■ TP Scala & Spark
Analyse exploratoire et détection de fraude sur des transactions bancaires

■ Objectifs pédagogiques

- Charger et manipuler des fichiers CSV et JSON avec Spark en Scala
- Comprendre la structure d'un dataset réel (sale, incomplet, hétérogène)
- Réaliser une analyse exploratoire structurée
- Enrichir des données via des jointures
- Identifier des patterns simples de fraude
- Produire des indicateurs exploitables pour un contexte métier

■ Contexte métier

Vous travaillez pour une équipe **Risk & Fraud** d'un établissement bancaire. Votre mission consiste à analyser des transactions bancaires afin de comprendre le comportement des clients, identifier des signaux faibles de fraude et préparer le terrain pour un futur modèle de Machine Learning.

Vous disposez d'un dossier contenant trois fichiers CSV et deux fichiers JSON issus de transactions réelles anonymisées.

■ Données fournies

- transactions_data.csv : transactions bancaires
- cards_data.csv : informations sur les cartes
- users_data.csv : informations clients
- mcc_codes.json : mapping code MCC → catégorie de commerce
- errors.json : types d'erreurs transactionnelles

■ Contraintes techniques

- Langage : Scala
- Framework : Apache Spark
- API : DataFrame / Dataset
- Interdiction d'utiliser Pandas
- Interdiction d'utiliser du SQL pur

- Utilisation obligatoire des fonctions Spark (groupBy, agg, join, when, count, etc.)

■ PARTIE 1 – Prise en main des données (EDA brute)

1. Chargement des données
Charger les trois fichiers CSV avec schéma inféré et les deux fichiers JSON. Afficher le schéma et les dix premières lignes de chaque DataFrame.

Questions : Combien de colonnes par fichier ? Quels types de données semblent incorrects ou suspects ?

2. Analyse de volumétrie
Calculer le nombre total de transactions, de clients uniques, de cartes uniques et de commerçants uniques.
Interprétation attendue : Qui génère le plus de lignes ?

3. Qualité des données
Identifier les colonnes avec valeurs nulles, les transactions avec montant ≤ 0, les transactions sans MCC et celles contenant des erreurs.
Produire un tableau récapitulatif avec colonne, nombre et pourcentage de valeurs manquantes.

■ PARTIE 2 – Analyse des montants & comportements

4. Analyse des montants
Calculer le montant moyen, médian, minimum et maximum. Étudier la distribution par tranche : < 10 €, 10–50 €, 50–200 €, > 200 €.
Question métier : Les montants élevés sont-ils rares ou fréquents ?

5. Analyse temporelle
Extraire l'heure, le jour et le mois à partir de la date. Calculer le nombre de transactions par heure et par jour de la semaine.
Interprétation : Existe-t-il des heures anormalement actives ?

■ PARTIE 3 – Enrichissement métier (MCC & erreurs)

6. Jointure avec les MCC
Joindre transactions_data avec mcc_codes.json et ajouter une colonne *merchant_category*.
Calculer le top 10 des catégories par volume et le montant moyen par catégorie.
Question : Certaines catégories sont-elles plus risquées ?

7. Analyse des erreurs
À partir de errors.json, identifier les types d'erreurs les plus fréquents, le taux d'erreur par carte et par client.
Indice : Un client avec beaucoup d'erreurs est-il suspect ?

■ PARTIE 4 – Approche fraude (sans Machine Learning)

8. Création d'indicateurs
Créer les indicateurs suivants : nombre de transactions par carte et par jour, montant total par carte et par jour, nombre de villes différentes utilisées par carte, ratio de transactions avec

erreur.

9. Détection de comportements suspects
Identifier les cartes avec plus de X transactions par jour, des transactions dans plus de trois villes ou un montant total journalier élevé. Produire un DataFrame *suspicious_cards*.

■ PARTIE 5 – Restitution

10. Synthèse finale
Chaque étudiant doit fournir un script Scala propre, des commentaires métier et une synthèse écrite répondant aux questions suivantes : quels patterns principaux sont observés ? Quels indicateurs semblent utiles pour un futur modèle ? Quelles limites présentent ces données ?

■ Bonus (optionnel)

Implémenter un score de risque simple

Sauvegarder les résultats en format Parquet

Comparer les clients normaux et les clients suspects