Rapport final : Détection de Fraude par carte bancaire

HOLO Amon-Donovan

Objectif

Construire un modèle de machine learning capable de détecter les transactions frauduleuses dans un dataset bancaire déséquilibré (moins de 0.2% de fraudes).

Étapes du projet

1. Préparation & Exploration

- Chargement du dataset (creditcard.csv).
- Vérification des doublons et des valeurs manquantes → rien d'anormal.
- Variables importantes :
  - Time (temps écoulé en secondes depuis la première transaction).
  - Amount (montant de la transaction).
  - V1–V28 (composantes PCA anonymisées).
  - Class (0 = non fraude, 1 = fraude).
- Distribution fortement déséquilibrée : ~99.8% non fraudes vs 0.2% fraudes.

---

2. Visualisations principales

- Histogrammes et camembert pour comparer fraude (1) vs non fraude (0).
- Analyse de Amount et Time :
  - Montants plus dispersés dans les fraudes.
  - Certaines périodes du temps semblent plus "propices" aux fraudes.

---

3. Baseline

- Régression logistique (avec class_weight="balanced") :
  - Rappel élevé (87%) → le modèle détecte presque toutes les fraudes.
  - Mais précision très faible (~6%) → beaucoup de faux positifs.
  - AUC ≈ 0.97

---

4. Modèles plus avancés

- Random Forest (class_weight + tuning) :
  - Très bon compromis : précision ≈ 88%, rappel ≈ 76%, F1 ≈ 82%.
  - AUC ≈ 0.94
- XGBoost (après tuning) :
  - Rappel un peu plus élevé (79–82%).
  - Précision plus faible que RF.
  - AUC ≈ 0.94

---

5. Rééquilibrage des classes (SMOTE)

- Application de SMOTE pour générer artificiellement des fraudes.
- Amélioration notable de la détection :
  - RF+SMOTE : Rappel ~76–81%, Précision ~88–90%
  - XGB+SMOTE : Rappel ~79–82%, Précision ~70–80%

---

6. Ajustement des seuils

- En baissant le seuil (0.5 → 0.3 ou 0.2), on augmente le recall (fraudes détectées).
- Mais la précision chute → plus de faux positifs.
- Bon compromis trouvé :
  - RF+SMOTE à seuil = 0.4
  - XGB+SMOTE à seuil = 0.3

---

7. Comparaisons globales

- ROC AUC : tous les modèles autour de 0.94–0.97
- Precision-Recall curves → montrent bien le compromis.
- Matrices de confusion (heatmaps annotées) → très parlantes pour expliquer les TN / FP / FN / TP.

Résultats clés

- Baseline (Logistic Regression) : Recall fort mais précision très faible.
- Random Forest : meilleur compromis précision/recall.
- XGBoost : recall plus fort, précision moins bonne.
- SMOTE + seuils ajustés → améliorations significatives.

Conclusion

- Dans un contexte fraude bancaire, le rappel (recall) est prioritaire : il vaut mieux détecter trop de fraudes (faux positifs) que d'en rater (faux négatifs).
- RandomForest+SMOTE @ seuil=0.4 semble être le meilleur compromis :
    - 88% de précision
    - 76% de recall
    - AUC ≈ 0.94
- Améliorations possibles :
    - Tester LightGBM / CatBoost
    - Optimisation via Precision-Recall AUC plutôt que ROC AUC
    - Intégration dans un pipeline métier (alertes en temps réel)