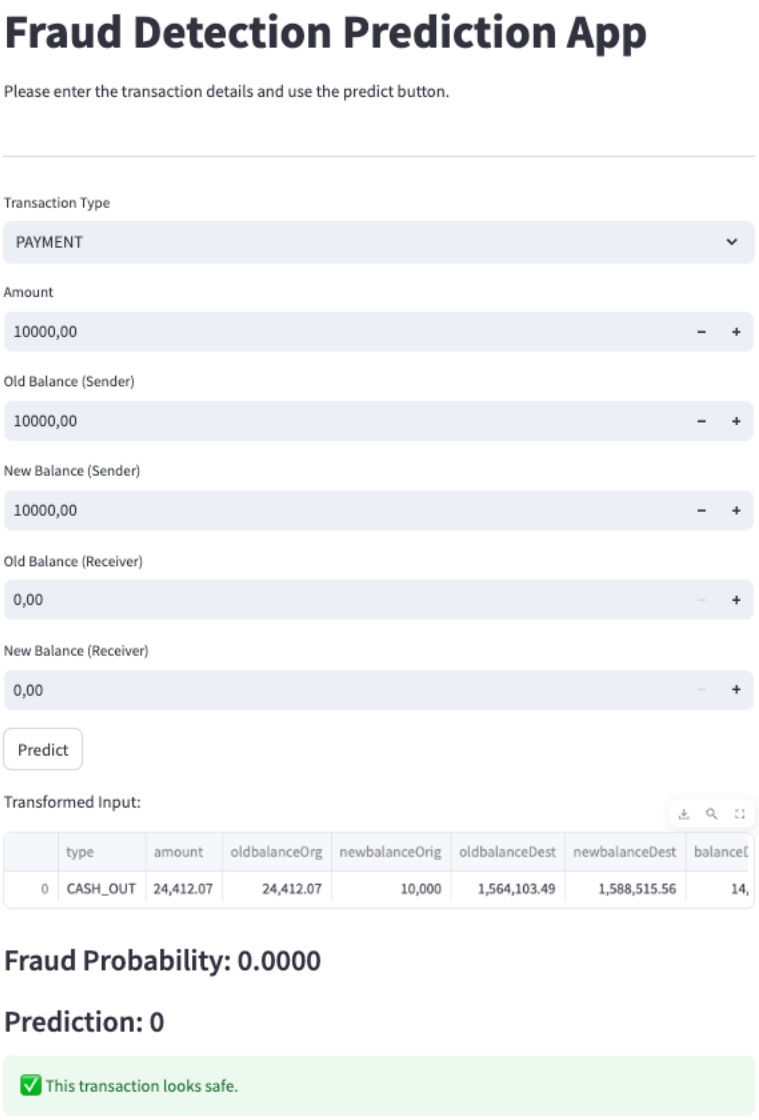
Par ailleurs, des logiques métiers ont été intégrées pour soutenir les décisions du modèle, en identifiant par exemple les cas de transferts suspects vers des comptes au solde initial nul, ou des opérations incohérentes au niveau des balances.

Cette application constitue ainsi une preuve de concept opérationnelle, démontrant comment un algorithme de machine learning peut être intégré dans un outil d'aide à la décision fiable et directement utilisable par les équipes métier.

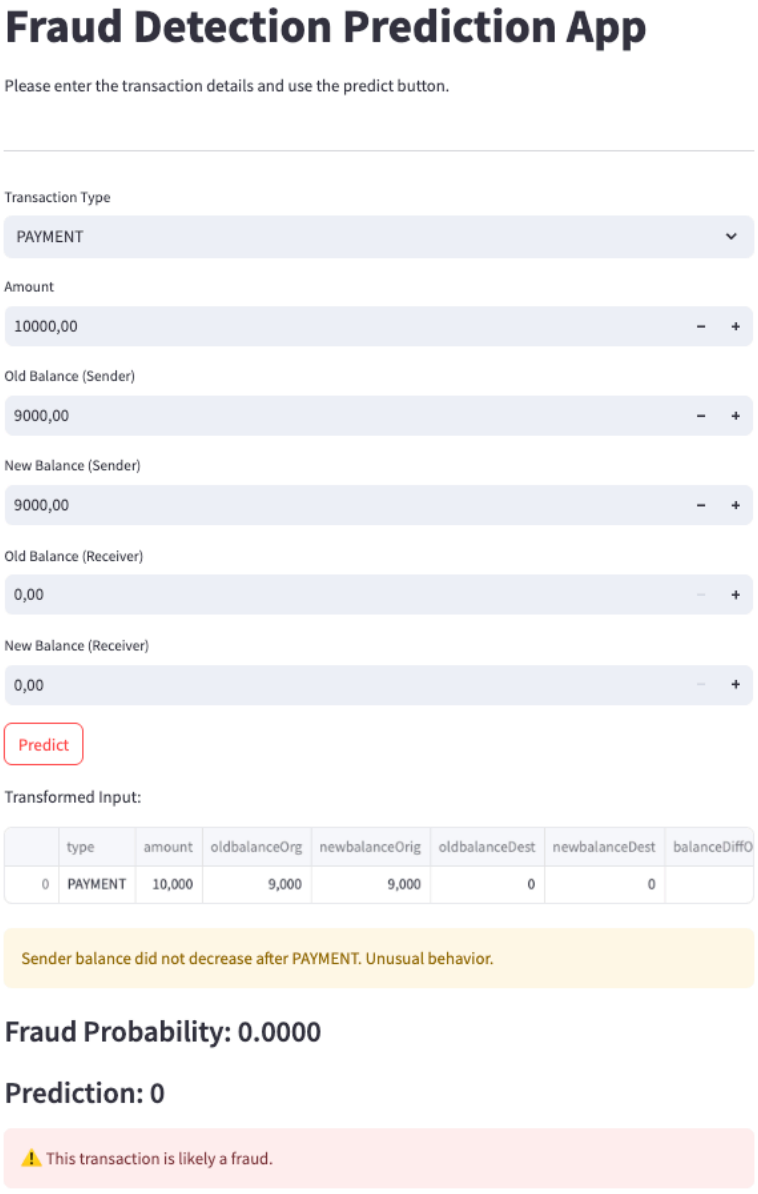
Visuel 1. Détection de fraude par le modèle



Visuel 2. Classification non-fraude d'une transaction par le modèle



Visuel 3. Logique métier en action



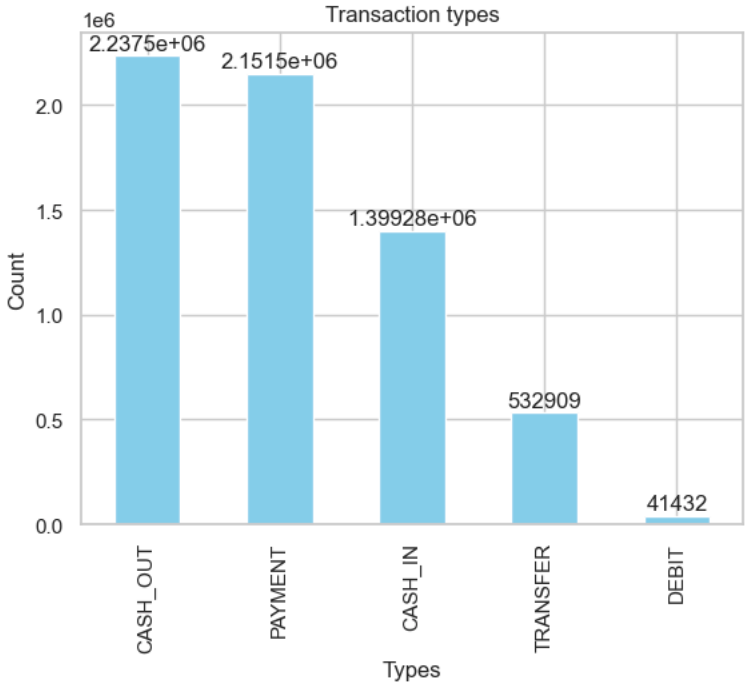
Conclusion

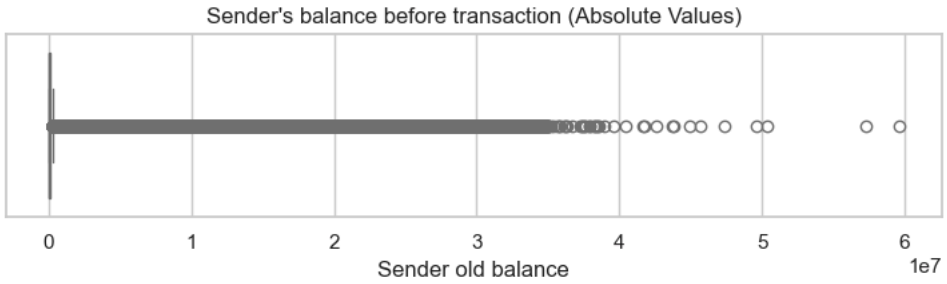
Ce projet a permis de construire un pipeline complet de détection de fraudes sur des transactions financières, en partant d'une analyse exploratoire rigoureuse jusqu'à l'implémentation d'un modèle opérationnel dans une application Streamlit. L'analyse des données a révélé des comportements spécifiques liés aux fraudes, notamment une concentration des cas frauduleux sur certains types de transactions comme *TRANSFER* et *CASH_OUT*. Des tests statistiques et des visualisations ont permis de mieux comprendre la structure des données et d'orienter les choix de modélisation.

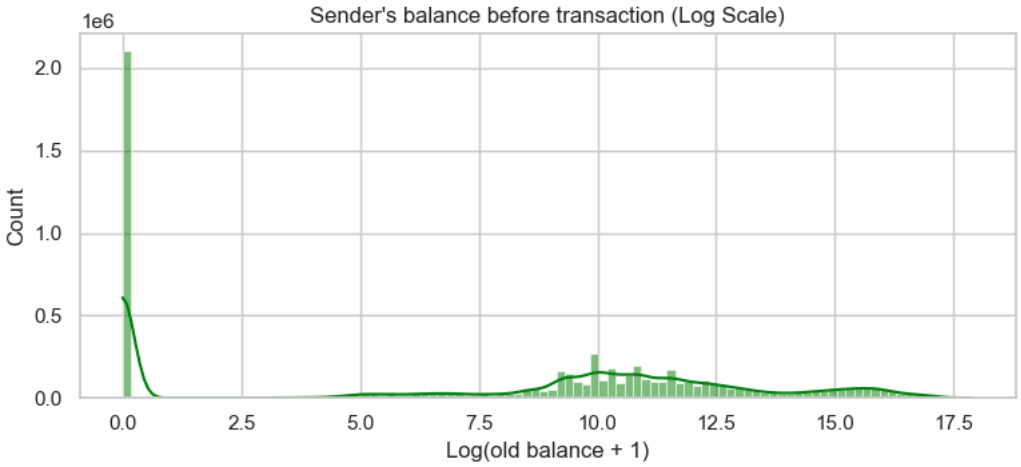
Après une première tentative avec une régression logistique, nous avons opté pour un modèle Random Forest, qui s'est montré particulièrement performant avec un recall de 96 % et une précision de 79 % sur les fraudes. Cette capacité à détecter les fraudes sans trop de faux négatifs est essentielle, car il vaut mieux déclencher une alerte inutile que laisser passer une fraude réelle. L'ajout de logiques métiers a renforcé la robustesse de l'approche, en combinant intelligence artificielle et règles expertes.

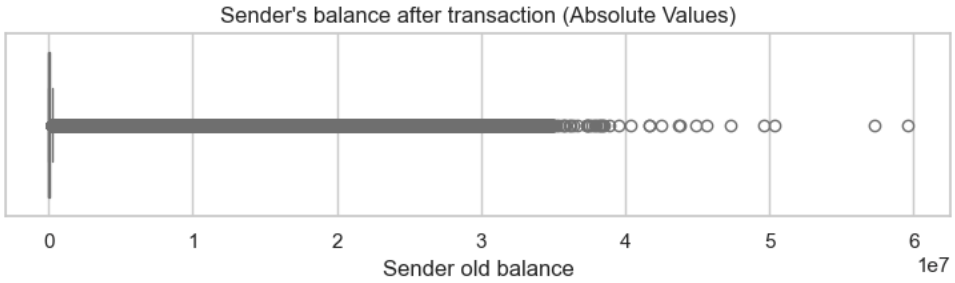
Enfin, le déploiement du modèle dans une interface utilisateur simple d'utilisation montre la faisabilité d'une intégration concrète dans un environnement métier. Ce projet illustre ainsi l'impact direct que peut avoir la data science dans la lutte contre la fraude et l'optimisation des systèmes de surveillance financière.

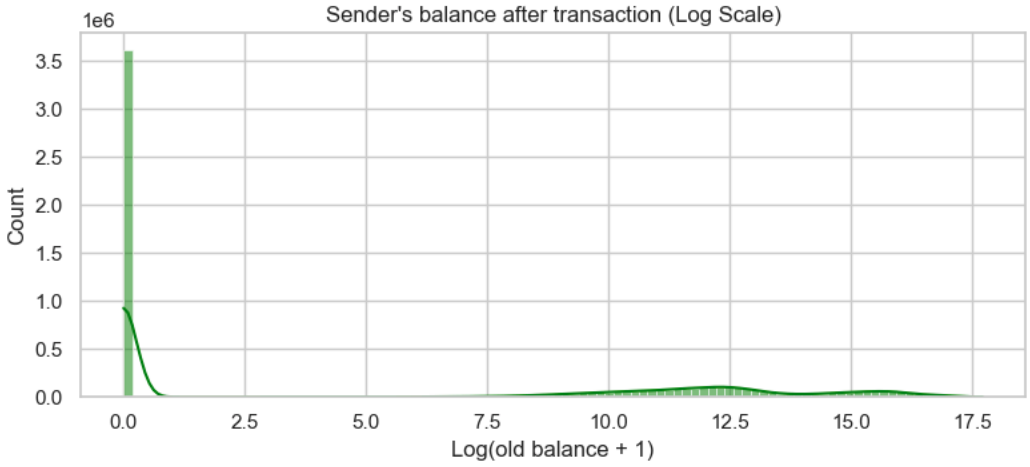
Annexe

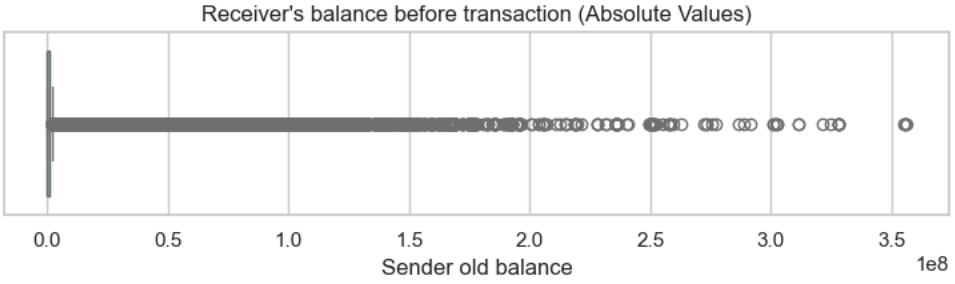

Transaction types
1e6
2.2375e+06
2.1515e+06
1.39928e+06
532909
41432
Count
CASH_OUT
PAYMENT
CASH_IN
TRANSFER
DEBIT
Types

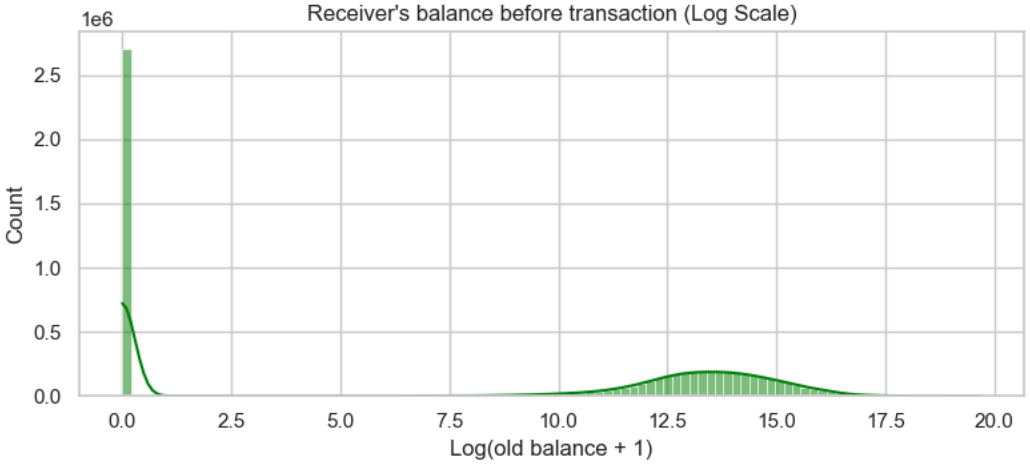

Sender's balance before transaction (Absolute Values)
Sender old balance
1e7


Sender's balance before transaction (Log Scale)
1e6
Count
Log(old balance + 1)

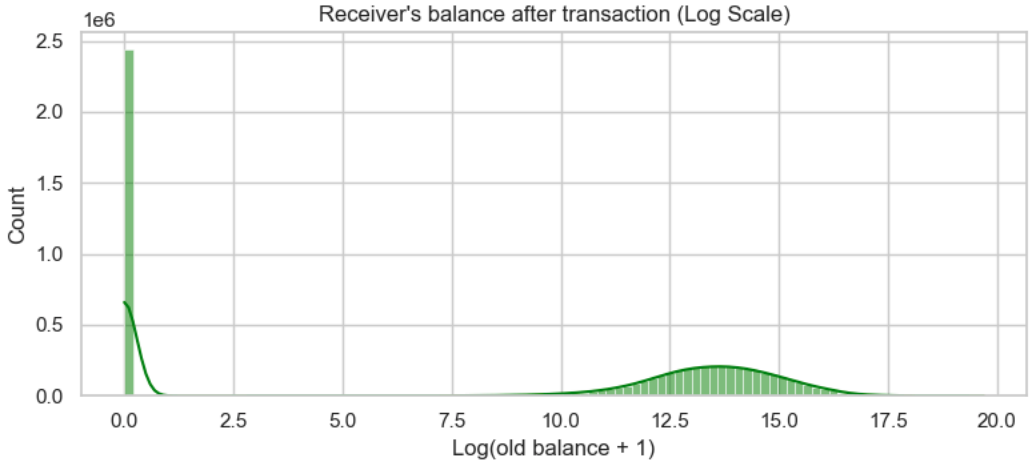

Sender's balance after transaction (Absolute Values)
0
1
2
3
4
5
6
1e7
Sender old balance

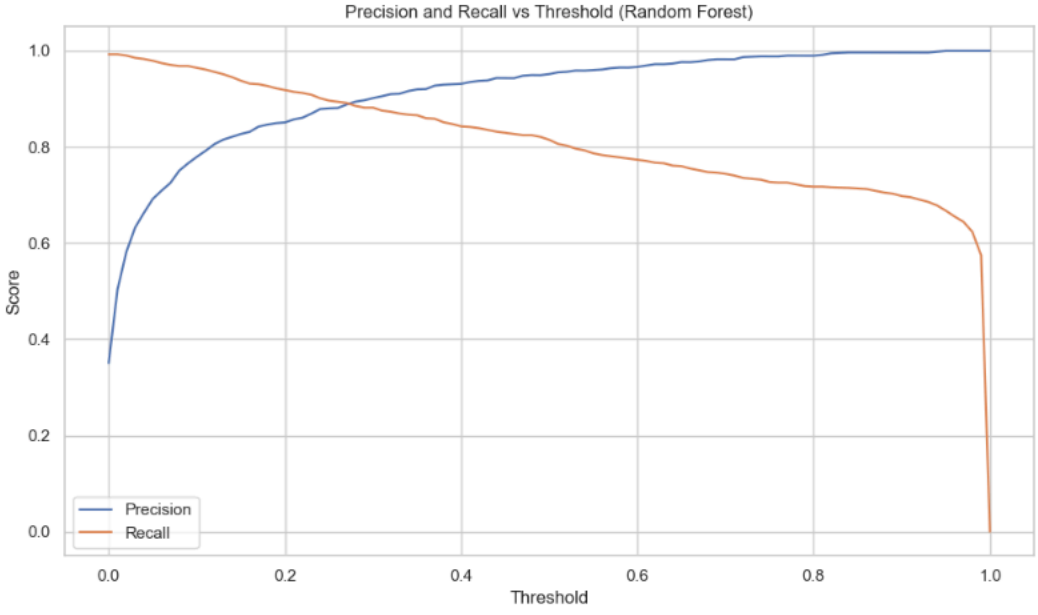

Sender's balance after transaction (Log Scale)
1e6
Count
Log(old balance + 1)


Receiver's balance before transaction (Absolute Values)
1e8
Sender old balance


Receiver's balance before transaction (Log Scale)
1e6
Count
Log(old balance + 1)


Receiver's balance after transaction (Absolute Values)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Sender old balance
1e8


Receiver's balance after transaction (Log Scale)
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Log(old balance + 1)


Precision and Recall vs Threshold (Random Forest)
1.0
0.8
0.6
0.4
0.2
0.0
Score
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Threshold