Système de Détection de Fraude

 Néobanque Fluzz

Un déséquilibre à résoudre

Le Challenge

284 807 transactions sur 2 jours

492 fraudes seulement (0,17%)

Ratio 1:578 = 1 fraude pour 578 transactions normales

28 variables anonymisées (PCA) + **montant** + **heure**

Contexte

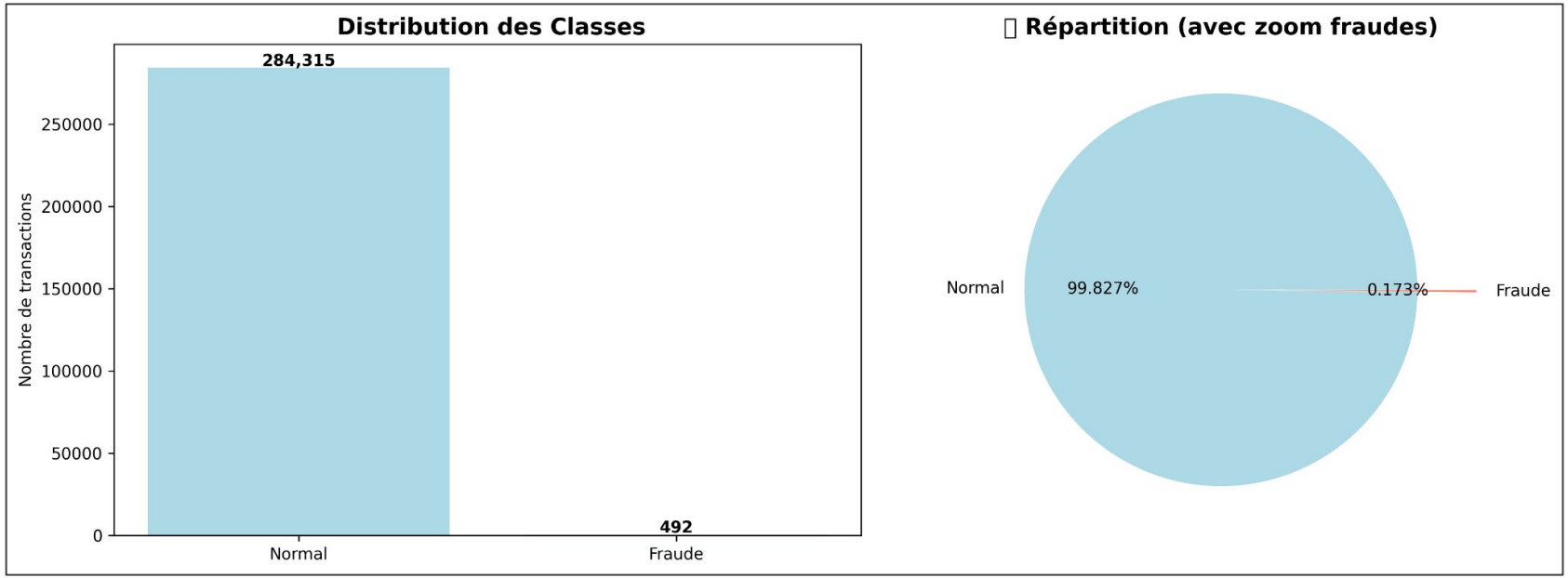
Avec un tel déséquilibre, un algorithme classique prédirait **'pas de fraude'** dans 100% des cas !

Description du problème

Impact

99.8% d'accuracy... mais 0% de fraudes détectées !

Comment entraîner un modèle avec seulement 0,17% de fraudes ?



Les défis à relever

Défi 1

Documentation

Fiche descriptive & cycle de vie

Architecture MLOps

Défi 2

Preprocessing

Feature engineering

Pipeline Airflow

8 variables métier

Défi 3

Éthique

Biais & conformité RGPD

Charte éthique

Les défis à relever

Défi 4

Modélisation

3 algorithmes + optimisation

F1-Score 0.825

Défi 5

Monitoring

Dashboard Grafana temps réel

Alertes automatiques

Défi 6

Sécurité

Détection anomalies + logs

Protection proactive

Les défis à relever

Défi 7

Industrialisation

Docker + Kubernetes

Déploiement scalable

Documentation

Fondation avant innovation

1 - Fiche descriptive des données - Détection de fraude

Vue d'ensemble rapide

- **Dataset** : Credit Card Fraud Detection (Kaggle/ULB)
- **Volume** : 284 807 transactions sur 2 jours
- **Défi principal** : Déséquilibre extrême (0.17% de fraudes)

Caractéristiques clés

- **39 variables** (31 originales + 8 enrichies) :
  - 28 anonymisées (PCA)
  - `Time` (temps en secondes depuis 1ère transaction)
  - `Amount` (montant en €)
  - `Class` (0=Normal, 1=Fraude)
  - **8 nouvelles variables créées** :
    - Temporelles : `Hour` , `Day` , `Is_Night` , `Is_Weekend`
    - Montants : `Amount_log` , `Amount_Category`
    - PCA : `PCA_Magnitude` , `PCA_Extreme_Count`
- **492 fraudes** vs **284 315 normales** → Ratio ≈ **1:578**
- Données collectées en **Europe (septembre 2013)**, anonymisées pour confidentialité
- **Qualité** : 0 valeurs manquantes ✅

Déséquilibre des classes

- **Normales** : 99.83%
- **Fraudes** : 0.17%
- **Impact** : les modèles tendent à prédire "normal" par défaut
- **Solution** : techniques de rééquilibrage (SMOTE, sous-échantillonnage)

Analyses exploratoires

Montants

- Transactions normales → Moyenne ≈ **88€**, médiane ≈ **23€**
- Transactions frauduleuses → Moyenne ≈ **122€**, médiane ≈ **9€**
- **Insight** : Les fraudes concernent souvent de **petits montants (0-50€)** mais quelques cas isolés à très gros montants augmentent la moyenne.

Enrichissement des variables

De 31 à 39 variables enrichies

Variables Temporelles

- Hour, Day, Is_Night (22h-6h), Is_Weekend

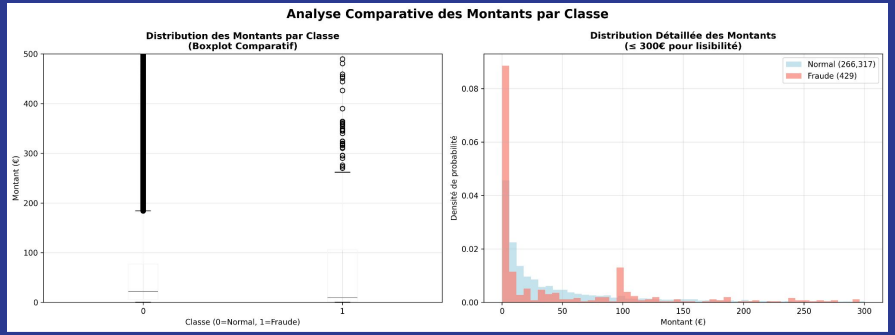
Variables Montants

- Amount_log, Amount_Category (Micro/Small/Medium/Large)

Variables PCA Enrichies

- PCA_Magnitude : Norme euclidienne des 28 composantes

- PCA_Extreme_Count : Variables avec des valeurs > 3σ

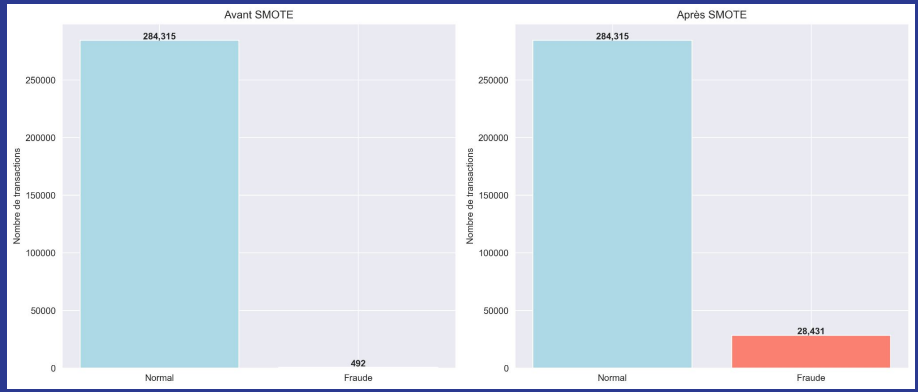


Rééquilibrage

3 méthodes testées

1 retenue : **SMOTE**

Méthode	*Échantillons*	*Temps*	*Score Qualité*	*Statut*
SMOTE	**311 578**	**2s**	0.92	✅ Retenu
SDV	285 726	15s	**0.95**	🔄 Backup
BorderLine SMOTE	**311 578**	3s	0.89	➡️ Alternative

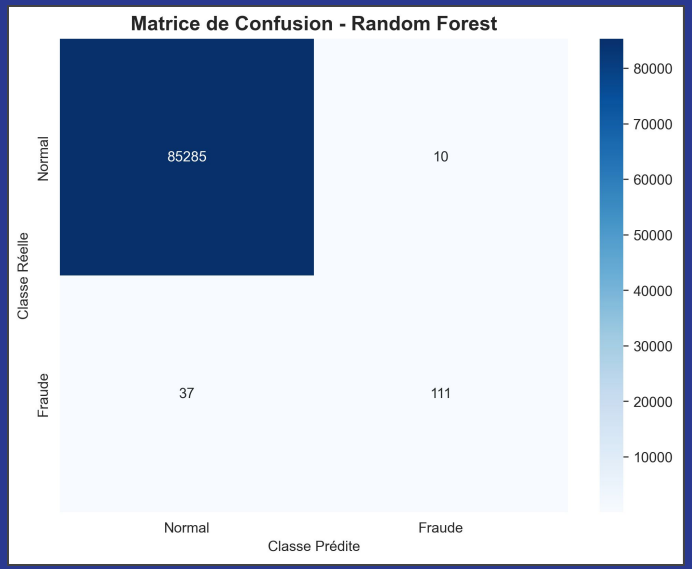


Modélisation

3 algorithmes testés

1 retenu : **Random Forest**

Modèle	*F1-Score*	*Recall*	*Precision*	*AUC-ROC*
Logistic Regression	0.120	**0.912**	0.064	0.971
Random Forest	**0.825**	0.750	**0.917**	0.945
Neural Network	0.784	0.770	0.799	**0.983**



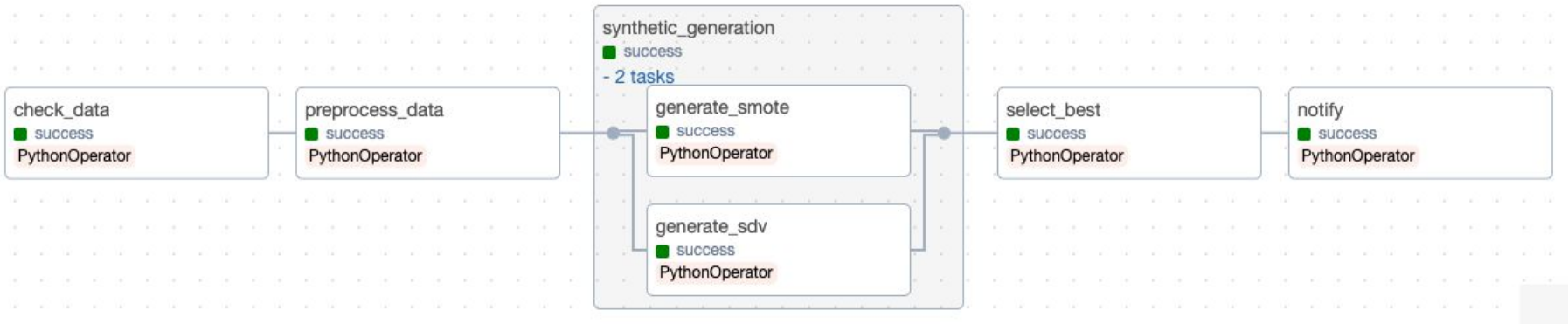
Optimisation

Hyperparamètres

GridSearchCV	*Configuration optimale*	*Amélioration*
3-fold **validation croisée**	**n_estimators**=200 **max_depth**=20 **min_samples_leaf**=2	**+1.2%** vs **paramètres par défaut**

Pipeline automatisé

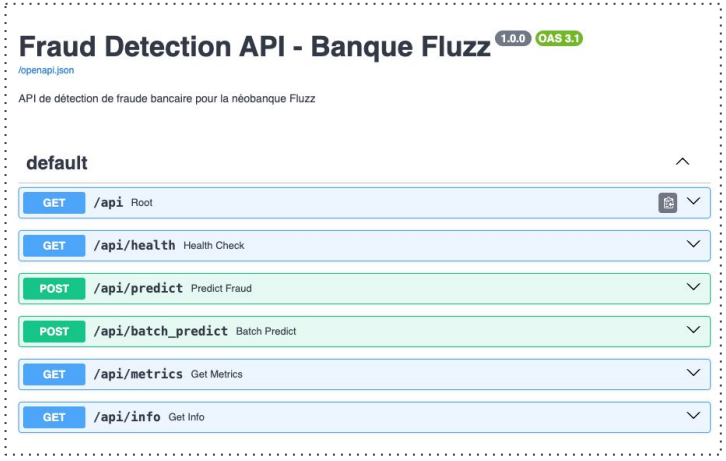
AIRFLOW

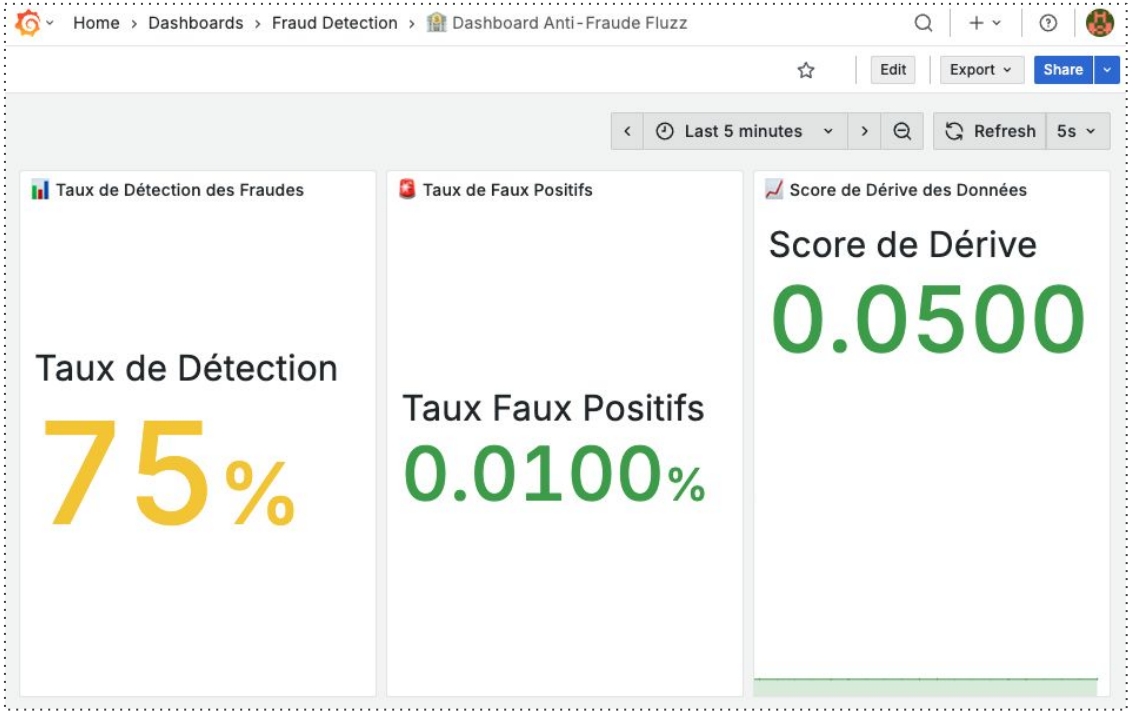


Architecture

Déploiement / Monitoring

FastAPI
Prometheus
Grafana





Conclusion

Performances

- **75% de détection**
- **0.01% faux positifs**
- **Random Forest avec F1-Score 0.825**
- **Pipeline automatisé**