Akademi

Bootcamp Data Science & IA

Projet de détection de fraude (IEEE Fraud Detection)

Réalisé par Mackenson JOSEPH et Abdielson LYVERT

Octobre 2025

Contexte

La fraude en ligne représente un risque majeur pour les institutions financières et les plateformes de paiement numérique.
Face à l'augmentation du volume de transactions et à la complexité des comportements frauduleux, les méthodes classiques de détection ne suffisent plus.

Le projet **IEEE Fraud Detection** s'inscrit dans le cadre du **challenge international organisé sur Kaggle par l'IEEE Computational Intelligence Society**, visant à développer des modèles de machine learning capables d'identifier les transactions potentiellement frauduleuses à partir de données massives et variées.

L'objectif est de montrer notre capacité à comprendre le système financier et à apporter des solutions pouvant aider les entreprises du secteur en **Haïti** à lutter contre la fraude en ligne.

Objectif du projet

L'objectif principal est de **concevoir, entraîner et déployer un modèle prédictif performant** capable de détecter les fraudes avec un **taux de faux positifs minimal**.
Le projet vise à fournir une solution qui pourrait être intégrée dans un système bancaire réel pour renforcer la sécurité des paiements électroniques.

Composition de l'equipe

Mackenson JOSEPH – *Responsable du développement du modèle et de la mise en production (MLOps)*

Abdielson LYVERT – *Responsable de l'analyse exploratoire des données (EDA)*

Données utilisées

Les données proviennent du **jeu de données officiel du concours IEEE-CIS Fraud Detection** sur Kaggle.
Elles comprennent plusieurs fichiers volumineux et anonymisés :

- train_transaction.csv et train_identity.csv pour l'entraînement.
- test_transaction.csv et test_identity.csv pour la prédiction finale.

Caractéristiques principales :

- Des centaines de variables numériques et catégorielles.
- Des données sensibles (cartes bancaires, adresses IP, appareils, etc.) anonymisées.
- La variable cible isFraud (1 = transaction frauduleuse, 0 = transaction normale).

Méthodologie

Le travail a été organisé en plusieurs **notebooks** afin de structurer le projet selon un flux de travail professionnel, inspiré de la méthodologie utilisée en entreprise :

Étape	Notebook	Responsable	Description
1	01_data_exploration.ipynb	Abdielson LYVERT	Analyse exploratoire des données (distributions, corrélations, valeurs manquantes, etc.)
2	02_correlation_analysis.ipynb	Abdielson & Mackenson	Nettoyage des données et préparation des variables
3	03_model_training.ipynb	Mackenson JOSEPH	Entraînement et optimisation des modèles (XGBoost, LightGBM, CatBoost)

4	production.py	Mackenson JOSEPH	Sauvegarde et mise en production du modèle via API Flask

Technologies utilisées

- **Langage :** Python
- **Outils d'analyse :** Pandas, NumPy, Matplotlib, Seaborn
- **Modélisation :** Scikit-learn, XGBoost, LightGBM, CatBoost
- **Déploiement :** Flask / Pickle
- **Environnement :** Jupyter Notebook, Anaconda
- **Versioning :** Git & GitHub

Résultats attendus

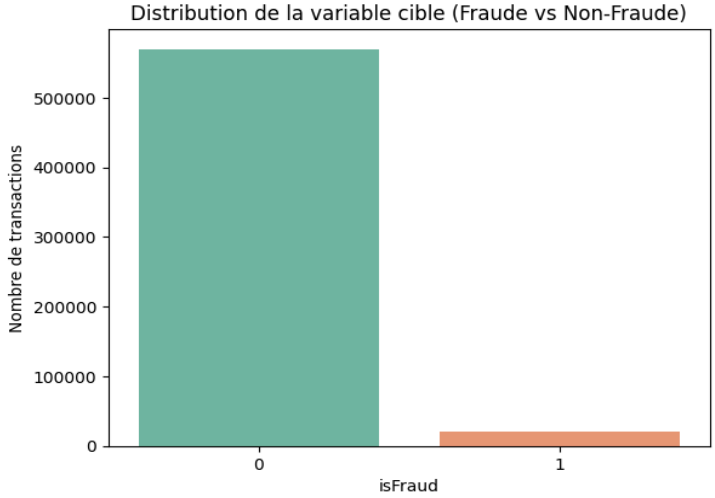
- Développement d'un modèle capable de détecter les fraudes avec une **AUC > 0.90**.
- Réduction significative des faux positifs pour éviter les alertes inutiles.
- Mise à disposition d'une **API REST** permettant de soumettre des transactions et de recevoir une probabilité de fraude en réponse.

Livrables

- Ensemble des notebooks de travail bien structurés et documentés.
- Modèle entraîné et sauvegardé (.pkl).
- API Flask pour le déploiement.
- Rapport technique (ce document).
- README professionnel pour GitHub.

Analyse exploratoire

1) Distribution de la variable cible

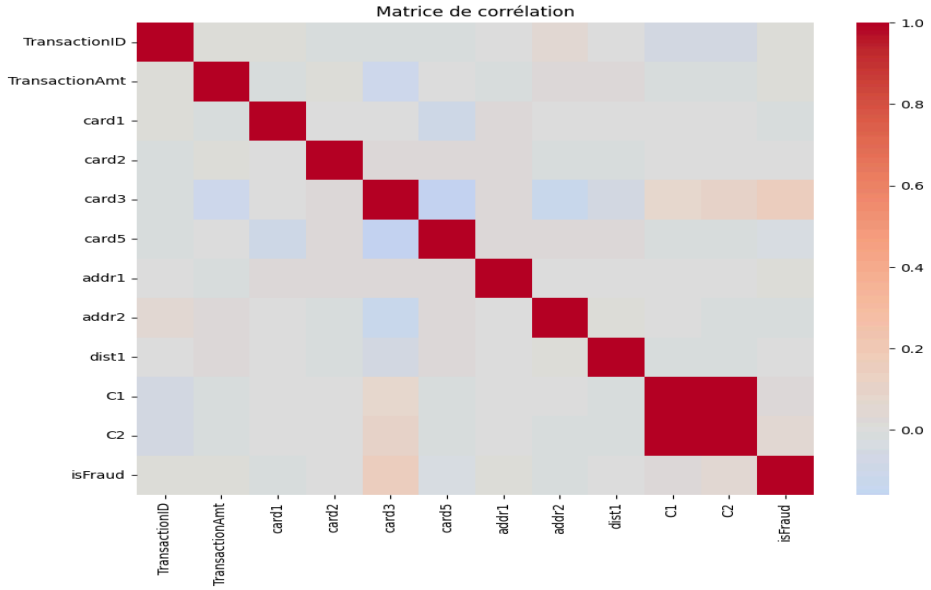


Répartition des classes

Non-Fraude : 96.5% (569,947 transactions)
Fraude : 3.5% (20,593 transactions)

Interprétation : Déséquilibre sévère nécessitant des techniques spécialisées

2) Analyse des Corrélations

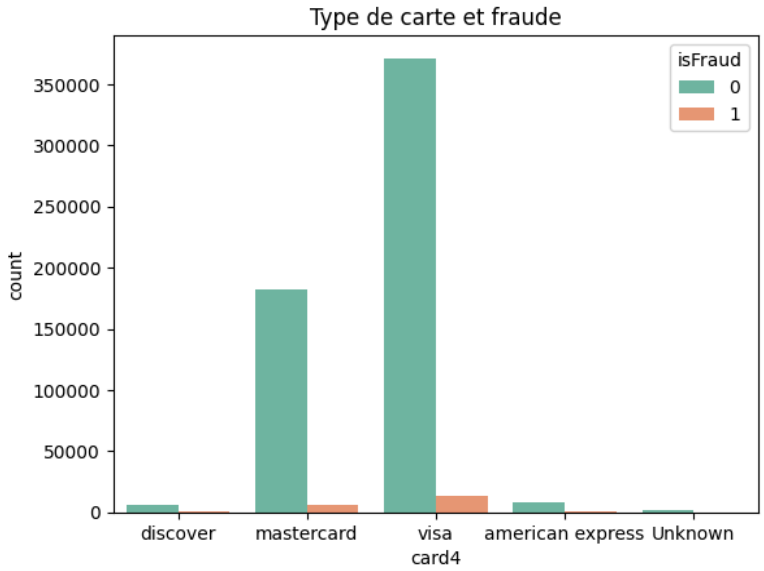


Top 5 des variables corrélées avec isFraud:

1. card3 : 0.154
2. C2 : 0.037
3. C1 : 0.031
4. TransactionID : 0.014
5. TransactionAmt : 0.011

Interprétation : Aucune corrélation forte individuelle → la fraude dépend de signaux faibles combinés .

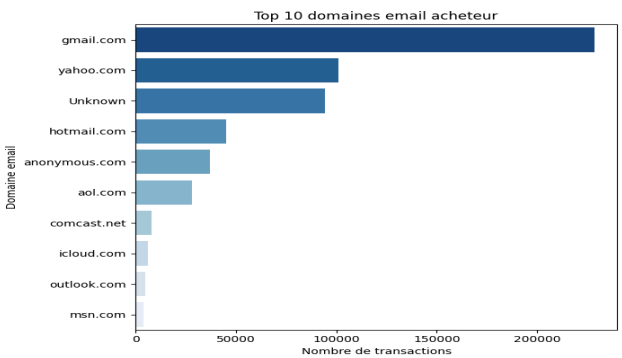
3) Analyse par Type de Carte

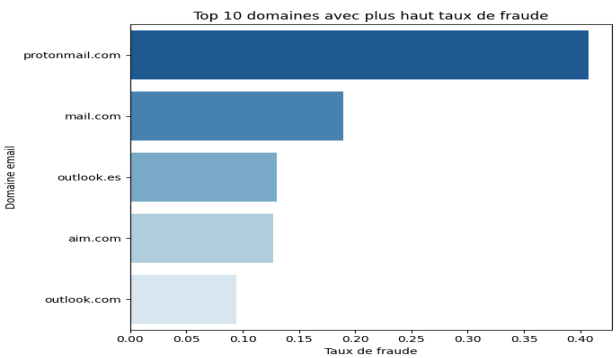


Cartes dominantes : Visa et Mastercard

Interprétation : Les fraudeurs utilisent les mêmes types de cartes que les clients légitimes

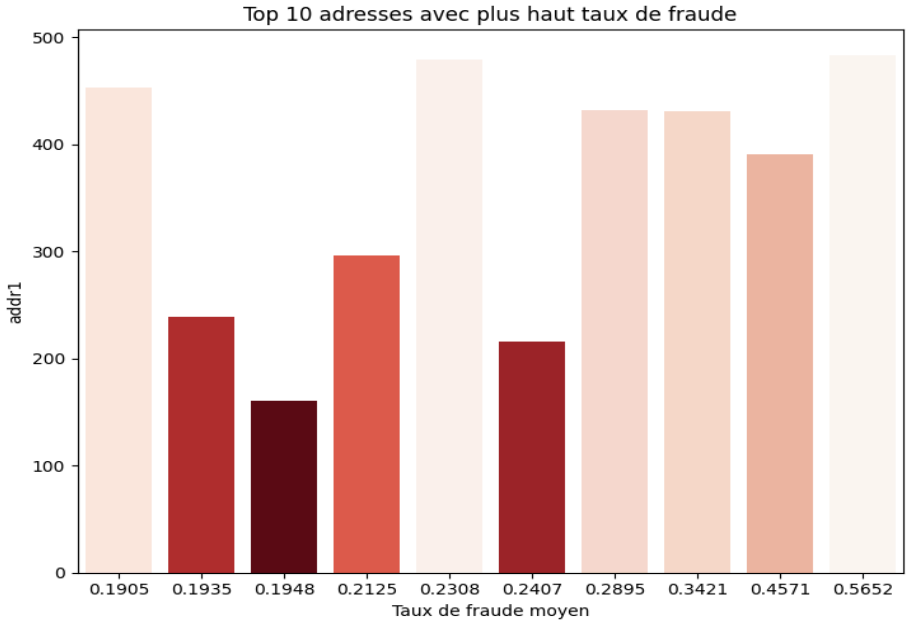
4) Analyse et visualisation des domaines email





Interprétation : Les acheteurs utilisent beaucoup les domaines mail comme *gmail.com, yahoo.com* avec plus de *180 000 transactions*, Mais ce qui ne signifie pas que tous ces acheteurs commettent des transactions frauduleuses, ce qui prouve avec le deuxième graphe qui montre que les acheteurs qui utilisent *protonmail.com* ont plus de *38% transaction frauduleuses*

5) Analyse et visualisation des features géographiques



Interprétation : Ce graphe nous montre que les zones géographique *483, 391* sont très dangereuses et *431, 432* ont un haut risque, ce qui nécessite une intervention immédiate.

Interprétation des Résultats

Dans notre 'analyse exploratoire des données (EDA) et le prétraitement nous avons révélé plusieurs insights clés :

- **Déséquilibre des classes** : Seulement *3,5 % des transactions sont frauduleuses*, ce qui confirme la nature *déséquilibrée* du dataset et la nécessité d'utiliser des techniques adaptées (rééchantillonnage, métriques spécifiques).

- **Variables corrélées à la fraude** : Aucune variable ne présente une *corrélation forte* avec la cible `*isFraud*`. La variable `card3` présente la corrélation la plus

élevée (0,15), suivie de `C1` et `C2`. Cela suggère que la fraude résulte d'une combinaison de signaux faibles plutôt que d'un seul facteur explicite.

- **Email et risque de fraude** : Les domaines `protonmail.com` et `mail.com` présentent des taux de fraude significativement plus élevés (jusqu'à 38 %), tandis que les domaines grand public (`gmail.com`, `yahoo.com`) sont majoritairement utilisés mais moins risqués.

- **Géographie et fraude** : Certaines zones géographiques (`addr1` = 483, 391, 431, 432) présentent des taux de fraude très élevés (jusqu'à 56 %), indiquant des points chauds nécessitant une surveillance renforcée.

- **Type de carte** : Les cartes *Visa et Mastercard* sont les plus utilisées, tant par les clients légitimes que par les fraudeurs, reflétant leur dominance sur le marché.

Analyse de corrélation

Pour améliorer la rapidité et la robustesse du modèle, ainsi que réduire le risque de surapprentissage, nous avons analysé la corrélation entre les colonnes tout en tenant compte des valeurs manquantes.

Objectif :
Réduire le nombre de colonnes en conservant les plus informatives, tout en supprimant celles fortement corrélées ou contenant trop de valeurs manquantes.

Méthode :

1. Regrouper les colonnes selon le nombre de valeurs manquantes.

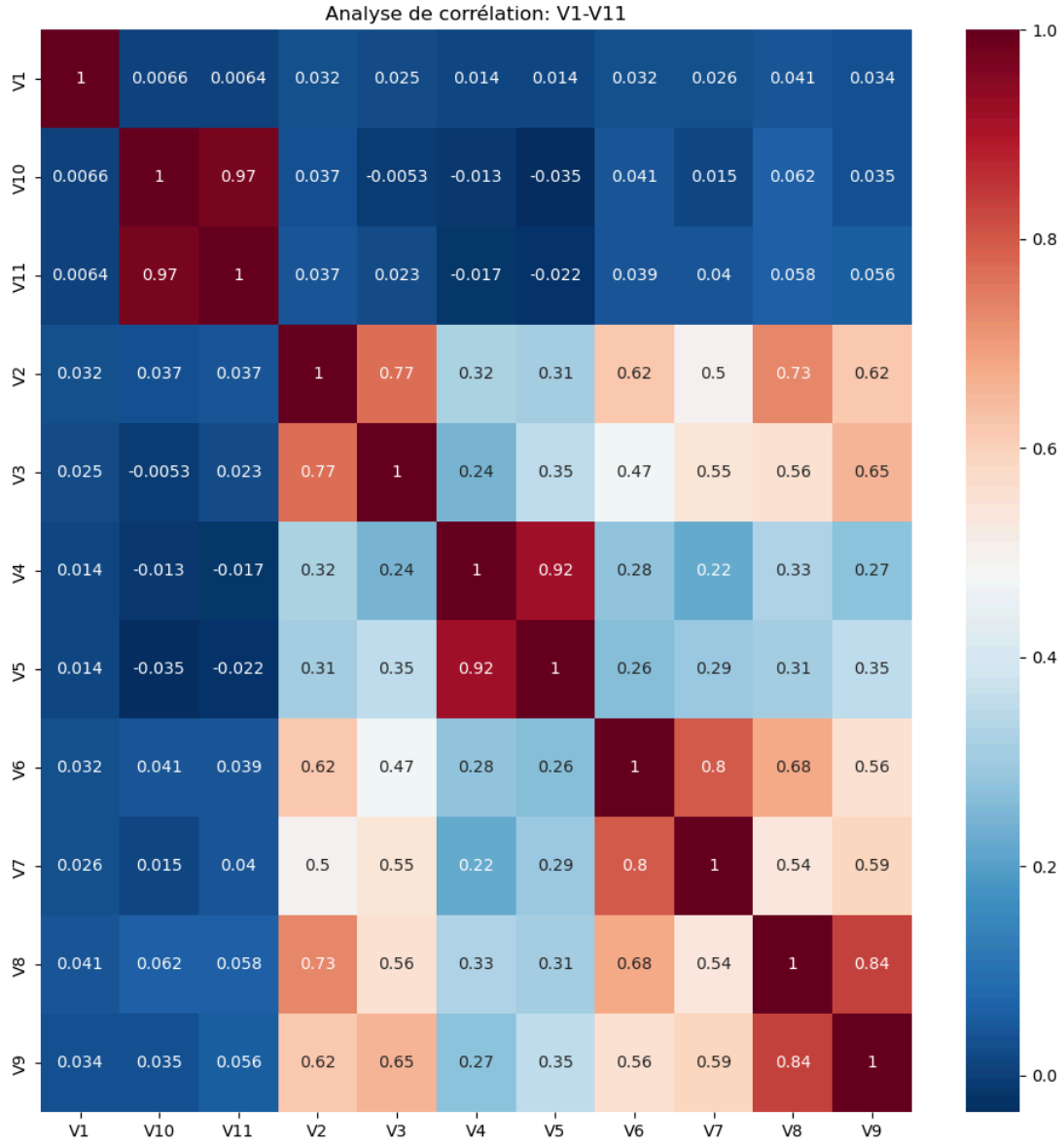
2. Dans chaque groupe, calculer la matrice de corrélation et identifier les paires fortement corrélées ($|corr| \geq 0{,}75$).

3. Fusionner les colonnes corrélées en sous-groupes et, dans chaque sous-groupe, conserver la colonne la plus informative (celle avec le plus de valeurs uniques).

Résultat attendu :
Une sélection de colonnes réduite et pertinente, permettant un modèle plus rapide, stable et performant.

Pour cette partie notre attention etait beaucoup plus fixée sur les colonnes V étant donné qu'elles sont les plus nombreuses.

Voici un exemple du travail fait:



Ici nous avons les pairs [['V1'],['V2','V3'],['V4','V5'],['V6','V7'],['V8','V9'],['V10','V11']], où chaque sous-groupe contenait les colonnes les plus corrélées entre eux, par exemple V2 et V3 sont très corrélées (0.77). Ensuite après regroupement on conserve la colonne avec le plus de valeurs unique pour apporter un maximum d'infos à nos modèles.
Après reduction on a ['V1', 'V3', 'V4', 'V6', 'V8', 'V11']

L'etape après selection de toutes ces colonnes etait de supprimer celles qui n'ont pas de réel valeurs ajouté pour les modèles.

Entrainement du modèle

L'objectif de cette étape est de définir un **modèle de référence** pour la détection de fraude, servant de point de comparaison pour toutes les améliorations futures.

Approche :
Nous avons utilisé trois algorithmes de **gradient boosting**, performants pour la classification sur des données structurées :

- **XGBoost** – reconnu pour sa précision et sa robustesse.
- **LightGBM** – rapide et optimisé pour de grands ensembles de données.
- **CatBoost** – efficace pour les variables catégorielles sans encodage complexe.

Étapes clés :

1. Préparation des données (encodage et sélection des features).
2. Entraînement des modèles sur l'ensemble d'entraînement.
3. Évaluation des performances avec des métriques adaptées à la fraude (Précision, Rappel, F1-score, AUC-ROC).
4. Comparaison des modèles pour identifier le baseline le plus performant.

Résultat attendu :
Un modèle de référence fiable, servant de benchmark pour les optimisations et développements ultérieurs.

Precision du type des données dans le DataFrame

Pour préparer les données avant l'entraînement du modèle :

- **Identification des colonnes catégorielles** (ProductCD, cartes, adresses, emails, variables M et Device) et conversion en type string.

- **Identification des colonnes numériques** restantes et conversion en type float64.

- **Gestion des valeurs manquantes** : suppression des colonnes ayant plus de 80 % de valeurs manquantes afin de simplifier le modèle et réduire le bruit.

Objectif :
Assurer une cohérence des types de données et réduire les colonnes inutiles pour un apprentissage plus efficace et rapide.

Encodage des variables catégorielles et Scaling des variables numerique

Encodage des variables catégorielles :

- Les colonnes catégorielles ont été transformées en entiers via pd.factorize() (Label Encoding).

- Cette approche est adaptée aux modèles basés sur les arbres (XGBoost, CatBoost, LightGBM), car elle conserve l'information de catégorie sans introduire d'ordre numérique artificiel.

Création de nouvelles features temporelles :

- Pour les colonnes D1 à D15, de nouvelles variables D1n à D15n ont été générées en recentrant chaque valeur par rapport à la date de la transaction (TransactionDT).

- Cette transformation permet aux modèles de mieux capturer les patterns temporels liés à la fraude.

Normalisation des variables numériques :

- Les colonnes numériques (hors identifiants et variables catégorielles) ont été mises à l'échelle avec un **Min-Max Scaling**.

- Les valeurs manquantes ont été remplacées par -1 pour éviter les erreurs lors de l'entraînement.

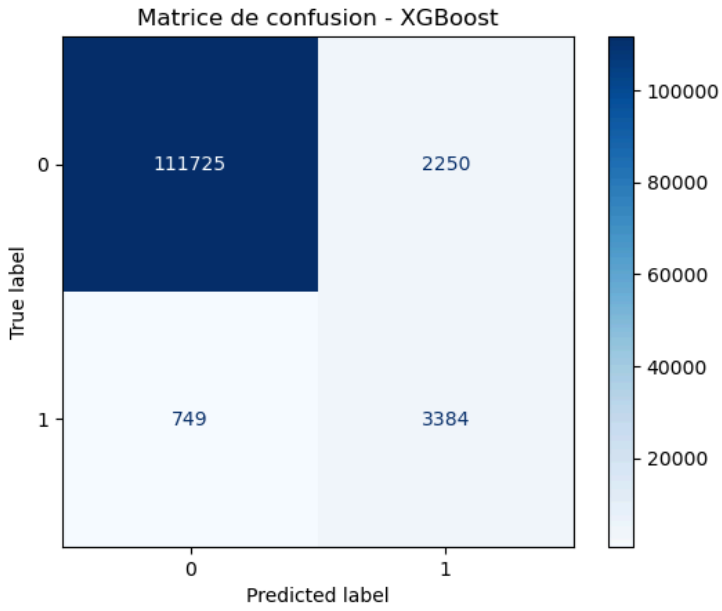
Objectif :
Garantir que les données sont correctement formatées, comparables et exploitables par les modèles, tout en enrichissant le dataset avec des informations temporelles pertinentes

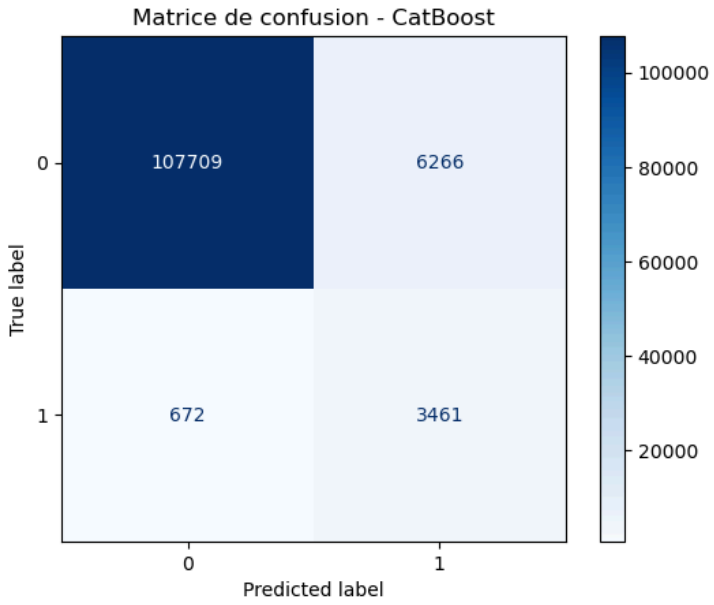
Comparaison des modèles

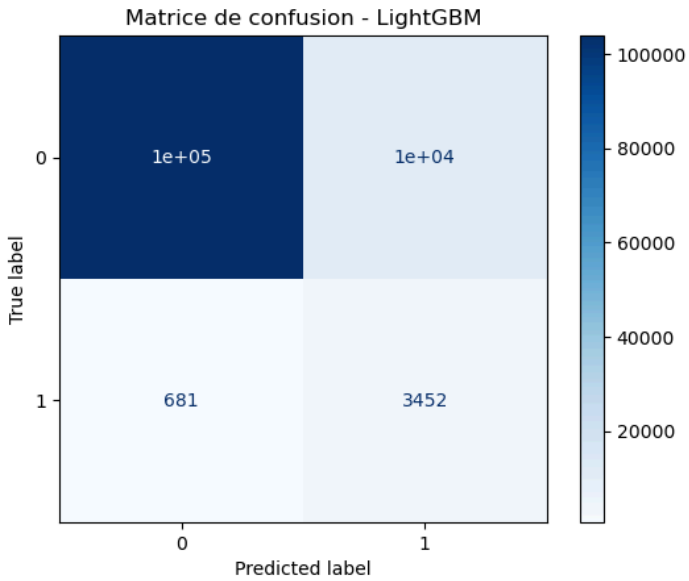
Modèle	AUC (ROC)	FPR (erreurs sur classe négative)	TPR (rappel de la classe positive)	Spécificité	Précision	Observations clés
XGBoost	**≈ 0.97**	Très faible	Très élevé	Excellent	Équilibré	Meilleur compromis entre précision et rappel
CatBoost	≈ 0.95	Moyen	Bon	Correct	Léger overfit possible	Bon mais un peu moins précis
LightGBM	≈ 0.94	Plus élevé	Bon mais légèrement inférieur	Moins bon	Stable mais moins performant	Plus rapide, mais perte de précision

La **matrice de confusion** montre que XGBoost réussit à détecter davantage de cas positifs tout en minimisant les faux positifs :

Modèle	TN	FP	FN	TP
XGBoost	111725	2250	749	3384
CatBoost	107709	6266	672	3461
LightGBM	103894	10081	681	3452


Matrice de confusion - XGBoost
111725
2250
749
3384
True label
Predicted label
0
1
100000
80000
60000
40000
20000


Matrice de confusion - CatBoost
107709
6266
672
3461
True label
Predicted label
0
1
100000
80000
60000
40000
20000



Choix du modèle final

Après comparaison :

- **XGBoost** obtient la **meilleure performance AUC** (≈ 0.97),
- Il offre le **meilleur équilibre** entre **rappel (TPR)** et **précision**,
- Et sa **gestion du déséquilibre via scale_pos_weight** s'est montrée plus efficace que les mécanismes équivalents des deux autres modèles.

Ainsi, **XGBoost a été sélectionné comme modèle final** pour la phase d'inférence et d'intégration dans l'API.