Détection de Fraude par Apprentissage Automatique

Abdoulaye SALL

6 mars 2025

Résumé

Ce rapport présente le développement d'un système de détection de fraude basé sur des techniques d'apprentissage automatique.
L'étude exploite plusieurs sources de données liées aux clients, aux transactions et aux indicateurs de fraude pour construire un modèle prédictif efficace.
Après une exploration approfondie des données et une préparation minutieuse, un modèle d'arbre de décision a été entraîné pour identifier les transactions frauduleuses.
Les résultats montrent la pertinence de cette approche et ses implications pour la détection précoce des activités frauduleuses dans le secteur financier.

Table des matières

1 Introduction

La fraude financière représente un défi majeur pour les institutions financières, engendrant des pertes estimées à plusieurs milliards chaque année.
Face à cette menace persistante, l'utilisation de techniques avancées d'analyse de données et d'apprentissage automatique devient cruciale.

Ce projet vise à développer un système robuste de détection de fraude utilisant des algorithmes d'apprentissage supervisé.
L'approche adoptée repose sur l'intégration de multiples sources de données, comprenant des informations sur les clients, les transactions, les commerçants et des indicateurs de comportements suspects.

Les objectifs principaux de cette étude sont :

— Identifier les facteurs clés associés aux transactions frauduleuses
— Concevoir un modèle prédictif capable de détecter efficacement les fraudes
— Évaluer la performance du modèle et proposer des améliorations potentielles

2 Méthodologie

2.1 Sources de données

Le projet s'appuie sur dix ensembles de données complémentaires, regroupés en plusieurs catégories :

Catégorie	**Ensemble de données**
Profils clients	account_activity.csv, customer_data.csv
Indicateurs de fraude	fraud_indicators.csv, suspicious_activity.csv
Informations commerçants	merchant_data.csv, transaction_category_labels.csv
Montants des transactions	amount_data.csv, anomaly_scores.csv
Données transactionnelles	transaction_metadata.csv, transaction_records.csv

TABLE 1 – Sources de données utilisées dans le projet

2.2 Processus d'analyse

L'analyse a été réalisée selon un processus structuré en plusieurs étapes :

1. **Chargement et exploration initiale des données**
2. **Fusion des ensembles de données** pour créer :
   — Un ensemble de données client
   — Un ensemble de données transactionnel
   — Un jeu de données final intégrant toutes les informations
3. **Analyse exploratoire** comprenant :
   — Étude des distributions des variables catégorielles et numériques
   — Analyse de la variable cible (indicateur de fraude)
   — Identification des corrélations entre variables
4. **Préparation des données** incluant :
   — Suppression des variables non pertinentes
   — Feature engineering (création de nouvelles variables)
   — Encodage des variables catégorielles
5. **Modélisation et évaluation**

3 Exploration et préparation des données

3.1 Fusion des données

La première étape consistait à fusionner les différents ensembles de données pour obtenir une vue complète. Cette intégration a été réalisée en deux phases :

```
# Fusion des bases de donn es pour constituer un dataset client
customer_data = pd.merge(customer, account, on='CustomerID')
customer_data = pd.merge(customer_data, suspision, on='CustomerID')

# Fusion des bases de donn es pour constituer un dataset
    transactionnel
transaction_data1 = pd.merge(fraud, tran_cat, on="TransactionID")
transaction_data2 = pd.merge(amount, anamoly, on="TransactionID")
transaction_data3 = pd.merge(tran_data, tran_rec, on="TransactionID
    ")
transaction_data = pd.merge(transaction_data1, transaction_data2,
    on="TransactionID")
transaction_data = pd.merge(transaction_data, transaction_data3, on
    ="TransactionID")

# Fusion des donn es clients et transactionnelles
final_data = pd.merge(transaction_data, customer_data, on="
    CustomerID")
```

Cette approche a permis de créer un jeu de données riche contenant à la fois les caractéristiques des clients et les détails des transactions, essentiels pour l'identification des schémas de fraude.

3.2 Analyse exploratoire

L'analyse exploratoire a révélé plusieurs aspects importants des données :

- **Variables catégorielles** : La visualisation des distributions a permis d'identifier les catégories prédominantes et potentiellement les valeurs aberrantes.
- **Variables numériques** : L'utilisation de boxplots a mis en évidence la dispersion des variables quantitatives et la présence d'outliers pouvant influencer le modèle.
- **Distribution de la variable cible** : L'analyse a révélé une distribution déséquilibrée entre transactions frauduleuses et légitimes, caractéristique typique des problèmes de détection de fraude.
- **Corrélations** : La matrice de corrélation a permis d'identifier les relations linéaires entre variables numériques, fournissant des indications sur les facteurs potentiellement prédictifs de fraude.

3.3 Feature Engineering

Plusieurs nouvelles variables ont été créées pour améliorer la capacité prédictive du modèle :

```
# Feature engineering : cr ation de nouvelles variables
data_cleaned['Timestamp'] = pd.to_datetime(data_cleaned['Timestamp'])
data_cleaned['Hour'] = data_cleaned['Timestamp'].dt.hour
data_cleaned['LastLogin'] = pd.to_datetime(data_cleaned['LastLogin'])
data_cleaned['GapDays'] = (data_cleaned['Timestamp'] - data_cleaned['LastLogin']).dt.days.abs()
```

Ces nouvelles caractéristiques incluent :

— **Hour** : L'heure de la transaction, permettant d'identifier des patterns temporels de fraude

— **GapDays** : Le nombre de jours entre la dernière connexion et la transaction, un indicateur potentiel de comportement inhabituel

4 Modélisation

4.1 Préparation pour la modélisation

Avant l'entraînement du modèle, plusieurs étapes préparatoires ont été réalisées :

```
# Pr paration des donn es pour le mod le
X = data_cleaned.drop(['FraudIndicator', 'Timestamp', 'LastLogin'],
    axis=1)
Y = data_cleaned['FraudIndicator']

# Encodage des variables cat goriques
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
X['Category'] = label_encoder.fit_transform(X['Category'])

# Division des donn es en ensemble d'entra nement et de test
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size
   =0.2, random_state=42)
```

Les principales étapes comprenaient :
— Séparation des variables explicatives (X) et de la variable cible (Y)
— Encodage des variables catégorielles pour permettre leur utilisation dans le modèle
— Division des données en ensembles d'entraînement (80%) et de test (20%)

4.2 Algorithme et entraînement

Un modèle d'arbre de décision a été choisi pour sa capacité à capturer des relations non linéaires et sa bonne interprétabilité :

```
# Entra nement d'un mod le de classification
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score,
   recall_score, f1_score, confusion_matrix
model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
y_pred = model.predict(X_test)
```

L'arbre de décision présente plusieurs avantages pour la détection de fraude :
— Capacité à gérer différents types de variables
— Robustesse face aux outliers
— Interprétabilité des règles de décision générées

5 Évaluation du modèle

Le modèle a été évalué à l'aide de plusieurs métriques complémentaires :

```
#   valuation   du  mod le
print("Accuracy:", accuracy_score(Y_test, y_pred))
print("Precision:", precision_score(Y_test, y_pred))
print("Recall:", recall_score(Y_test, y_pred))
print("F1␣Score:", f1_score(Y_test, y_pred))
print("Confusion␣Matrix:")
print(confusion_matrix(Y_test, y_pred))
```

5.1 Interprétation des métriques

Dans le contexte de la détection de fraude, les différentes métriques ont une signification particulière :

- **Précision** : Proportion de transactions identifiées comme frauduleuses qui le sont réellement. Une précision élevée minimise les faux positifs, limitant ainsi les interventions inutiles.
- **Rappel (Recall)** : Proportion de transactions frauduleuses effectivement détectées par le modèle. Un rappel élevé est crucial pour minimiser les pertes financières liées aux fraudes non détectées.
- **F1-Score** : Moyenne harmonique de la précision et du rappel, fournissant une mesure équilibrée de la performance du modèle.
- **Matrice de confusion** : Détail des prédictions correctes et incorrectes, permettant une analyse fine des types d'erreurs commises par le modèle.

6 Perspectives

6.1 Forces et limites de l'approche

Le système développé présente plusieurs forces :
— Intégration de multiples sources de données offrant une vue complète du contexte
— Création de variables temporelles pertinentes pour la détection de comportements suspects
— Utilisation d'un modèle interprétable facilitant l'analyse des facteurs de risque
Cependant, certaines limites subsistent :
— Absence de techniques spécifiques pour gérer le déséquilibre des classes
— Utilisation d'un seul algorithme de classification
— Manque de validation croisée pour une évaluation plus robuste

6.2 Améliorations potentielles

Plusieurs pistes d'amélioration pourraient être explorées :
— **Techniques d'échantillonnage** : Implémenter des méthodes comme SMOTE pour rééquilibrer les classes
— **Optimisation des hyperparamètres** : Réaliser une recherche par grille pour optimiser les paramètres du modèle
— **Algorithmes avancés** : Tester des approches comme les forêts aléatoires, le gradient boosting ou les réseaux de neurones
— **Enrichissement des données** : Intégrer des sources externes comme les données géographiques ou les informations sur les appareils utilisés
— **Modélisation séquentielle** : Exploiter la dimension temporelle des transactions pour détecter des séquences suspectes

7 Conclusion

Cette étude démontre la pertinence des techniques d'apprentissage automatique pour la détection de fraude dans le secteur financier. L'approche développée, basée sur l'intégration de multiples sources de données et l'utilisation d'arbres de décision, permet d'identifier efficacement les transactions potentiellement frauduleuses.

Les résultats obtenus constituent une base solide pour le développement d'un système opérationnel de détection de fraude. Ils soulignent également l'importance d'une analyse approfondie des données et d'une préparation minutieuse pour obtenir des modèles performants.

Les perspectives futures incluent l'amélioration continue du modèle, l'intégration de nouvelles sources de données et le développement d'une architecture permettant la détection en temps réel des activités suspectes.

Références

[1] Pedregosa, F. et al. (2011). Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[2] McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56.

[3] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection : A review. *Statistical science*, 17(3), 235-255.

[4] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9), 1263-1284.