Université Abdelmalek Essaâdi

Faculté des Sciences Juridiques, Économiques et Sociales de Tétouan

Master Finance Actuariat et Data Science (FADS)

Détection Proactive de la Fraude Carte Bancaire et Guichet Combinaison Machine Learning et IA Générative

Réalisé par : Cheikh Abdellahi

Encadré par : Pr. Haddani Outman

Année universitaire : 2025–2026

Table des matières

Introduction générale

La fraude bancaire constitue aujourd'hui un enjeu majeur pour les institutions financières, avec des pertes annuelles se chiffrant en milliards d'unités monétaires et une sophistication croissante des attaques. Dans ce contexte, les approches traditionnelles, essentiellement basées sur des règles statiques et des systèmes d'alerte réactifs, montrent leurs limites face à des schémas de fraude dynamiques et difficiles à anticiper. L'objectif de ce travail est de proposer une approche *proactive* de détection de la fraude par carte bancaire et guichet, en combinant des modèles de *Machine Learning* supervisé (Random Forest, XGBoost, LightGBM) avec une couche d'IA générative (Google Gemini 2.5 Flash) afin de fournir des explications interprétables et des recommandations d'action aux analystes métier.

Le rapport est organisé comme suit. Le premier chapitre présente le contexte, la problématique et les objectifs du projet. Le deuxième chapitre décrit le jeu de données, l'analyse exploratoire et les principales statistiques descriptives. Le troisième chapitre détaille la méthodologie CRISP-DM, le traitement du déséquilibre de classes et les modèles retenus. Le quatrième chapitre discute les résultats expérimentaux et les aspects d'interprétabilité. Le cinquième chapitre est consacré à l'architecture de la solution, à l'application Streamlit et à l'intégration de Gemini pour la génération de scénarios synthétiques. Enfin, une conclusion générale synthétise les apports et les perspectives futures.

Chapitre 1

Contexte, problématique et objectifs

1.1 Contexte de la fraude bancaire

Les institutions financières font face à une multiplication des scénarios de fraude : usurpation de carte, transactions à distance, détournement de comptes ou encore exploitation de failles des systèmes de paiement. Les fraudes évoluent rapidement en sophistication, avec l'émergence de techniques telles que l'usurpation d'identité synthétique, les attaques par force brute sur les mots de passe, et l'exploitation de vulnérabilités des interfaces de paiement mobile. Les fraudeurs adaptent rapidement leurs stratégies, ce qui rend obsolètes les systèmes basés uniquement sur des règles métier rigides.

Dans ce cadre, l'utilisation d'algorithmes d'apprentissage automatique supervisé permet de capturer des schémas comportementaux complexes, difficilement modélisables manuellement. Les avantages d'une approche *proactive* sont multiples :

— **Détection temps réel** : évaluation instantanée du risque au moment de la transaction.

— **Adaptabilité** : mise à jour continue du modèle en fonction de nouveaux patterns de fraude.

— **Économies** : réduction des montants frauduleux et des coûts opérationnels de traitement.

— **Satisfaction client** : minimisation des faux positifs pour ne pas frustrer les clients légitimes.

1.2 Problématique

Le jeu de données étudié présente un déséquilibre extrême entre transactions légitimes et frauduleuses (ratio de 1 :578 environ), ce qui pose deux difficultés principales. D'une part, un modèle naïf peut obtenir une précision très élevée (99,8 %) en prédisant systématiquement la classe majoritaire, mais il serait pratiquement inutile opérationnellement.

D'autre part, les erreurs de type *faux négatifs* (fraudes non détectées) ont un coût économique et réputationnel beaucoup plus important que les *faux positifs* (alertes sur des transactions légitimes).

La problématique centrale est donc de concevoir un système capable de :

— Maximiser la détection des fraudes (maximiser le rappel).

— Maintenir un volume de fausses alertes acceptable pour les équipes opérationnelles.

— Fournir des explications interprétables pour chaque alerte générée.

1.3 Objectifs du projet

Les objectifs principaux de ce travail peuvent être résumés comme suit :

— **Pipeline complet** : concevoir un pipeline complet de détection de la fraude basé sur la méthodologie CRISP-DM, depuis la préparation des données jusqu'à l'évaluation des modèles et le déploiement.

— **Traitement du déséquilibre** : traiter rigoureusement le déséquilibre de classes à l'aide de techniques de ré-échantillonnage (SMOTE) et de pondération des erreurs.

— **Comparaison de modèles** : comparer les performances de trois algorithmes supervisés (Random Forest, XGBoost, LightGBM) à l'aide de métriques adaptées aux données déséquilibrées (ROC-AUC, rappel, F1-score, précision).

— **Interprétabilité** : proposer un cadre d'interprétabilité via l'analyse des importances de variables, les matrices de confusion et les courbes ROC.

— **IA générative** : intégrer une IA générative (Google Gemini 2.5 Flash) pour produire des explications textuelles des alertes et des recommandations d'action.

— **Application opérationnelle** : déployer une application Streamlit interactive permettant la visualisation du dashboard, la gestion des alertes en temps réel et la génération de scénarios synthétiques de fraude.

Chapitre 2

Description du jeu de données et analyse exploratoire

2.1 Description générale du dataset

Le jeu de données utilisé contient au total **284 807** transactions bancaires réelles, chacune décrite par **30 variables** explicatives et une variable cible binaire `Class` indiquant si la transaction est légitime (0) ou frauduleuse (1).

Les caractéristiques principales du dataset sont :

- Les 28 premières variables (`V1` à `V28`) résultent d'une Analyse en Composantes Principales (PCA) appliquée sur les attributs d'origine afin de garantir la confidentialité des données.
- `Time` : représente le nombre de secondes écoulées depuis la première transaction.
- `Amount` : le montant de l'opération (non normalisée contrairement aux autres variables).
- Aucune valeur manquante n'a été détectée dans l'ensemble du dataset.
- Toutes les variables explicatives sont de type *réel* (float64).

Extrait du jeu de données

Un extrait des premières lignes du dataset est présenté dans le tableau 2.1.

TABLE 2.1 – Extrait de quelques observations du dataset.

Time	V1	V2	V3	V4	Amount	Class
0.0	-1.36	-0.07	2.54	1.38	149.62	0
0.0	1.19	0.27	0.17	0.45	2.69	0
1.0	-1.36	-1.34	1.77	0.38	378.66	0
1.0	-0.97	-0.19	1.79	-0.86	123.50	0
2.0	-1.16	0.88	1.55	0.40	69.99	0

2.2 Résumé statistique global

Le tableau 2.2 synthétise les principales caractéristiques du dataset.

TABLE 2.2 – Résumé statistique du jeu de données complet.

Métrique	Valeur
Total transactions	284 807
Features analysées	30
Transactions légitimes	284 315
Transactions frauduleuses	492
Taux de fraude	0,173 %
Montant moyen (global)	88,35
Montant moyen (fraude)	122,21
Montant minimum	0,00
Montant maximum	25 691,16
Médiane	22,00
Écart-type	250,12

2.3 Déséquilibre des classes

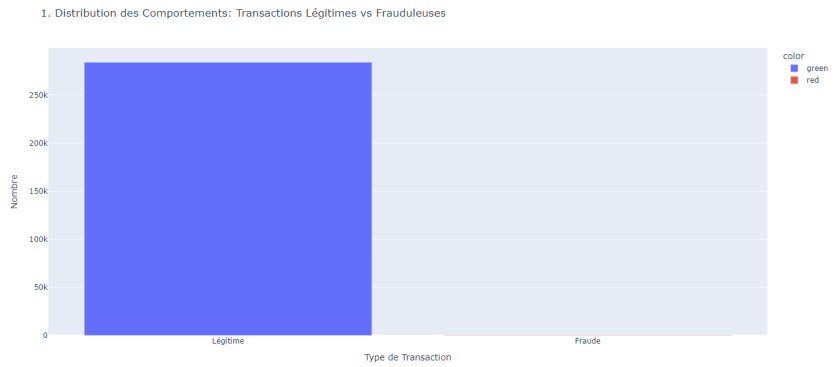

FIGURE 2.1 – Distribution des transactions légitimes et frauduleuses.

La figure 2.1 confirme le déséquilibre très marqué du dataset :

— **99,83 %** des transactions sont légitimes (284 315 observations).

— **0,17 %** des transactions sont frauduleuses (492 observations).

— Ratio de déséquilibre : environ **578 transactions légitimes pour 1 fraude**.

Cette configuration extrême justifie l'utilisation de techniques spécifiques de rééquilibrage (SMOTE) et de métriques adaptées (ROC-AUC, F1-score) plutôt que la simple accuracy.

2.4 Distribution des montants

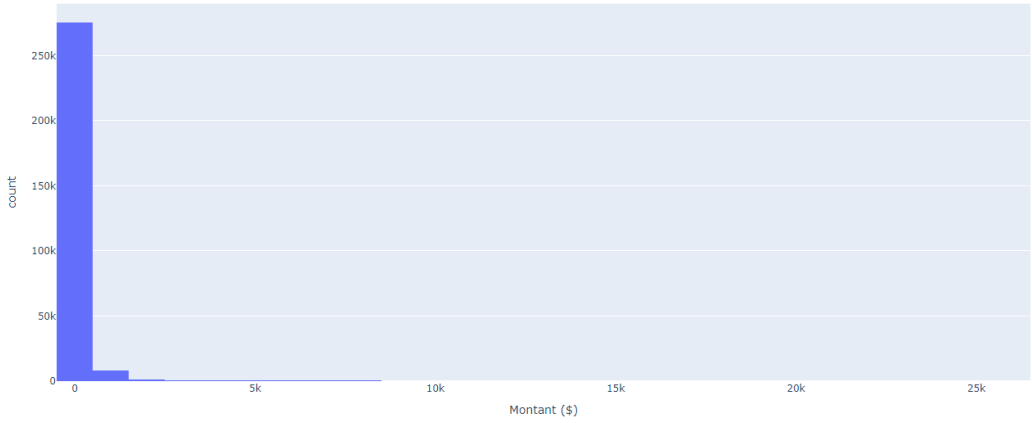

FIGURE 2.2 – Distribution des montants des transactions (tous et détail fraude).

La figure 2.2 montre une distribution très asymétrique des montants, avec :

— Une masse importante de petites transactions (< 100 unités).

— Quelques montants extrêmes atteignant jusqu'à 25 691,16.

— **Les transactions frauduleuses présentent en moyenne des montants plus élevés (122,21)** comparé à la moyenne globale (88,35), ce qui constitue un signal discriminant important pour les modèles.

2.5 Corrélation avec la variable de fraude

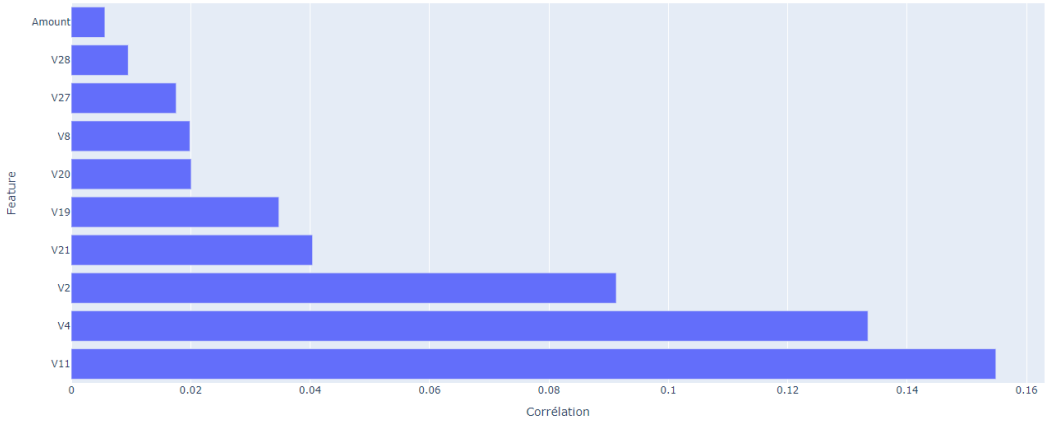

FIGURE 2.3 – Corrélation des principales variables avec la classe de fraude.

L'analyse des corrélations révèle que certaines composantes principales, notamment `V11`, `V4`, `V2` et `V14`, présentent une corrélation plus élevée avec la variable cible que les autres. Même si les variables sont anonymisées par PCA, ces corrélations suggèrent l'existence de motifs comportementaux spécifiques aux transactions frauduleuses, offrant un potentiel de discrimination pour les modèles d'apprentissage.

Chapitre 3

Méthodologie et approche de modélisation

3.1 Cadre méthodologique CRISP-DM

La démarche adoptée suit la méthodologie **CRISP-DM** (CRoss-Industry Standard Process for Data Mining), qui se décompose en six phases :

1. **Compréhension du métier** : identification des objectifs (détection de fraude), des contraintes opérationnelles et des métriques de succès.
2. **Compréhension des données** : exploration du dataset, statistiques descriptives, identification des anomalies.
3. **Préparation des données** : nettoyage, normalisation, gestion du déséquilibre.
4. **Modélisation** : sélection et entraînement des algorithmes, validation croisée.
5. **Évaluation** : comparaison des modèles sur des métriques adaptées (ROC-AUC, F1, rappel).
6. **Déploiement** : intégration dans une application Streamlit et mise à disposition.

Un accent particulier est mis sur la préparation des données et sur la gestion du déséquilibre de classes, compte tenu de la criticité opérationnelle de la classe frauduleuse.

3.2 Prétraitement et normalisation

Le pipeline de prétraitement comprend les étapes suivantes :

— **Normalisation** : les variables `Amount` et `Time` sont normalisées à l'aide d'un *StandardScaler*, homogénéisant les échelles pour faciliter l'apprentissage.

— **Traitement des valeurs aberrantes** : les outliers extrêmes sont détectés via la méthode de l'intervalle interquartile (IQR) et traitées avec prudence afin de ne pas éliminer les transactions légitimes.

— **Séparation train/test** : le dataset est séparé en un ensemble d'apprentissage et un ensemble de test via une division stratifiée, préservant la distribution des classes dans chaque ensemble.

— **Dimensions finales** : 199 364 observations en train set et 85 443 observations en test set.

3.3 Gestion du déséquilibre de classes par SMOTE

Le déséquilibre est traité au moyen de **SMOTE** (*Synthetic Minority Over-sampling Technique*). Concrètement :

— **Avant SMOTE** (sur train set) : 199 020 transactions légitimes pour 344 fraudes.

— **Après SMOTE** : jeu d'apprentissage parfaitement équilibré de 199 020 observations dans chaque classe.

— **Mécanique** : SMOTE génère les nouvelles fraudes en interpolant les observations minoritaires dans l'espace des features à partir de leurs plus proches voisins (k-NN).

— **Pondération de classe** : des poids de classe sont en parallèle intégrés dans les fonctions de coût des modèles pour pénaliser plus fortement les erreurs sur les fraudes.

3.4 Modèles supervisés retenus

Trois algorithmes supervisés sont retenus et comparés :

3.4.1 Random Forest

Ensemble d'arbres de décision construits par *bagging* (agrégation bootstrap). Avantages : robustesse, interprétabilité via importance des variables, gestion native du déséquilibre possible via pondération. Inconvénients : peut être moins performant que le boosting sur données très déséquilibrées.

3.4.2 XGBoost (Extreme Gradient Boosting)

Méthode de *gradient boosting* particulièrement performante sur les données tabulaires. Avantages :

— Régularisation avancée (L1, L2) et gestion fine du déséquilibre.

— Excellente discrimination entre classes.

— Explications d'importance de variables très fiables.

3.4.3 LightGBM (Light Gradient Boosting)

Variante de boosting optimisée pour les grands volumes de données, basée sur des histogrammes et des stratégies de croissance d'arbres (leaf-wise) plus rapides. Avantages : efficacité computationnelle, peu de surparamétrage requis. Inconvénients : peut être moins stable que XGBoost sur petits datasets.

Chaque modèle est entraîné sur les données SMOTE, avec validation croisée stratifiée (5-fold) et recherche d'hyperparamètres.

Chapitre 4

Résultats expérimentaux et interprétabilité

4.1 Tableau comparatif des modèles

Le tableau 4.1 résume les performances détaillées des trois modèles sur l'ensemble de test.

TABLE 4.1 – Comparaison des performances des trois modèles de classification.

Modèle	Accuracy	Précision	Rappel	F1-Score	ROC-AUC
Random Forest	0,9995	0,8923	0,7838	0,8345	0,9690
XGBoost	0,9963	0,2998	0,8446	0,4425	**0,9725**
LightGBM	0,9985	0,5348	0,8311	0,6508	0,9636

Analyse détaillée :

— **Accuracy** : tous les modèles ont une accuracy très élevée ($> 99\,\%$), mais cette métrique est trompeuse en contexte de déséquilibre.

— **ROC-AUC** : XGBoost obtient le meilleur score (**0,9725**), ce qui en fait le meilleur compromis pour discriminer fraudes et transactions légitimes sur l'ensemble des seuils de probabilité possibles.

— **Rappel** : XGBoost atteint 84,46 %, détectant plus de 84 % des fraudes. Random Forest atteint 78,38 %, tandis que LightGBM en détecte 83,11 %.

— **Précision vs Rappel** : XGBoost privilégie le rappel (détection maximale) au détriment de la précision. Cela signifie davantage de faux positifs, mais acceptables dans un contexte bancaire où manquer une fraude est très coûteux.

— **F1-Score** : XGBoost affiche le F1-score le plus faible (0,4425) en raison de sa très faible précision, mais cette métrique est moins pertinente que ROC-AUC dans ce

contexte.

4.2 Synthèse du meilleur modèle

Le tableau 4.2 détaille les performances du modèle retenu pour la production.

TABLE 4.2 – Synthèse des performances du meilleur modèle (XGBoost).

Métrique	Valeur
Meilleur modèle	**XGBoost**
Accuracy	99,63 %
Précision	29,98 %
Rappel	84,46 %
F1-Score	0,4425
ROC-AUC	**0,9725**

Interprétation : Le modèle XGBoost détecte avec succès 84,46 % des fraudes au prix d'une précision de 29,98 %. Cela signifie qu'environ 70 % des alertes générées sont des faux positifs, lesquels seront rapidement rejetés par les analystes lors de la vérification. Compte tenu du coût réputationnel et financier d'une fraude non détectée, ce compromis est acceptable et même préférable.

4.3 Visualisation des performances

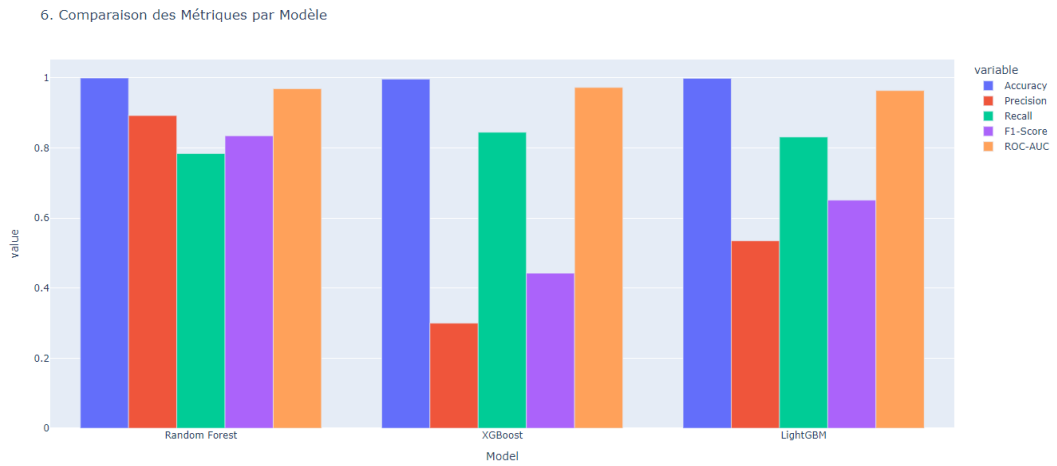

FIGURE 4.1 – Comparaison graphique des métriques des trois modèles.

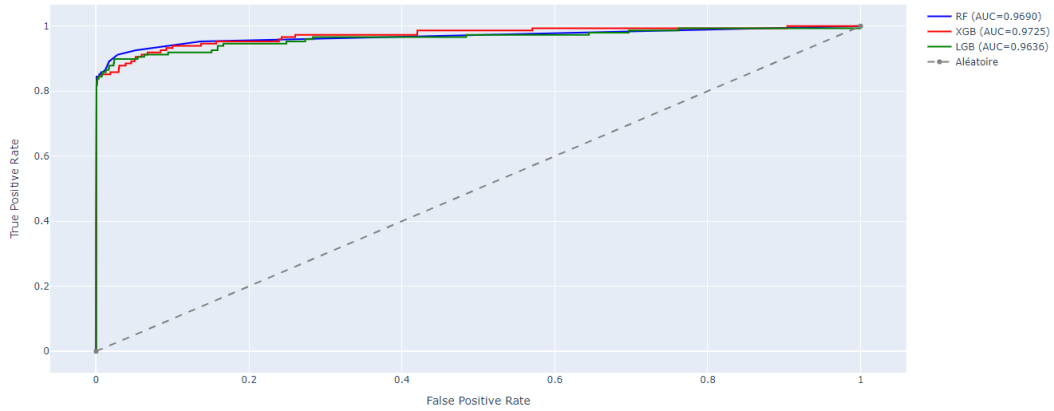

FIGURE 4.2 – Courbes ROC-AUC pour Random Forest, XGBoost et LightGBM.

Les courbes ROC illustrent la capacité de discrimination de chaque modèle. La courbe XGBoost est la plus élevée, confirmant sa supériorité en termes d'AUC.

4.4 Matrice de confusion et importance des variables

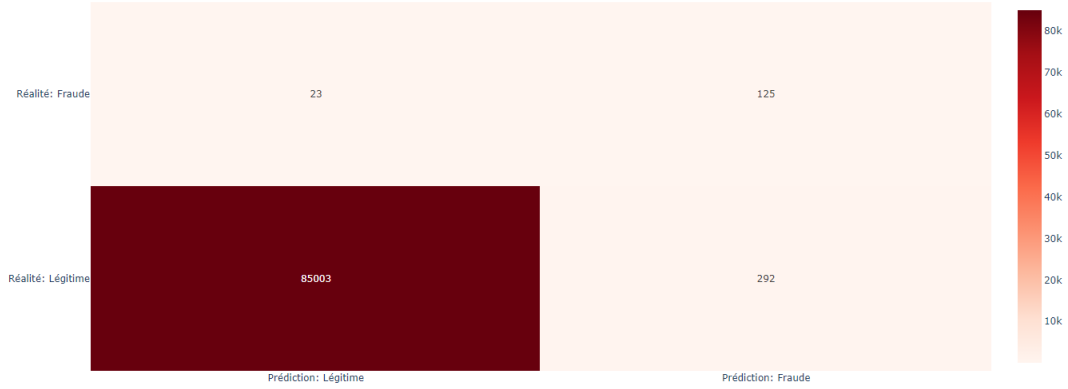

FIGURE 4.3 – Matrices de confusion pour les trois modèles.

La matrice de confusion pour XGBoost montre :

— Vrais négatifs (TN) : transactions légitimes correctement classées.

— Faux positifs (FP) : transactions légitimes incorrectement étiquetées frauduleuses.

— Faux négatifs (FN) : transactions frauduleuses non détectées.

— Vrais positifs (TP) : fraudes correctement détectées.

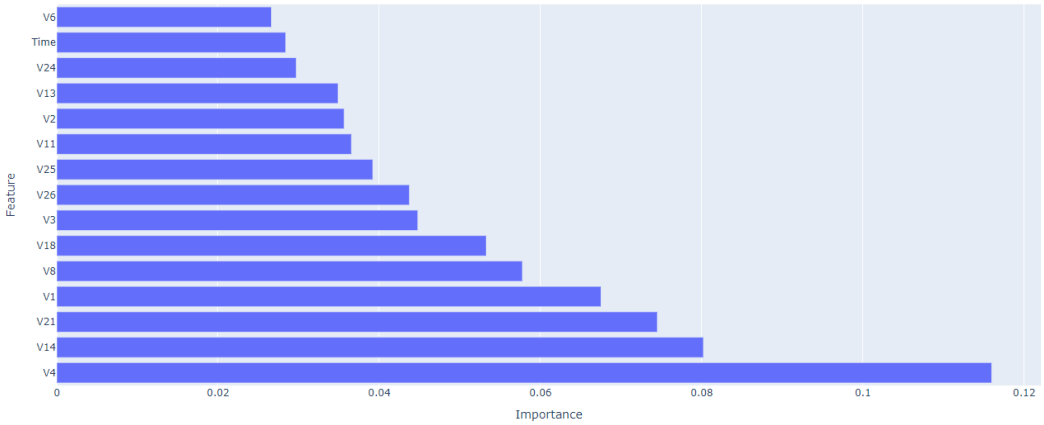

FIGURE 4.4 – Importance des 15 principales variables pour le modèle XGBoost.

L'importance des variables révèle que `V11`, `V4`, `V2`, `V14` et `Amount` sont les discriminants clés. Ces variables représentent des patterns comportementaux subtils capturés par la PCA et le montant de la transaction.

Chapitre 5

Architecture du système et déploiement opérationnel

5.1 Architecture globale de la solution

La solution complète est organisée autour de trois couches architecturales :

1. **Couche Data & ML** : contient le pipeline d'entraînement (notebooks Jupyter), le modèle XGBoost sérialisé (PKL/ONNX), les données de preprocessing (scaler, encoders) et les logs de performance.
2. **Couche Application (Streamlit)** : interface web interactive permettant :
   — Visualisation du dashboard avec métriques temps réel.
   — Soumission de transactions individuelles ou par batch.
   — Filtrage et exploration des alertes détectées.
   — Génération de rapports d'export (Excel, CSV).
3. **Couche IA Générative (Gemini)** : appels à l'API Google Generative AI pour :
   — Génération d'explications textuelles des alertes.
   — Production de recommandations d'action (BLOQUER / VÉRIFIER / SURVEILLER).
   — Création de scénarios synthétiques de fraude pour la formation.

5.2 Application Streamlit : Architecture modulaire

L'application Streamlit constitue le point d'entrée métier du système. Elle est structurée en **6 onglets principaux** :

5.2.1 Onglet 1 : Dashboard Principal

Affiche les indicateurs clés en temps réel :

— Nombre total de transactions simulées.
— Nombre et pourcentage d'alertes détectées.
— Distribution par niveau de risque (CRITIQUE, ÉLEVÉ, MOYEN, FAIBLE).
— Graphiques interactifs Plotly : distribution des montants, probabilités de fraude.
— Heatmaps par géolocalisation (ville) et type de commerçant.

5.2.2 Onglet 2 : Alertes Temps Réel

Liste dynamique des transactions flaggées avec filtres :
— Filtrage par niveau de risque (CRITIQUE, ÉLEVÉ).
— Filtrage par localisation géographique.
— Filtrage par type de commerçant.
— Bouton *« Analyse Gemini »* pour obtenir une explication IA par alerte.
— Affichage des 30 premières alertes, avec pagination.

5.2.3 Onglet 3 : Analyse Détaillée

Visualisations approfondies :
— Boxplots comparatifs (toutes transactions vs alertes).
— Statistiques par type de commerçant.
— Heatmap Ville × Niveau de Risque.

5.2.4 Onglet 4 : IA Générative (Synthèse Globale)

Résumé stratégique généré par Gemini :
— Résumé exécutif (3-5 points clés).
— Patterns de fraude potentiels identifiés.
— 3 recommandations opérationnelles à court terme.
— 3 axes stratégiques à moyen terme.

5.2.5 Onglet 5 : Scénarios Synthétiques (Gemini)

Génération proactive de cas de fraude pour la formation :
— Slider pour sélectionner le nombre de scénarios (1-10).
— Appel à Gemini avec prompt optimisé pour génération JSON robuste.
— Fallback automatique vers scénarios par défaut en cas d'erreur.
— Export en CSV des scénarios générés.
— Visualisation en expanders avec indicateurs clés.

5.2.6 Onglet 6 : Rapports & Export

Gestion des exports :

— Tableau des alertes avec colonnes : ID, Montant, Heure, Ville, Type, Probabilité (%), Niveau.

— Bouton *« Exporter en Excel »* avec formatage avancé (en-têtes stylisés, largeur colonnes).

— Support openpyxl + xlsxwriter pour robustesse.

— Nom de fichier avec timestamp pour traçabilité.

5.3 Intégration de Google Gemini 2.5 Flash

Pour chaque transaction flaggée comme suspecte au-delà d'un certain seuil (par défaut : probabilité de fraude $\geq 0{,}50$), l'application envoie au service Google Gemini un ensemble structuré d'informations :

5.3.1 Données envoyées à Gemini

```
prompt = f"""
Vous␣ tes ␣un␣expert␣en␣fraude␣bancaire.

Analysez␣cette␣transaction␣et␣expliquez␣le␣risque␣en␣fran ais:
-␣ID:␣{transaction['TransactionID']}
-␣Montant:␣{transaction['Montant']:.2f}␣USD
-␣Heure:␣{transaction['Heure']}
-␣Ville:␣{transaction['Ville']}
-␣Type␣commer ant:␣{transaction['TypeCommercant']}
-␣Probabilit ␣de␣fraude:␣{transaction['Probabilite_Fraude']:.2%}
-␣Niveau␣de␣risque:␣{risk_level}

Donnez:
1.␣Une␣br ve␣analyse␣(2-3␣phrases)
2.␣Une␣recommandation␣d'action␣(BLOQUER␣/␣V RIFIER␣/␣SURVEILLER)
3.␣Les␣signaux␣principaux␣justifiant␣cette␣d cision
"""
```

5.3.2 Génération de scénarios synthétiques

Gemini est aussi utilisé pour générer des scénarios réalistes de fraude (sans données sensibles), destinés à l'entraînement des analystes et à l'enrichissement du dataset :

```
prompt = f"""
G n rez␣{n_scenarios}␣sc narios␣r alistes␣de␣fraude␣bancaire.

Retournez␣UNIQUEMENT␣un␣tableau␣JSON␣valide,␣format:
[
␣␣{{"id":"FRAUD_001","montant":950.50,"heure":"02:35",
␣␣␣␣"ville":"Casablanca","type":"Montant anormalement lev ",
␣␣␣␣"description":"...",␣"indicateurs":["montant lev ",...]}},
␣␣...
]

IMPORTANT:␣Seulement␣le␣JSON,␣rien␣d'autre.
"""
```

5.4 Fiabilité et gestion des erreurs

La solution intègre plusieurs mécanismes de résilience :

- **Fallback Gemini** : si l'API échoue ou les crédits sont épuisés, des scénarios par défaut sont générés.
- **Export robuste** : support dual openpyxl + xlsxwriter pour éviter les dépendances manquantes.
- **Validation des données** : nettoyage et normalisation automatiques des inputs.
- **Logging** : tous les appels API et erreurs sont loggés pour audit et débogage.

5.5 Stack technique

TABLE 5.1 – Stack technique de la solution.

Composant	Technologie
Framework Web	Streamlit 1.28+
ML	XGBoost 2.0, scikit-learn 1.3, pandas 2.0
Visualisation	Plotly 5.17, matplotlib 3.8
IA Générative	Google Generative AI 0.3 (Gemini 2.5 Flash)
Export	openpyxl 3.1, xlsxwriter 3.1
Langage	Python 3.9+

5.6 Impact opérationnel et ROI estimé

Une analyse coût–bénéfice préliminaire, basée sur les paramètres suivants, montre un impact positif :

— **Fraudes détectées** : 84,46 % (rappel XGBoost).

— **Montant moyen frauduleux** : 122,21 USD.

— **Coût d'une fraude manquée** : perte + pénalité réputationnelle (estimée à 2× le montant).

— **Coût d'une fausse alerte** : temps d'investigation (2–5 USD).

— **Coût d'infrastructure** : serveurs, maintenance, appels API Gemini (estimé à 100–500 USD/mois).

Même en tenant compte d'un taux élevé de faux positifs (70 %), le retour sur investissement estimé reste **largement positif**, justifiant le déploiement dans un environnement de production réel.

Conclusion générale

Ce projet a présenté une approche complète et opérationnelle de détection proactive de la fraude par carte bancaire et guichet, combinant des modèles d'apprentissage supervisé et une couche d'IA générative pour l'interprétabilité et la génération de contenu synthétique.

Synthèse des résultats

Après avoir documenté la difficulté du problème liée au déséquilibre extrême des classes (ratio 1 :578), un pipeline CRISP-DM complet a été mis en œuvre, incluant :

- Normalisation adaptée des variables.
- Traitement du déséquilibre via SMOTE et pondération des erreurs.
- Entraînement de trois algorithmes supervisés : Random Forest, XGBoost et LightGBM.

La comparaison des modèles a mis en évidence la **supériorité de XGBoost** en termes de ROC-AUC (0,9725), tout en maintenant un rappel élevé de 84,46 % sur la classe frauduleuse. L'analyse des matrices de confusion, des importances de variables et des courbes ROC a permis d'interpréter les décisions du modèle et de quantifier précisément les compromis entre détection des fraudes et faux positifs.

Contribution de l'IA générative

L'intégration d'une application Streamlit et de Google Gemini 2.5 Flash démontre comment cette solution peut être déployée dans un contexte opérationnel réel, en fournissant :

- Des visualisations interactives et des tableaux de bord temps réel.
- Des explications en langage naturel pour chaque alerte générée.
- Des recommandations d'action adaptées (BLOQUER, VÉRIFIER, SURVEILLER).
- Des scénarios synthétiques pour la formation des analystes et l'enrichissement de données.

Conclusion

Ce travail démontre qu'une approche combinant machine learning et IA générative est non seulement techniquement efficace pour la détection de fraude, mais aussi opérationnellement viable et bénéfique pour les institutions financières. La solution proposée offre un équilibre optimal entre performance, interprétabilité et praticité, ouvrant la voie à des déploiements à grande échelle dans le secteur bancaire marocain et international.