Détection de Fraude par Carte de Crédit : Une Analyse par Apprentissage Automatique

Jules Odje

December 8, 2024

Abstract

Cette étude présente une analyse approfondie de la détection de fraude par carte de crédit utilisant des techniques d'apprentissage automatique. L'analyse se concentre sur un jeu de données fortement déséquilibré, comprenant plus de 284,000 transactions, dont seulement 0.17% sont frauduleuses. Plusieurs modèles ont été évalués, avec une attention particulière portée au traitement du déséquilibre des classes et à l'optimisation des performances de détection.

Contents

1 Introduction

1.1 Contexte

La fraude par carte de crédit représente un défi majeur pour les institutions financières, avec des pertes annuelles estimées à plusieurs milliards de

dollars. La détection automatique de ces fraudes est cruciale mais complexe, notamment en raison du fort déséquilibre entre transactions légitimes et frauduleuses.

1.2 Objectifs

Les objectifs principaux de cette étude sont :

- Analyser et préparer un jeu de données fortement déséquilibré
- Développer et évaluer des modèles de détection de fraude
- Comparer différentes approches de modélisation
- Proposer des améliorations potentielles

2 Données et Méthodologie

2.1 Description du Dataset

Le jeu de données utilisé provient d'une collection de transactions par carte de crédit effectuées en septembre 2013 par des titulaires de cartes européens. Ce dataset présente plusieurs caractéristiques importantes :

- **Volume** : 284,807 transactions sur une période de deux jours
- **Variables** : 30 variables dont :
  - 28 composantes principales (V1-V28) obtenues par PCA
  - Le montant de la transaction ('Amount')
  - La classe (0 pour normale, 1 pour frauduleuse)
- **Déséquilibre** : 492 transactions frauduleuses (0.17%) contre 284,315 transactions normales (99.83%)

Les variables V1-V28 sont déjà le résultat d'une ACP (Analyse en Composantes Principales) pour des raisons de confidentialité, rendant impossible l'identification des variables originales.

2.2 Analyse Exploratoire

2.2.1 Distribution des Classes

La distribution extrêmement déséquilibrée des classes constitue un défi majeur :

Classe	Nombre	Pourcentage
Non-frauduleuse (0)	284,315	99.83%
Frauduleuse (1)	492	0.17%

Table 1: Distribution des classes dans le dataset

2.2.2 Analyse de la Variable Amount

La variable Amount présente les caractéristiques suivantes :

- Moyenne : 88.38
- Écart-type : 253.07
- Minimum : 0.00
- Maximum : 25,691.16

Ces statistiques montrent une grande dispersion des montants de transaction, justifiant la nécessité d'une standardisation.

2.3 Prétraitement des Données

2.3.1 Standardisation

Seule la variable Amount a nécessité une standardisation, les autres variables étant déjà normalisées par PCA. La standardisation a été effectuée selon la formule :

$$X_{standardis} = \frac{X - \mu}{\sigma}$$

Après standardisation :

- Moyenne 0
- Écart-type 1
- Distribution centrée-réduite

2.3.2 Gestion du Déséquilibre

Pour gérer le fort déséquilibre des classes, nous avons appliqué la technique SMOTE (Synthetic Minority Over-sampling Technique) :

- **Principe** : Création synthétique d'exemples de la classe minoritaire

- **Application** : Uniquement sur les données d'entraînement
- **Résultat** : Distribution équilibrée (50%-50%) dans l'ensemble d'entraînement

2.3.3 Partition des Données

La séparation des données a été effectuée comme suit :

- 70% pour l'entraînement
- 30% pour le test
- Stratification maintenue pour préserver la proportion de fraudes

3 Modélisation

3.1 Approches Considérées

Dans cette étude, trois modèles d'apprentissage automatique ont été évalués :

3.1.1 Random Forest

- **Configuration** :
  - class_weight = 'balanced'
  - n_estimators = 100
  - random_state = 42
- **Performance observée** :
  - Précision (classe 1) : 0.88
  - Recall (classe 1) : 0.77
  - F1-score (classe 1) : 0.82

3.1.2 Gradient Boosting

- **Configuration** :
  - random_state = 42
  - paramètres par défaut
- **Performance observée** :

- Précision (classe 1) : 0.11
- Recall (classe 1) : 0.86
- F1-score (classe 1) : 0.19

3.1.3 XGBoost

- **Configuration** :
  - scale_pos_weight = 1
  - random_state = 42
- **Performance observée** :
  - Précision (classe 1) : 0.74
  - Recall (classe 1) : 0.80
  - F1-score (classe 1) : 0.77

3.2 Évaluation des Modèles

3.2.1 Métriques d'Évaluation

Les modèles ont été évalués selon plusieurs métriques :

Métrique	Random Forest	Gradient Boosting	XGBoost
F1-score global	0.756	0.19	0.77
AUROC	0.945	-	-
AUPRC	0.792	-	-

Table 2: Comparaison des performances des modèles

3.2.2 Analyse des Courbes ROC et PRC

Les courbes ROC (Receiver Operating Characteristic) et PRC (Precision-Recall) ont été générées pour évaluer les performances des modèles :

[Insérer image des courbes ROC et PRC]

Figure 1: Courbes ROC et PRC pour le modèle Random Forest

3.2.3 Sélection du Meilleur Modèle

Le Random Forest a été sélectionné comme meilleur modèle pour les raisons suivantes :

- Meilleur équilibre entre précision et recall
- F1-score le plus élevé pour la détection des fraudes
- Excellente performance globale (AUROC = 0.945)
- Bonne gestion du déséquilibre des classes

3.3 Pipeline Final

Le pipeline final inclut :

- Standardisation de la variable Amount
- Application de SMOTE sur les données d'entraînement
- Sélection des meilleures features (SelectKBest)
- Classification par Random Forest

4 Résultats et Discussion

4.1 Comparaison Détaillée des Modèles

4.1.1 Performances par Classe

Modèle	Classe	Précision	Recall	F1-score
2*Random Forest	Non-fraude (0)	1.00	1.00	1.00
	Fraude (1)	0.88	0.77	0.82
2*Gradient Boosting	Non-fraude (0)	1.00	0.99	0.99
	Fraude (1)	0.11	0.86	0.19
2*XGBoost	Non-fraude (0)	1.00	1.00	1.00
	Fraude (1)	0.74	0.80	0.77

Table 3: Métriques détaillées par classe et par modèle

4.2 Analyse des Performances

4.2.1 Analyse du Random Forest

Le Random Forest a démontré les meilleures performances globales :

- **Forces :**
  - Excellente détection des transactions normales
  - Haute précision dans la détection des fraudes (0.88)
  - Bon équilibre entre précision et recall
- **Limitations :**
  - Recall modéré pour les fraudes (0.77)
  - Complexité computationnelle plus élevée

4.2.2 Analyse du Gradient Boosting

Le Gradient Boosting a montré des résultats mitigés :

- **Forces :**
  - Excellent recall pour les fraudes (0.86)
  - Bonne performance sur les transactions normales
- **Limitations :**
  - Très faible précision sur les fraudes (0.11)
  - Nombreux faux positifs

4.2.3 Analyse du XGBoost

XGBoost a fourni des performances équilibrées :

- **Forces :**
  - Bon équilibre général des métriques
  - Performance stable sur les deux classes
- **Limitations :**
  - Performances légèrement inférieures au Random Forest

4.3 Implications Pratiques

4.3.1 Coûts des Erreurs

L'analyse des erreurs de classification révèle :

- **Faux Positifs :**
  - Impact sur l'expérience client
  - Coûts opérationnels de vérification
- **Faux Négatifs :**
  - Pertes financières directes
  - Risques de sécurité

4.4 Limites de l'Étude

- Données limitées à une période de deux jours
- Variables originales masquées par PCA
- Absence de validation temporelle
- Possible sur-apprentissage dû au SMOTE

4.5 Recommandations

- **Choix du Modèle :** Utiliser le Random Forest comme modèle principal
- **Seuil de Décision :** Ajuster selon le compromis souhaité entre précision et recall
- **Monitoring :** Mettre en place un suivi des performances en production
- **Mise à Jour :** Prévoir des réentraînements réguliers du modèle

5 Conclusions et Perspectives

5.1 Synthèse Générale

Cette étude a permis de développer un système de détection de fraude par carte de crédit avec des résultats prometteurs :

- Performance globale élevée (AUROC = 0.945)
- Gestion efficace du déséquilibre des classes
- Identification fiable des transactions frauduleuses (F1-score = 0.82)
- Pipeline robuste et reproductible

5.2 Contributions Principales

5.2.1 Méthodologiques

- Approche systématique du traitement des données déséquilibrées
- Comparaison approfondie de différents algorithmes
- Évaluation multi-métrique des performances

5.2.2 Pratiques

- Pipeline complet de détection de fraude
- Métriques adaptées au contexte métier
- Recommandations pour l'implémentation

5.3 Limitations

5.3.1 Données

- Période d'observation limitée (2 jours)
- Variables transformées par PCA
- Absence de contexte temporel

5.3.2 Modélisation

- Possible sur-apprentissage avec SMOTE
- Complexité computationnelle du Random Forest
- Absence de validation temporelle

5.4 Perspectives d'Amélioration

5.4.1 Améliorations Techniques

- **Optimisation des Hyperparamètres :**
  - Recherche par grille plus exhaustive
  - Validation croisée temporelle
  - Tests de différentes architectures
- **Feature Engineering :**
  - Création de variables temporelles
  - Agrégations par profil client
  - Interactions entre variables
- **Ensemble Learning :**
  - Combinaison de différents modèles
  - Stacking/Blending
  - Votes pondérés

5.4.2 Perspectives de Recherche

- **Approches Alternatives :**
  - Deep Learning
  - Apprentissage par renforcement
  - Détection d'anomalies non supervisée
- **Aspects Métier :**
  - Analyse des coûts d'erreur
  - Profilage comportemental
  - Détection en temps réel

5.5 Recommandations Finales

- **Implémentation :**
  - Déploiement progressif
  - Monitoring continu
  - Mise à jour régulière des modèles
- **Validation :**
  - Tests en conditions réelles
  - Évaluation des coûts opérationnels
  - Mesure de l'impact business
- **Maintenance :**
  - Processus de réentraînement
  - Gestion des versions
  - Documentation continue

5.6 Mot de Fin

Cette étude constitue une base solide pour la détection de fraude par carte de crédit, tout en ouvrant de nombreuses perspectives d'amélioration et de recherche future. La combinaison d'une approche rigoureuse et d'une compréhension des enjeux métier permet d'envisager des développements prometteurs dans ce domaine crucial de la sécurité financière.