🧾Rapport Technique – Détection de Fraudes par Carte Bancaire

📄**Auteur : Ben Aymar Youssouf**

Étudiant à l'ESIEA – TD38, en mobilité à l'UQAC\ Cours : Machine Learning\ Encadrant : Oussama El Boujjadia

---

1. Introduction

La détection de fraudes bancaires est un enjeu majeur pour les institutions financières. Grâce au machine learning, il est désormais possible de créer des modèles prédictifs capables de distinguer les transactions normales des transactions frauduleuses. Ce projet s'inscrit dans le cadre du cours de Machine Learning à l'UQAC et a pour objectif de mettre en pratique les techniques vues en cours sur un jeu de données réel.

2. Objectif du projet

L'objectif est de construire un modèle d'apprentissage supervisé capable de prédire si une transaction est frauduleuse ou non, en se basant sur un ensemble de variables numériques extraites d'un jeu de données anonymisé.

3. Jeu de données

Le dataset utilisé provient de la plateforme Kaggle : **Credit Card Fraud Detection**. Il comprend :

- 284 807 transactions réalisées par des porteurs de cartes européens.
- Une classe cible `Class` : 0 = normale, 1 = fraude.
- Les variables `V1` à `V28` sont issues d'une **analyse en composantes principales (PCA)** pour préserver l'anonymat.
- Deux variables complémentaires :
- `Time` : nombre de secondes écoulées depuis la première transaction
- `Amount` : montant de la transaction

Le jeu est **très déséquilibré** : seules 492 transactions sont frauduleuses (soit 0.17 %).

4. Préparation des données

- **Vérifications initiales** : pas de valeurs manquantes ni de doublons.
- **Standardisation** des variables `Amount` et `Time` à l'aide de `StandardScaler`.
- Suppression des colonnes d'origine après transformation.
- **Séparation du jeu de données** :
- 70 % pour l'apprentissage
- 30 % pour le test
- Échantillonnage **stratifié** pour conserver la proportion des classes

5. Modèles testés

Deux modèles de classification binaire ont été utilisés :

- **Régression Logistique**
- **Random Forest Classifier**

Ces deux modèles ont été entraînés à l'aide de `scikit-learn` et évalués sur les données de test.

6. Évaluation des performances

Métriques utilisées :

- **Recall (rappel)** : mesure les fraudes détectées correctement
- **Precision** : indique la fiabilité des alertes de fraude
- **F1-score** : moyenne harmonique de la précision et du rappel
- **Matrice de confusion** : pour visualiser les TP, TN, FP, FN
- **Courbe ROC** et **AUC** : pour comparer la performance globale

Résultats comparés :

Modèle	Recall (Fraude)	Précision (Fraude)	F1-score (Fraude)	AUC
Régression Logistique	0.61	0.86	0.72	\~0.91
Random Forest	**0.76** ✅	**0.95** ✅	**0.84** ✅	**\~0.98** ✅

Observations :

- Le modèle Random Forest **dépasse largement** la régression logistique en rappel, F1-score et AUC.
- Il détecte plus de fraudes tout en générant moins de faux positifs.

7. Visualisations incluses

- Histogramme des classes (fraude vs normale) : voir `repartition_classes.png`
- Matrices de confusion comparées : voir `matrices_confusion_comparaison.png`
- Courbe ROC des deux modèles : voir `courbe_roc_comparaison.png`

8. Conclusion

Ce projet a permis de mettre en œuvre toutes les étapes classiques du machine learning supervisé :

- Préparation et exploration de données
- Standardisation
- Modélisation
- Évaluation avec métriques adaptées à un **contexte de classes déséquilibrées**

Le modèle Random Forest offre des performances très satisfaisantes pour détecter les fraudes, avec un **recall de 76 %** et un **AUC de 0.98**.

Ce projet peut servir de base à une intégration future dans un système de scoring bancaire.

9. Fichiers fournis

- `Projet_ccfd_final.ipynb` : notebook complet avec code, graphes et commentaires
- Images générées :
- `repartition_classes.png`
- `matrices_confusion_comparaison.png`
- `courbe_roc_comparaison.png`

10. Remarques

Le notebook est commenté étape par étape, prêt à être présenté ou relu. Le code est compatible Jupyter, Python 3.9+, scikit-learn, pandas, matplotlib, seaborn.

Lien vers le dépôt GitHub (à compléter après création) : [Ben9760/fraude-carte-bancaire-projet: Projet de détection de fraudes par carte bancaire avec scikit-learn]