MONNIER Laetitia
DAILLIEZ Patrick-Sébastien
BOUDRIAS Merwan

PROJET CLASSIFICATION

PREDIRE LA FRAUDE DANS LES SERVICES DE PARIEMENT FINANCIERS (CARTE DE CREDIT)

Promotion 4

Mercredi 12 février 2025

SOMMAIRE

I. Introduction

a) Contexte

L'évolution des modes de consommation, notamment avec l'essor des paiements en ligne et des transactions numériques, a profondément transformé le secteur financier. L'utilisation des cartes de crédit est devenue incontournable, facilitant les paiements tout en répondant aux besoins de rapidité et de praticité des utilisateurs. Cette transformation, a toutefois exposé les institutions bancaires et leurs clients à des risques croissants de fraude. La sophistication des méthodes employées par les fraudeurs complique la tâche des mécanismes traditionnels de détection, nécessitant des solutions plus innovantes et performantes.

b) Problématique

Dans un tel contexte, comment concevoir un système capable d'identifier efficacement les transactions frauduleuses parmi des millions de données, souvent déséquilibrées, où les fraudes représentent une infime minorité ? Les enjeux sont multiples : réduire les faux négatifs pour éviter de laisser passer des fraudes réelles, tout en limitant les faux positifs qui pénaliseraient des utilisateurs légitimes. Ce projet vise à répondre à cette problématique en développant un modèle d'intelligence artificielle performant, capable d'allier précision et efficacité, afin de renforcer la sécurité des transactions financières.

II. Planification

a) Organisation générale du projet

La gestion de ce projet s'est appuyée sur l'utilisation d'Asana, un outil collaboratif permettant une organisation efficace des tâches. Le projet a été découpé en tâches et sous-tâches clairement définies, couvrant toutes les étapes, de l'analyse des données à la mise en place de l'interface utilisateur en passant par la préparation des livrables.

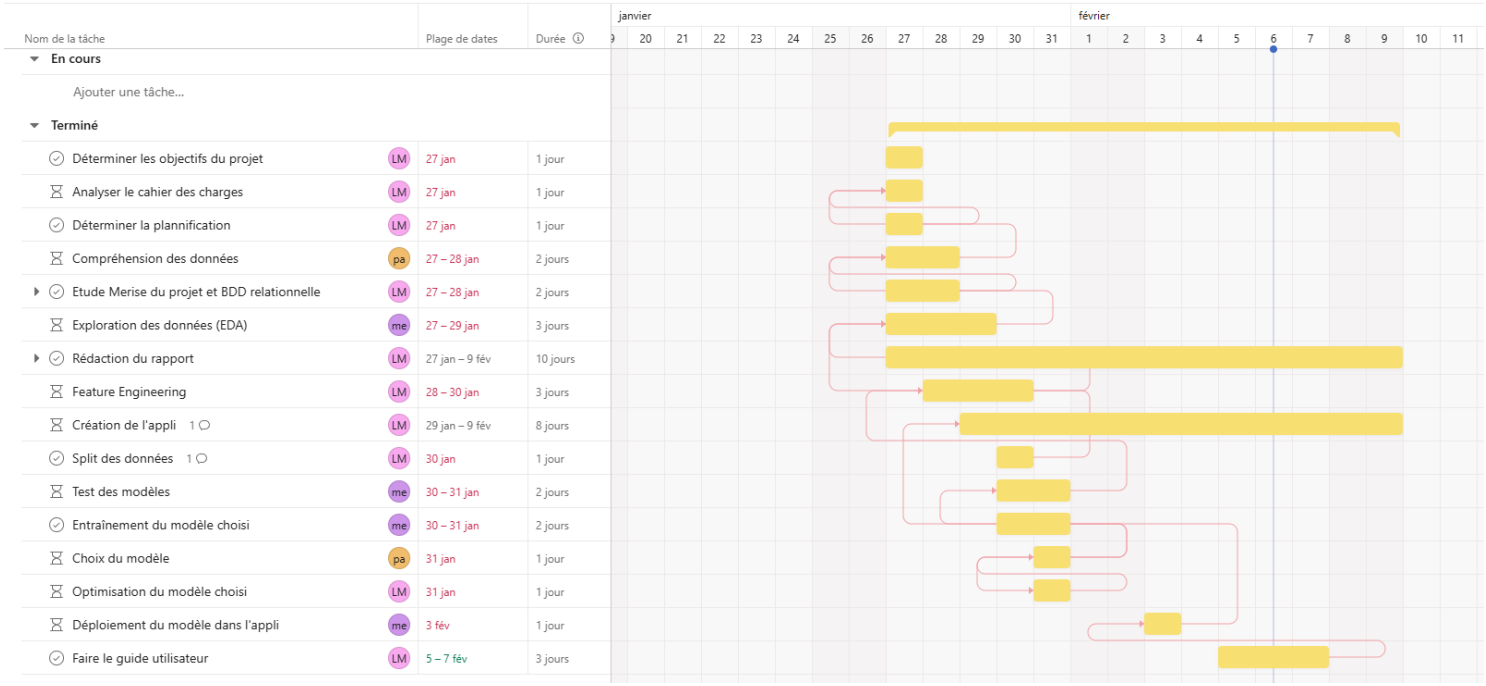
b) Répartition des tâches

La répartition des tâches entre les membres de l'équipe a été réalisée en tenant compte des compétences et des affinités de chacun. Pour assurer un suivi rigoureux, des réunions ont été planifiées quotidiennement :

- Matin : lancement des activités et définition des priorités,
- Milieu de matinée : point d'avancement pour identifier les blocages,
- Début d'après-midi : ajustements nécessaires et résolution des problèmes éventuels.

Cette organisation méthodique a permis une progression fluide et coordonnée vers l'atteinte de l'objectif principal.

c) Planification du projet



III. Les différentes étapes du projet

a) Transformation du Fichier Excel en Fichier CSV

Tout d'abord, le fichier de base en notre possession est un fichier Excel. Nous avons décidé de transformer le fichier Excel en un fichier CSV pour différentes raisons :

- Compatibilité (facilement manipulable avec le langage python)
- Performance (Fichier léger qui se charge assez vite)
- Facilité de traitement des données (via des pipelines)
- Facilité d'opération avec les bases de données

Pour se faire, nous avons utilisé une bibliothèque python (« openpxyl ») qui nous a permis de créer un nouveau fichier en format CSV dans lequel se trouve enregistré toutes les lignes du jeu de données. Ceci nous a permis de garder le fichier de base Excel intact et de travailler sur le fichier CSV tout au long du projet. Voici le code :

```python
#Importation des bibliothèques
import pandas as pd
import openpyxl
import csv
import seaborn as sns
import matplotlib.pyplot as plt
import os

#Définition des chemins d'accès des fichiers Excel et CSV
xlsx_file = "00_ETL_excel.xlsx"  #Chemin du fichier Excel à convertir
csv_file = "ETL_csv.csv"  #Chemin du fichier CSV de sortie

try:
    #Vérification si le fichier CSV existe déjà
    if os.path.exists(csv_file):  #Si le fichier CSV existe déjà, ne rien faire
        print(f"Le fichier {csv_file} existe déjà. Aucun besoin de le recréer.")
    else:
        #Tentative d'ouverture du fichier Excel
        workbook = openpyxl.load_workbook(xlsx_file)  #Chargement du fichier Excel
        sheet = workbook.active  #Accéder à la première feuille active du fichier Excel

        #Créer le fichier CSV si il n'existe pas
        with open(csv_file, mode="w", newline="", encoding="utf-8") as file:
            writer = csv.writer(file)  #Création de l'objet d'écriture CSV

            #Parcourir chaque ligne de la feuille Excel et écrire dans le fichier CSV
            for row in sheet.iter_rows(values_only=True):
                writer.writerow(row)  #Écrire chaque ligne dans le fichier CSV

        print(f"Conversion terminée : {csv_file}")  #Message de succès
```

```
except FileNotFoundError:
    #Si le fichier Excel n'est pas trouvé
    print(f"Erreur : Le fichier {xlsx_file} n'a pas été trouvé.")

except PermissionError:
    #Si le fichier CSV est ouvert ou inaccessible (problèmes de permission)
    print(f"Erreur : Impossible d'écrire dans le fichier {csv_file}. Vérifiez les permissions.")

except Exception as e:
    #Gérer d'autres types d'erreurs
    print(f"Une erreur inattendue s'est produite : {e}")

#Ouverture du fichier CSV pour l'affichage
df = pd.read_csv(csv_file)  #Lecture du fichier CSV
print(df.head())  #Afficher les premières lignes du fichier CSV
```

b) La base de données

Avant de commencer à explorer l'ensemble des données, nous avons décidé de nous lancer sur la création d'une Base de données regroupant la totalité du jeu de données.

Pour se faire, nous avons étudié l'appellation des colonnes ainsi que leur type dans le but d'avoir des renseignements précis et cohérents, facilitant ainsi l'analyse et le traitement des données. Nous les avons renommés pour gagner en clarté.

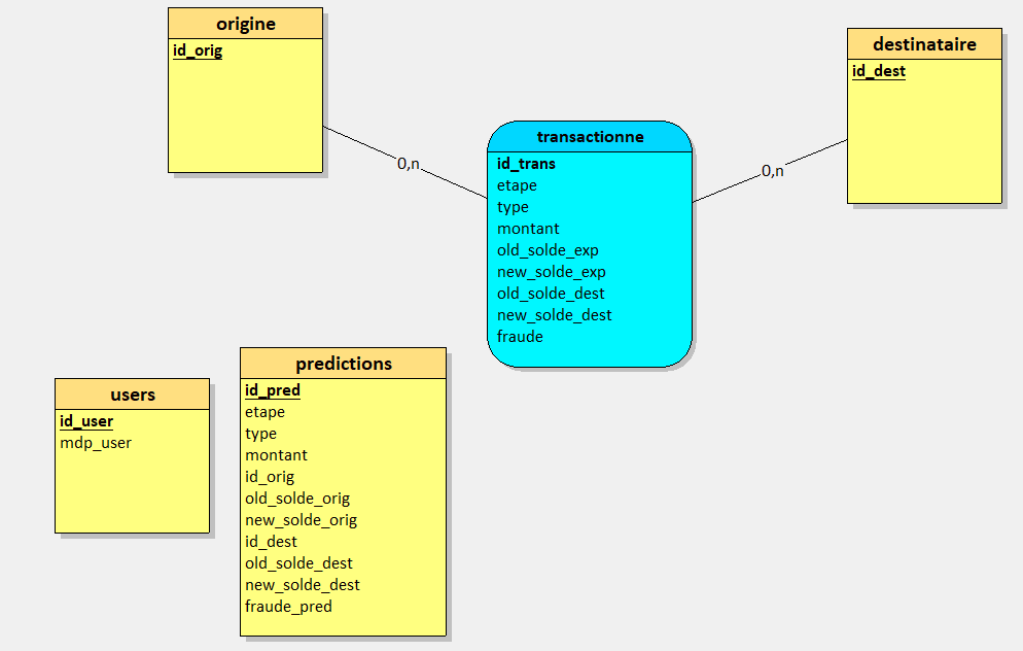
Nous avons donc défini le dictionnaire de données suivant :

Table	Attributs	Type	Contrainte	Description
origine	id_orig	TEXT	PRIMARY KEY	Identifiant unique de l'expéditeur (bancaire).
destinataire	id_dest	TEXT	PRIMARY KEY	Identifiant unique du destinataire (bancaire).
transactionne	id_orig	TEXT	FOREIGN KEY (id_orig) REFERENCES origine(id_orig)	Référence à l'id_orig de la table **origine**.
	id_dest	TEXT	FOREIGN KEY (id_dest) REFERENCES destinataire(id_dest)	Référence à l'id_dest de la table **destinataire**.
	id_trans	INTEGER	PRIMARY KEY	Identifiant unique de la transaction.
	type	TEXT	-	Type de la transaction (ex : PAYMENT, TRANSFER).
	etape	TEXT	-	Etape de la transaction (ex : 1, 2, 3).
	montant	REAL	-	Montant de la transaction.
	old_solde_exp	REAL	-	Ancien solde de l'expéditeur avant la transaction.
	new_solde_exp	REAL	-	Nouveau solde de l'expéditeur après la transaction.
	old_solde_dest	REAL	-	Ancien solde du destinataire avant la transaction.
	new_solde_dest	REAL	-	Nouveau solde du destinataire après la transaction.
	fraude	INTEGER	-	Indicateur de fraude (1 si fraude, 0 sinon).
users	id_user	TEXT	UNIQUE, NOT NULL	Identifiant unique de l'utilisateur.
	password_user	TEXT	NOT NULL	Mot de passe de l'utilisateur.

predictions	id_trans	INTEGER	PRIMARY KEY	Identifiant de la transaction prédite (référence à id_trans dans **transactionne**).
	id_orig	TEXT	FOREIGN KEY (id_orig) REFERENCES origine(id_orig)	Référence à l'id_orig dans la table **origine**.
	id_dest	TEXT	FOREIGN KEY (id_dest) REFERENCES destinataire(id_dest)	Référence à l'id_dest dans la table **destinataire**.
	type	TEXT	-	Type de la transaction prédite (ex : PAYMENT, TRANSFER).
	etape	TEXT	-	Etape de la transaction prédite (ex : 1, 2, 3).
	montant	REAL	-	Montant de la transaction prédite.
	old_solde_exp	REAL	-	Ancien solde de l'expéditeur avant la transaction prédite.
	new_solde_exp	REAL	-	Nouveau solde de l'expéditeur après la transaction prédite.
	old_solde_dest	REAL	-	Ancien solde du destinataire avant la transaction prédite.
	new_solde_dest	REAL	-	Nouveau solde du destinataire après la transaction prédite.
	fraude_pred	INTEGER	-	Indicateur de fraude prédite (1 si fraude, 0 sinon).

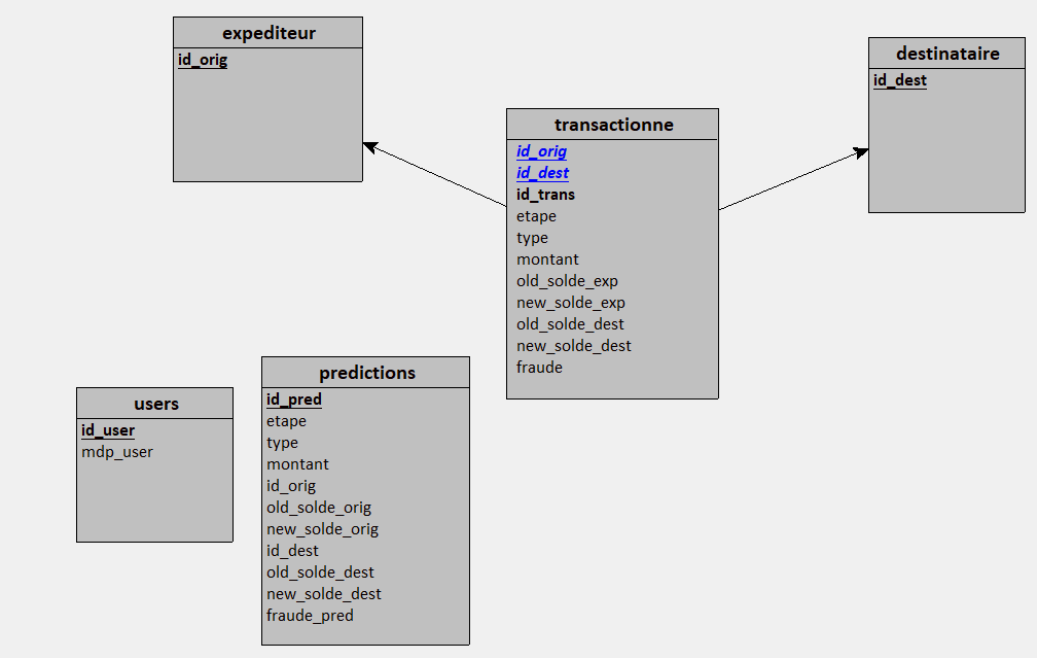
A la suite de cela vient s'ajouter une étude sur l'architecture de la base de données, c'est-à-dire la création des tables et les attributs pour chacune d'entre elles. Pour se faire, nous avons utilisé « Looping » un logiciel principalement employé dans le cadre de la méthode Merise. Ce logiciel permet :

- De créer le MCD (Modèle Conceptuel de Données) :



Nous avons identifié 5 tables (4 tables +1 table d'association), 3 tables servent pour le jeu de données au départ (les tables origine, destinataire et transactionne) et 2 tables pour l'interface d'utilisation et de visualisation du modèle d'intelligence artificiel entraîné (users et predictions).

- De générer automatiquement le MLD (Modèle Logique de Données)



- D'exporter le code SQL :

```
#Création des tables
cursor.execute("""
CREATE TABLE IF NOT EXISTS origine (
    id_orig TEXT PRIMARY KEY
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS destinataire (
    id_dest TEXT PRIMARY KEY
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS transactionne (
    id_orig TEXT,
    id_dest TEXT,
    id_trans INTEGER PRIMARY KEY,
    type TEXT,
    etape TEXT,
    montant REAL,
    old_solde_exp REAL,
    new_solde_exp REAL,
    old_solde_dest REAL,
    new_solde_dest REAL,
    fraude INTEGER,
    FOREIGN KEY(id_orig) REFERENCES origine(id_orig),
    FOREIGN KEY(id_dest) REFERENCES destinataire(id_dest)
)
""")
```

- D'utiliser la nouvelle base de données pour faire des requêtes et commencer l'analyse exploratoire de données. Voici le code :

```
# Connexion à la base de données
conn = sqlite3.connect("database.db")
cursor = conn.cursor()

# Récupérer les données d'une table (ex: 'data_table')
query = "SELECT * FROM transactionne"  # Remplace 'data_table' par le nom de ta table
df = pd.read_sql_query(query, conn)

# Fermer la connexion
conn.close()

# Afficher les premières lignes
print(df.head())
print(df.columns)
```

c) Analyse Exploratoire des Données (EDA)

Le processus d'Analyse Exploratoire de Données est crucial pour comprendre les relations entre les « features » et la « target ». Pour se faire, nous avons utilisé différentes bibliothèques python mis à notre disposition tels que les bibliothèques « pandas » pour le pré-traitement des données mais aussi « seaborn » et « matplotlib » pour la visualisation des données.

Voici la manière utilisée pour faire notre EDA :

- Description des types de données par colonne
- Analyse des statistiques descriptives (moyenne, médiane, écart-type, etc.).
- Exploration des valeurs manquantes et de leur traitement et des doublons potentiels
- Détection des anomalies et des valeurs aberrantes.
- Visualisation des distributions des variables (histogrammes, boxplots).
- Analyse des relations entre variables (matrices de corrélation, scatter plots).

De l'EDA, l'analyse est la suivante :

- Sur le dataset :
    - Classes déséquilibrées (environs 8000 fraudes VS 1 000 000 normales).
    - Pas de valeurs manquantes.
    - Pas de doublons numériques mais doublons dans les id de compte.
    - Pas de valeurs aberrantes.
    - Des écarts importants pour certaines colonnes (les valeurs numériques telles que les montants).

- Les corrélations entre les colonnes
    - Potentielle création de deux colonnes différences de montant pour les comptes expéditeur et destinataire.
    - Pas de corrélation « évidente » entre les autres features.

- Les fraudes :
    - Sont réalisées sur des gros montants.
    - Sont réalisées à des tranches horaires répétées.
    - Sont réalisées uniquement sur les types de transactions suivantes : virement et cash_out.

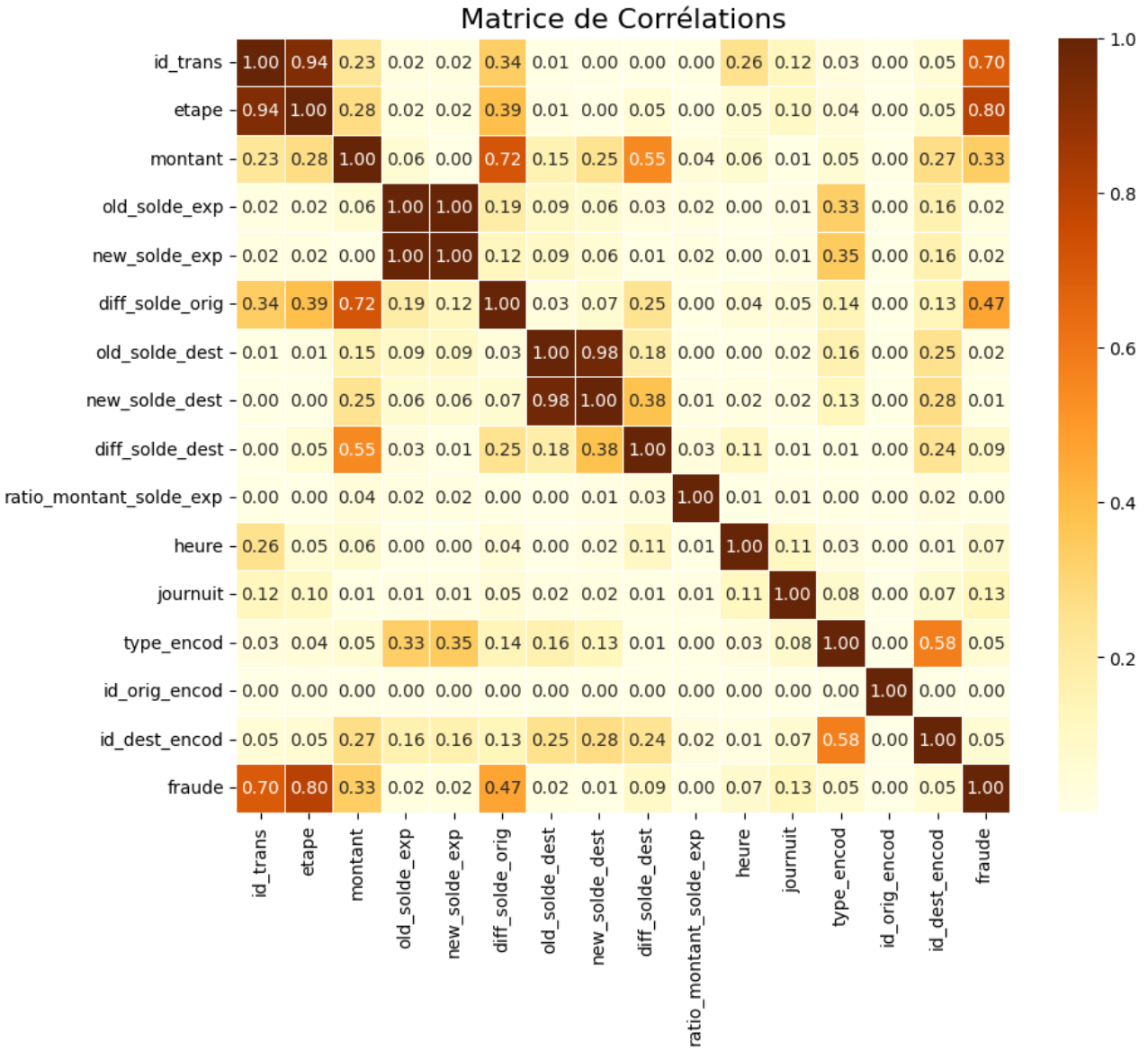
d) Features Engineering

L'idée maintenant est de sélectionner les colonnes qui nous semblent utiles pour la prédiction d'une fraude dans cette banque. Dans le DataSet nous avons ces colonnes : ['id_trans', 'etape', 'type', 'montant', 'id_orig', 'old_solde_exp', 'new_solde_exp', 'id_dest', 'old_solde_dest', 'new_solde_dest', 'fraude'].

Nous décidons d'ajouter d'autres colonnes afin de s'assurer qu'il n'y a pas de meilleures corrélations entre les données. Il nous semble judicieux d'ajouter :

- Une colonne qui calcule la différence entre le nouveau solde et l'ancien de l'expéditeur (origine),
- Une colonne qui calcule la différence entre le nouveau solde et l'ancien du destinataire,
- Une colonne qui périodise les étapes (step) en 24h,
- Une colonne qui divise les étapes (step) en jour ou nuit,
- Une colonne qui fait le ratio du montant sur l'ancien solde de l'expéditeur (origine).

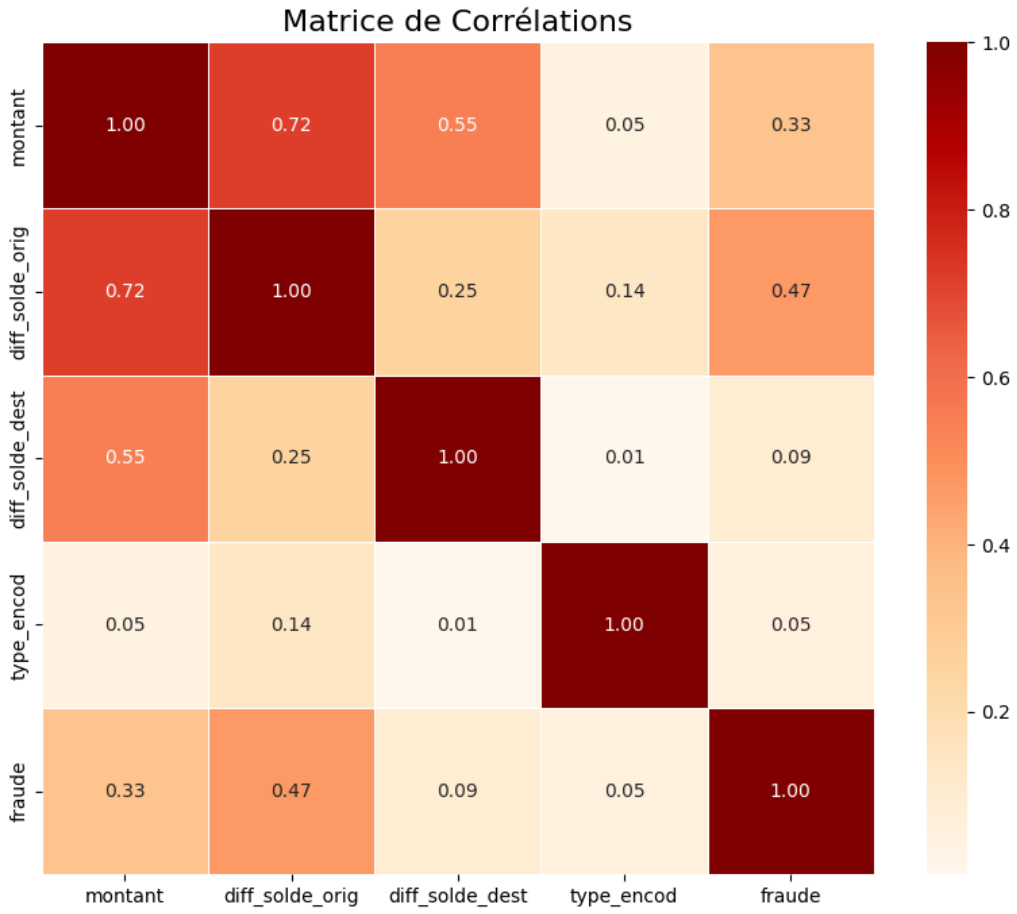
Suite à ces créations de features, nous affichons la matrice de corrélation de toutes les données, tout en encodant les données catégorielles pour qu'elles apparaissent également :



Cette matrice nous fait conclure qu'il y a de trop grosses corrélations au niveau des old et new soldes, c'est pourquoi nous décidons uniquement de garder nos nouvelles colonnes de différence des soldes. Nous concluons aussi que les colonnes id_trans, étape, ratio, heure, journuit, id_orig et id_dest ne sont pas assez pertinentes pour continuer avec le modèle. Les colonnes que nous gardons sont dont :

- Montant
- Diff_solde_orig
- Diff_solde_dest
- Type

Voici la matrice de corrélation à jour :



Il ne faudra pas oublier d'ajouter ses 2 nouvelles colonnes pour l'entrainement du modèle et normaliser les données catégorielles de la colonne "type".

e) Sélection du modèle

- **Technique d'échantillonnage**

Avant de sélectionner le modèle, il est préférable de gérer le déséquilibre des classes. Pour se faire, nous avons le choix entre "ajouter " ou "enlever des transactions frauduleuses" par le biais des techniques d'échantillonnages résumé dans le tableau ci-dessous :

Technique	Avantages	Inconvénients	Quand l'utiliser
SMOTE	Augmente la classe minoritaire	Peut introduire du bruit	Lorsque la classe minoritaire est sous-représentée.
SMOTEN	Génère des échantillons + supprime les erreurs (Tomek Links)	Peut supprimer trop de données	Quand il y a des paires de points mal classées.
SMOTEENN	Augmente la classe minoritaire + nettoie les erreurs (ENN)	Peut perdre des informations lors du nettoyage	Quand vous avez des points mal classés et des données déséquilibrées.
SMOTETomek	Augmente la classe minoritaire + nettoie les erreurs (Tomek Links)	Peut supprimer trop de points	Quand il y a des paires mal classées (Tomek Links).
NearMiss	Réduit la classe majoritaire pour équilibrer les classes	Peut supprimer des exemples utiles	Quand vous préférez réduire la classe majoritaire pour équilibrer les classes.
Poids de classes	Ne modifie pas les données, simple à appliquer.	Peut ne pas suffire si le déséquilibre est trop important.	Quand vous ne voulez pas modifier les données, mais ajuster l'importance de chaque classe pendant l'entraînement.
Random Undersampling	Simple, réduit le nombre de données majoritaires.	Peut perdre des informations utiles.	Quand vous avez un trop grand nombre de données majoritaires et voulez réduire le dataset.

Ce qui ressort de ce tableau comparatif est le constat suivant :

- **Si on a beaucoup de données** et qu'on souhaite éviter la perte d'informations, **SMOTE** ou **SMOTENN** sont généralement préférables.
- **Si on a moins de données** ou que l'échantillonnage prend trop de temps, le fait d'ajuster les **poids de classes** pourrait être plus efficace.

Nous avons testé différentes techniques et on a évalué leur impact sur les performances.

- **Modèle**

En même temps que les techniques d'échantillonnage, nous avons testé les modèles de classification ainsi que différents algorithmes, ce qui nous a permis de réduire le champ de possibles et d'enlever des modèles qui n'ont pas donné de bons résultats.

Nous avons réalisé un tableau comparatif des modèles de classification, leurs algorithmes ainsi que leur avantages et inconvénients :

Modèle	Algorithmes	Avantages	Inconvénients	Quand l'utiliser
Régression Logistique	Logistic Regression	- Facile à comprendre et interpréter - Rapide à entraîner - Efficace pour des données linéaires	- Performances limitées pour des relations non linéaires - Sensible aux outliers	Lorsque les relations entre les variables sont linéaires et que l'interprétabilité est importante
Arbres de Décision	Decision Tree	- Interprétable - Peut gérer des relations non linéaires - Robuste aux outliers	- Sujet au sur-apprentissage (overfitting) - Sensible aux petites variations des données	Lorsque l'interprétabilité est nécessaire et pour des données non linéaires simples
Forêts Aléatoires	Random Forest	- Bonne gestion des relations non linéaires - Robuste aux outliers - Prend en compte les interactions entre les variables	- Peut être plus lent à entraîner - Moins interprétable que les arbres de décision	Lorsque l'on a besoin d'un modèle robuste avec une grande capacité de généralisation
Gradient Boosting	Gradient Boosting Machines (GBM), XGBoost, LightGBM	- Haute précision - Excellente gestion des relations complexes et non linéaires - Très performant pour des données déséquilibrées	- Plus lent à entraîner - Moins interprétable - Sujet à l'overfitting sans régularisation	Lorsque la performance est cruciale et qu'il y a des relations complexes dans les données
Support Vector Machines (SVM)	Linear SVM, SVM avec noyaux (RBF, poly, etc.)	-Efficace dans des espaces de grande dimension - Fonctionne bien avec des classes séparables	- Très coûteux en calcul pour de grandes bases de données - Sensible aux paramètres du	Pour des problèmes où les classes sont bien séparables et lorsque les données sont de haute dimension

			noyau et à la normalisation	
K-Nearest Neighbors (KNN)	- KNN	- Simple à comprendre et à implémenter - Efficace avec des données peu bruitées	- Très lent à prédire (car il compare avec tout l'ensemble d'entraînement) - Sensible à la dimensionnalité	Lorsque les données sont peu bruitées et la rapidité des prédictions n'est pas cruciale
Naive Bayes	- Gaussian Naive Bayes, Multinomial Naive Bayes	- Rapide à entraîner - Efficace pour des jeux de données avec peu de caractéristiques - Pas besoin de standardisation	- Hypothèse d'indépendance souvent irréaliste - Moins performant avec des relations complexes	Lorsque les relations entre les variables sont relativement indépendantes et que la rapidité est essentielle
Régression Arbre (XGBoost)	- XGBoost, LightGBM, CatBoost	- Très performant - Gère bien les données déséquilibrées - Bon compromis entre vitesse et précision	- Plus lent à entraîner - Moins interprétable - Paramétrage complexe	Lorsqu'on recherche une solution robuste avec de grandes quantités de données et des relations complexes

En résumé :

Nous avons testé les modèles suivants avec les techniques d'équilibrage des données suivantes :

Modèles testés :

- Régression Logistique : LogisticRegression(),
- Naïve Bayes : GaussianNB(),
- Forêt Aléatoire: RandomForestClassifier(),
- KNN: KNeighborsClassifier(),
- XGBClassifier : XGBClassifier(),
- LGBMClassifier(),
- GradientBoostingClassifier()

Techniques d'échantillonage :

- Aucun
- SMOTE : SMOTE(),
- NearMiss : NearMiss(),
- SMOTEN : SMOTEN(),
- SMOTE+TomekLinks : SMOTETomek(),
- SMOTE+EditedNN : SMOTEENN()

Les modèles ont été testés avec leur hyperparamètres de base afin d'avoir une première estimation des résultats. Nous avons choisi de modifier les hyperparamètres par la suite pour encore plus optimiser les prédictions.

Nous avons imprimé les matrices de corrélations ainsi que l'accuracy et les rapports pour chaque test. Tous nos tests sont dans l'annexe 1 : Résultats des tests des modèles.

Voici ce qui en ressort :

- L'utilisation de la technique d'équilibrage NearMiss n'est pas bonne, nous ne l'utiliserons pas pour ce jeu de données,
- Le modèle Naïve Bayes n'est pas du tout adapté pour ce projet,
- Les modèles RandomForestClassifier() et KNeighborsClassifier() offrent de bons résultats au niveau du recall pour la détection de fraudes, avec une précision qui peut être surement améliorée avec des hyperparamètres, c'est pourquoi nous décidons d'approfondir ces 2 modèles.
-

f) Entrainement des modèles

Afin d'entrainer plus efficacement le modèle, nous utilisons GridSearchCV pour obtenir les meilleurs hyperparamètres adaptés aux données. Cela consiste à tester systématiquement toutes les combinaisons possibles d'un ensemble de valeurs prédéfinies pour les hyperparamètres d'un modèle.

Dans un premier temps nous basons nos recherches sur le modèle Random Forest. Voici les hyperparamètres qui sont ressortis de cette recherche :

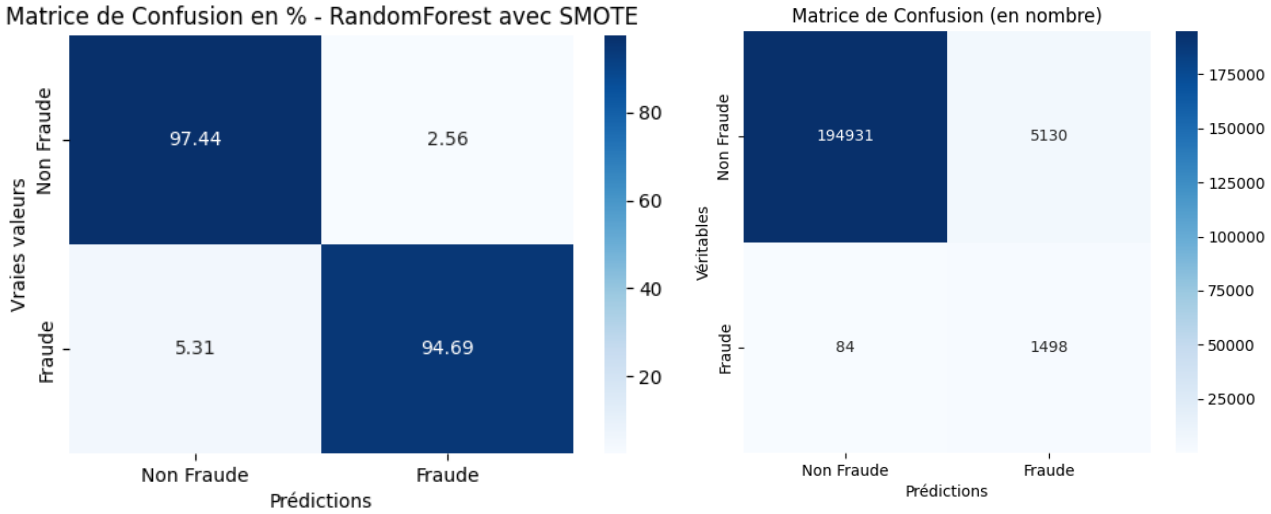
- random_state=42 : Assure la reproductibilité des résultats en fixant la graine aléatoire.
- n_jobs=-1 : Utilise tous les cœurs du processeur pour accélérer l'entraînement.
- n_estimators=200 : Nombre d'arbres dans la forêt (plus = meilleur, mais plus lent).
- max_depth=None : Pas de limite à la profondeur des arbres (risque de surapprentissage).
- min_samples_leaf=2 : Minimum de 2 échantillons par feuille pour éviter des feuilles trop petites.
- min_samples_split=2 : Minimum de 2 échantillons pour diviser un nœud.
- max_features='log2' : Nombre de variables prises en compte à chaque division = log2(n_features).

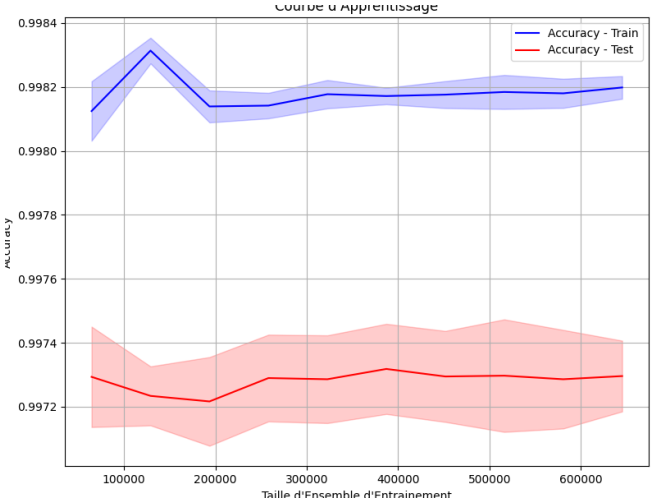
g) Evaluation du modèle

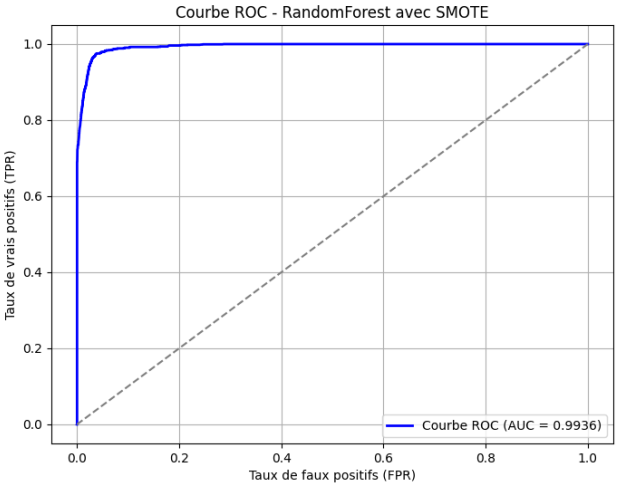
Voici la matrice de confusion ainsi que le rapport suite à l'entrainement du modèle Random Forest avec les hyperparamètres cités ci-dessus :

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	200061
1	0.23	0.95	0.36	1582
accuracy			0.97	201643
macro avg	0.61	0.96	0.68	201643
weighted avg	0.99	0.97	0.98	201643



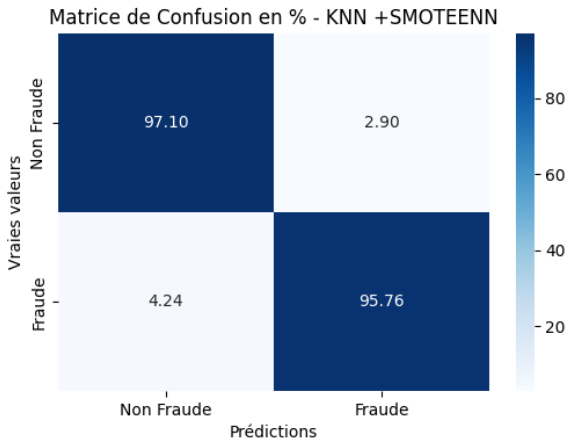


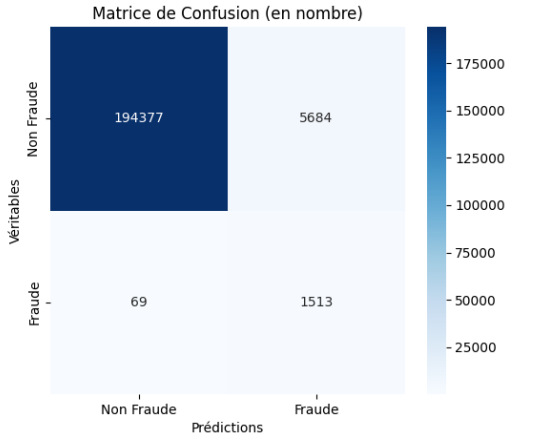


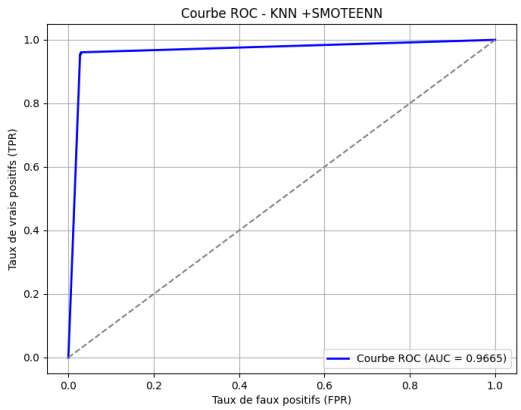
Suite à l'essai du modèle Random Forest dont les résultats sont déjà corrects, nous avons cherché à améliorer nos résultats, notamment pour optimiser la détection des fraudes et éviter qu'elles ne passent inaperçues. Pour cela, nous avons repris nos tests de base sur l'ensemble des modèles et analysé leurs performances afin d'identifier une alternative plus efficace.

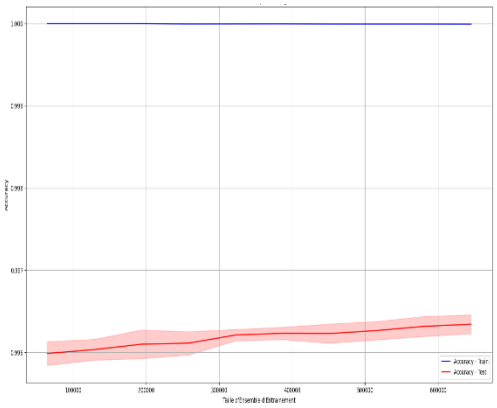
Après cette analyse, nous avons décidé d'utiliser le modèle k-Nearest Neighbors (k-NN) en le combinant avec la technique d'équilibrage SMOTEENN. Cette méthode associe SMOTE qui génère des exemples synthétiques pour la classe minoritaire, et ENN (Edited Nearest Neighbors), qui supprime les points ambigus afin d'améliorer la qualité des données.

Nous avons ainsi procédé à un nouvel entraînement du modèle, qui a montré de meilleures performances en matière de détection des fraudes. Ci-dessous, la matrice de confusion et le rapport de classification obtenus :









```
🔍 Démarrage de GridSearchCV pour KNeighborsClassifier avec SMOTEENN...
✅ GridSearchCV terminé en 976.53 secondes.
✅ Meilleurs hyperparamètres : {'metric': 'euclidean', 'n_neighbors': 3, 'weights': 'distance'}

🔄 Prédictions sur l'ensemble de test...
✅ Accuracy: 0.9709

📊 Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.99    200061
           1       0.21      0.96      0.34      1582

    accuracy                           0.97    201643
   macro avg       0.60      0.96      0.66    201643
weighted avg       0.99      0.97      0.98    201643
```

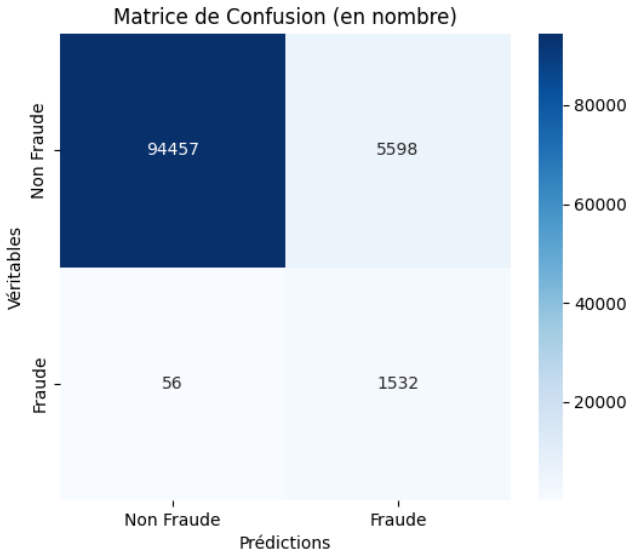
On remarque que les modèles RandomForest avec SMOTE et KNN avec SMOTEENN donne de bons résultats en terme d'accuracy et de recall mais la courbe d'entraînement stagne à 1 quand le nombre de données augmentent. La courbe de test converge vers cette courbe ce qui signifie un risque d'overfitting.

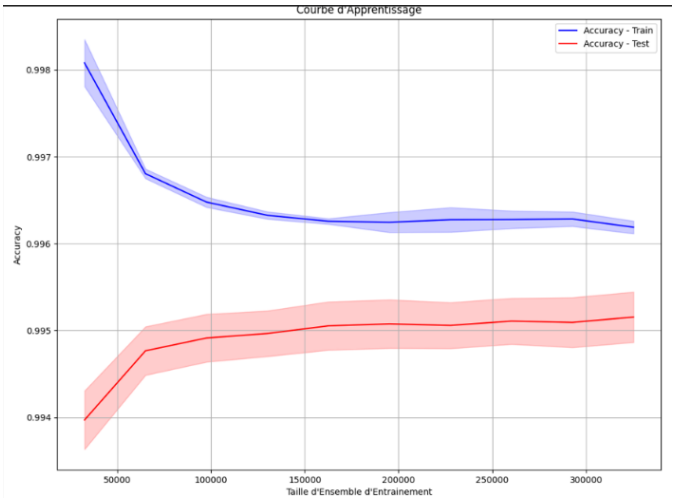
Afin de contrer cela, nous avons opté pour une autre approche notamment dans l'approche des données. En effet, nous avons réduit le dataset de 500 000 données qui ne sont pas des transactions frauduleuses et de type CASH_IN, DEBIT, PAYMENT de manière aléatoire, puis nous avons de nouveau entraîner le modèle de RandomForest. Voici le code :

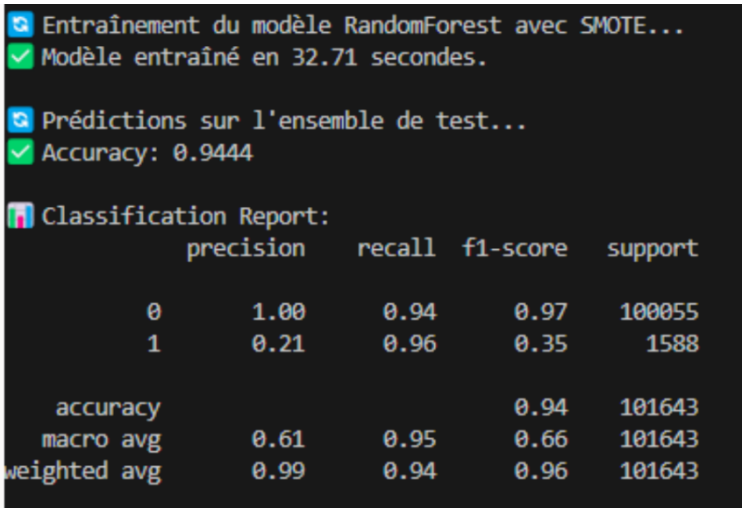
On obtient le résultat suivant avec RandomForest avec les hyperparamètres suivants :

```python
# Filtrer les types qu'on souhaite retirer
types_to_remove = ['CASH_IN', 'DEBIT', 'PAYMENT']

# Créer un DataFrame avec uniquement les lignes à retirer
df_to_remove = df[df['type'].isin(types_to_remove)]

# Vérifiez combien de lignes peuvent être retirées
print(f"Taille du DataFrame à retirer : {df_to_remove.shape[0]}")

# Si le DataFrame à retirer a plus de 500 000 lignes, échantillonnez 500 000 lignes aléatoirement
if df_to_remove.shape[0] > 500000:
    df_to_remove = df_to_remove.sample(n=500000, random_state=1)  # random_state pour la reproductibilité

# Retirer les lignes sélectionnées du DataFrame original
df_final = df.drop(df_to_remove.index)

# Vérifiez la taille finale du dataset
print(f"Taille finale du dataset : {df_final.shape[0]}")

# Sauvegarder le nouveau dataset
df_final.to_csv('dataset_modifie.csv', index=False)  # Remplacez par le chemin de votre fichier de sortie
```

```python
# 📌 Définition du modèle et des hyperparamètres
model = RandomForestClassifier(
    random_state=42,
    max_depth=20,
    max_features='log2',
    min_samples_leaf=1,
    min_samples_split=2,
    n_estimators=100
)
```





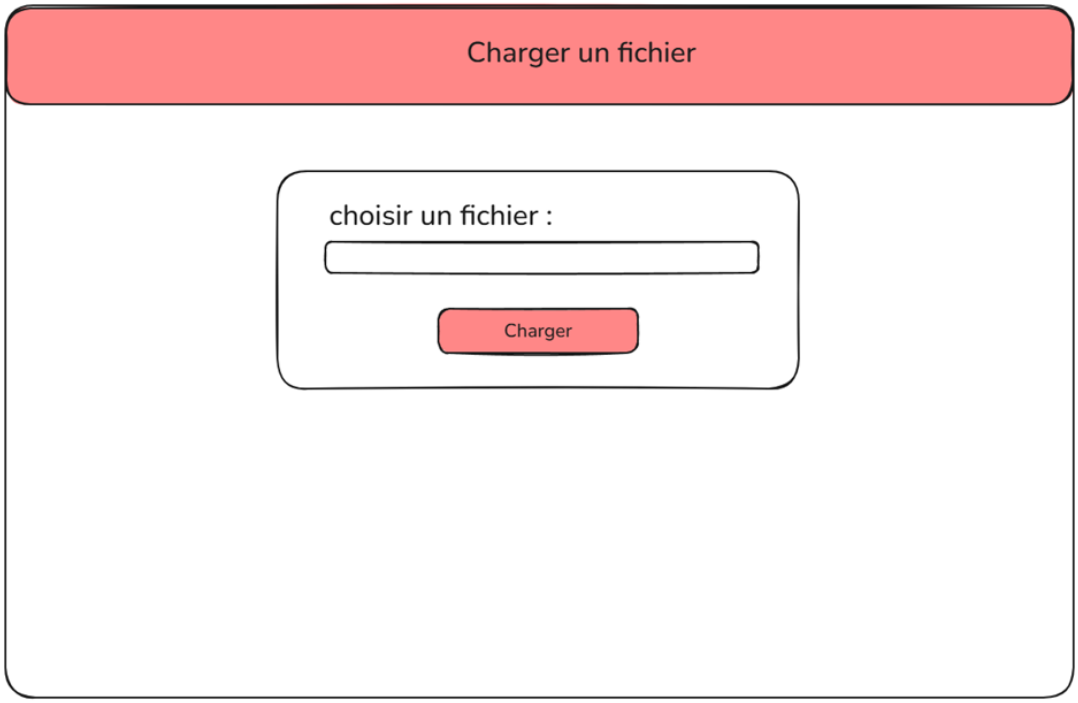


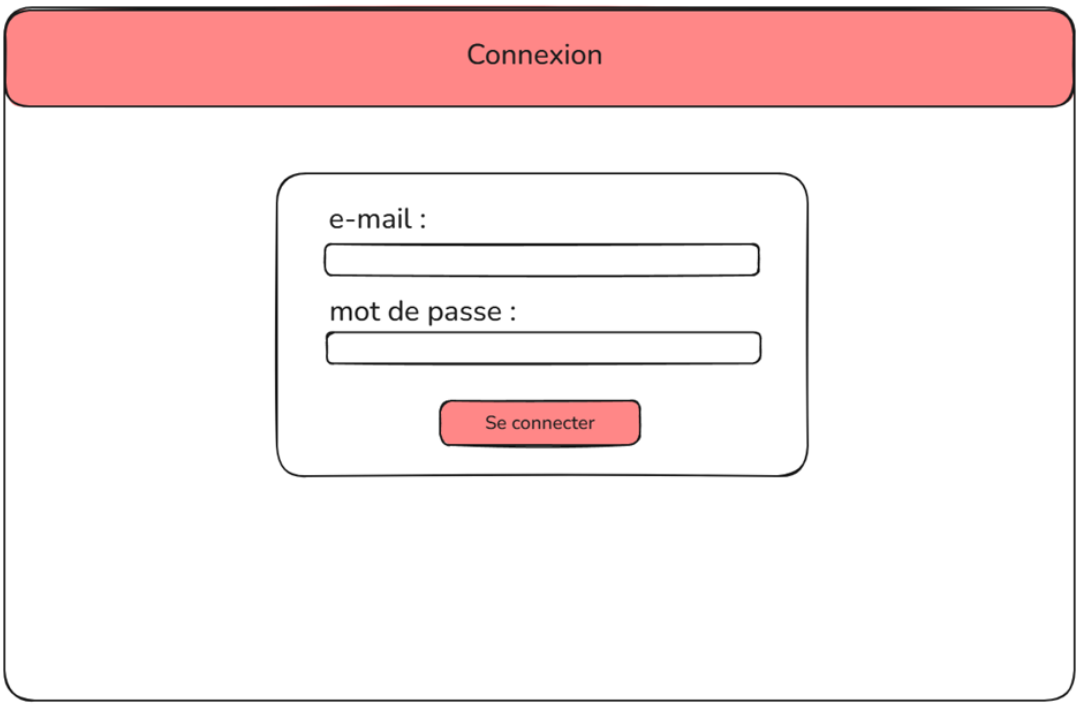
On remarque cette fois-ci que les courbes d'apprentissages convergent avec une courbe d'entraînement qui diminue progressivement et faiblement au fil des données ce qui présage d'un modèle cohérent et correct pour notre problématique de fraude. A ce stade, il faudrait maintenant « jouer » avec les paramètres en internes de chaque élément qui influence le modèle (technique d'échantillonnage, explorer le nouveau dataset (le random_state notamment…). Nous avons décidé de retenir ce modèle pour l'intégration dans l'API.

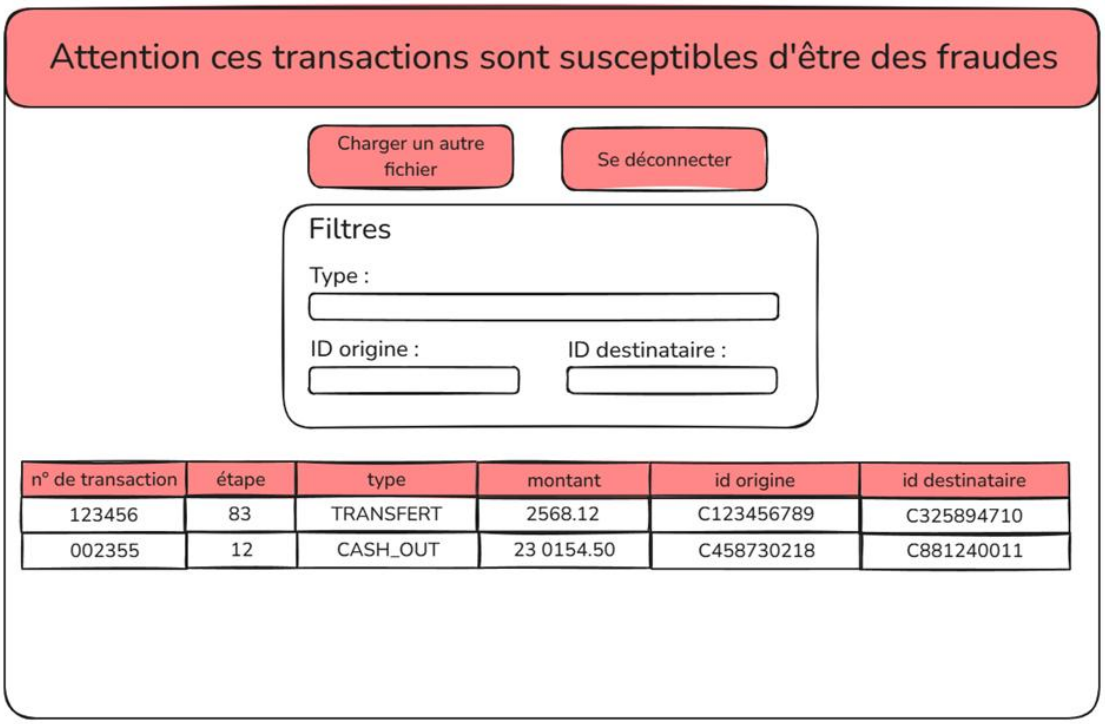
h) Maquette de l'interface

L'idée principale de l'interface et de pouvoir rentrer un fichier en format csv contenant les mêmes lignes que le tableau fourni au début du projet, à savoir qu'une ligne représente une nouvelle transaction. L'interface étant prévue pour une banque, il y est demandé dès le début de s'identifier pour pouvoir accéder à ce service de prédiction de fraudes.
Suite à l'envoi du fichier, l'interface permet de visualiser directement les nouvelles transactions qui semblent être des fraudes à la suite de la prédiction faite par le modèle IA. Des filtres sont ajoutés pour permet de mieux cibler certaines fraudes (filtre sur le type et recherche sur les identifiants)

Voici les maquettes de l'interface utilisateur :







n° de transaction	étape	type	montant	id origine	id destinataire
123456	83	TRANSFERT	2568.12	C123456789	C325894710
002355	12	CASH_OUT	23 0154.50	C458730218	C881240011

IV. Guide utilisateur

Présentation de l'Application

L'application a été développée avec le framework Flask et repose sur une architecture simple avec trois pages HTML principales :

- index.html (route /main ou /accueil)
- load.html (route /load)
- predict.html (route /predict)

Une feuille de style CSS nommée style.css est utilisée pour la mise en forme. La navigation entre les différentes pages est assurée par des boutons.

Authentification des Utilisateurs

La route /main (ou /accueil) est responsable de l'authentification des utilisateurs. Lorsqu'un utilisateur saisit son identifiant (email) et son mot de passe, le système vérifie leur présence dans la table "users" de la base de données.

En cas de correspondance, l'utilisateur est redirigé vers la page suivante.

Sinon, un message d'erreur est affiché à l'aide du système de messagerie flash de Flask.

Chargement du Fichier CSV

La route /load permet de télécharger un fichier de données au format CSV.

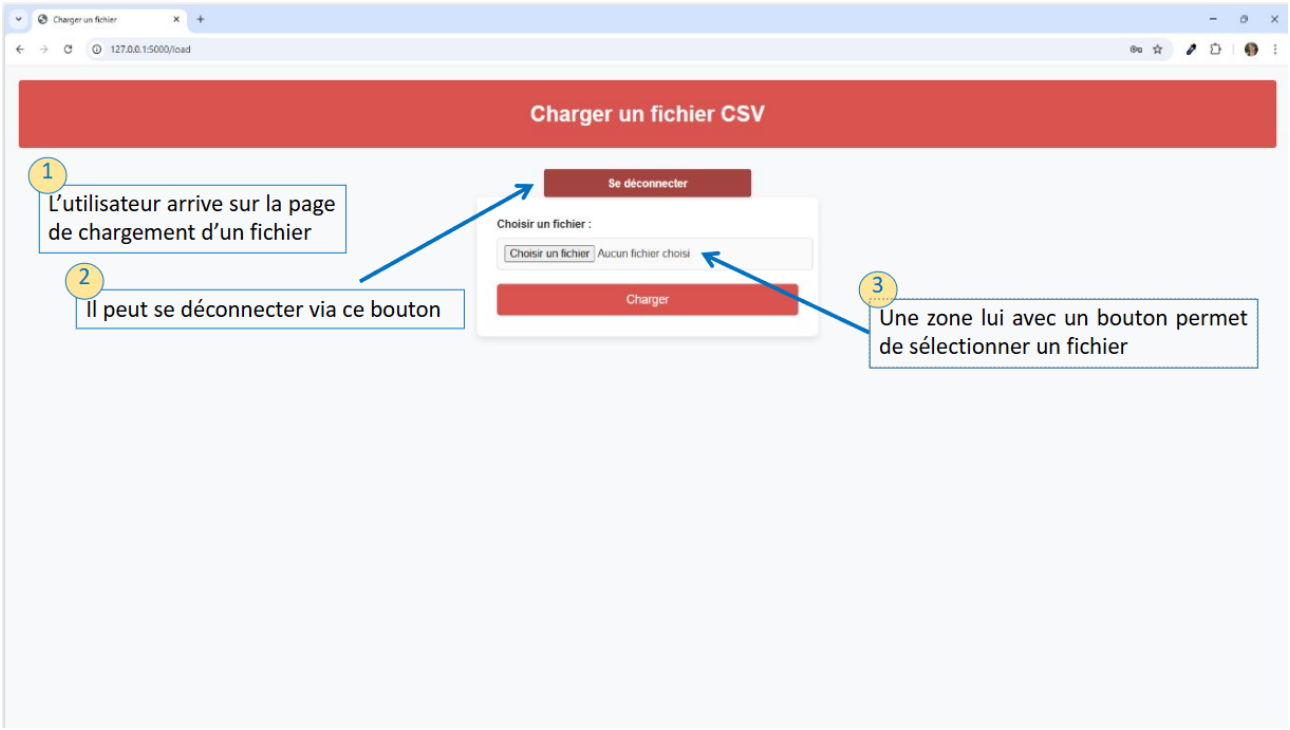
Une vérification est effectuée pour s'assurer qu'un fichier a bien été envoyé et qu'il est du bon format (CSV).

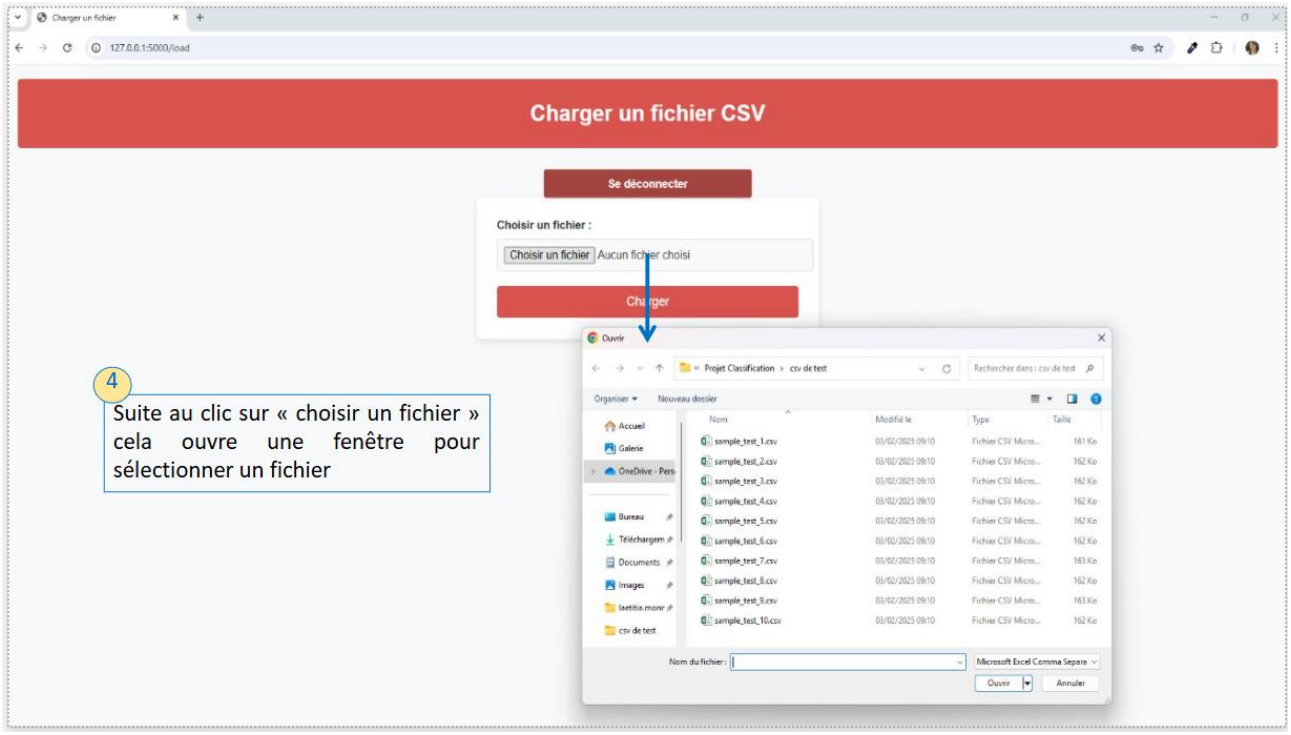
Traitement et Prédiction

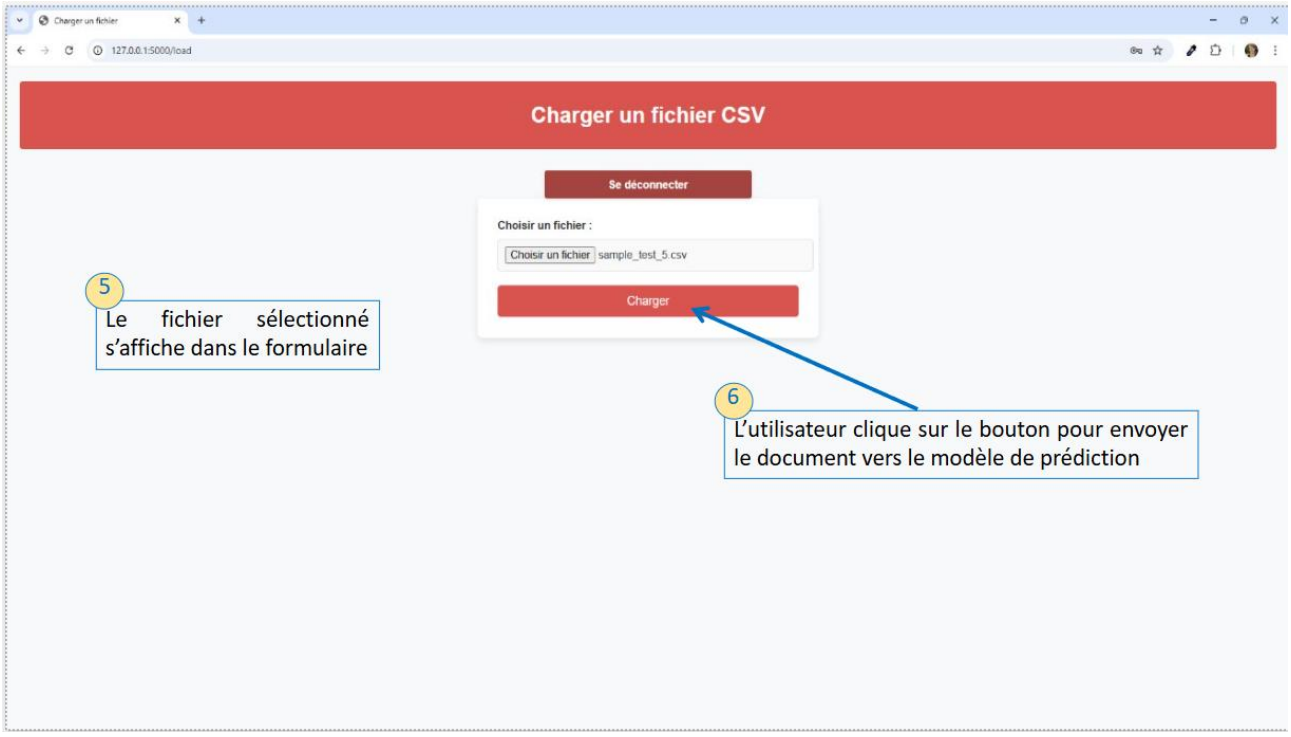
La route /predict est dédiée au traitement des données et à la prédiction des résultats.

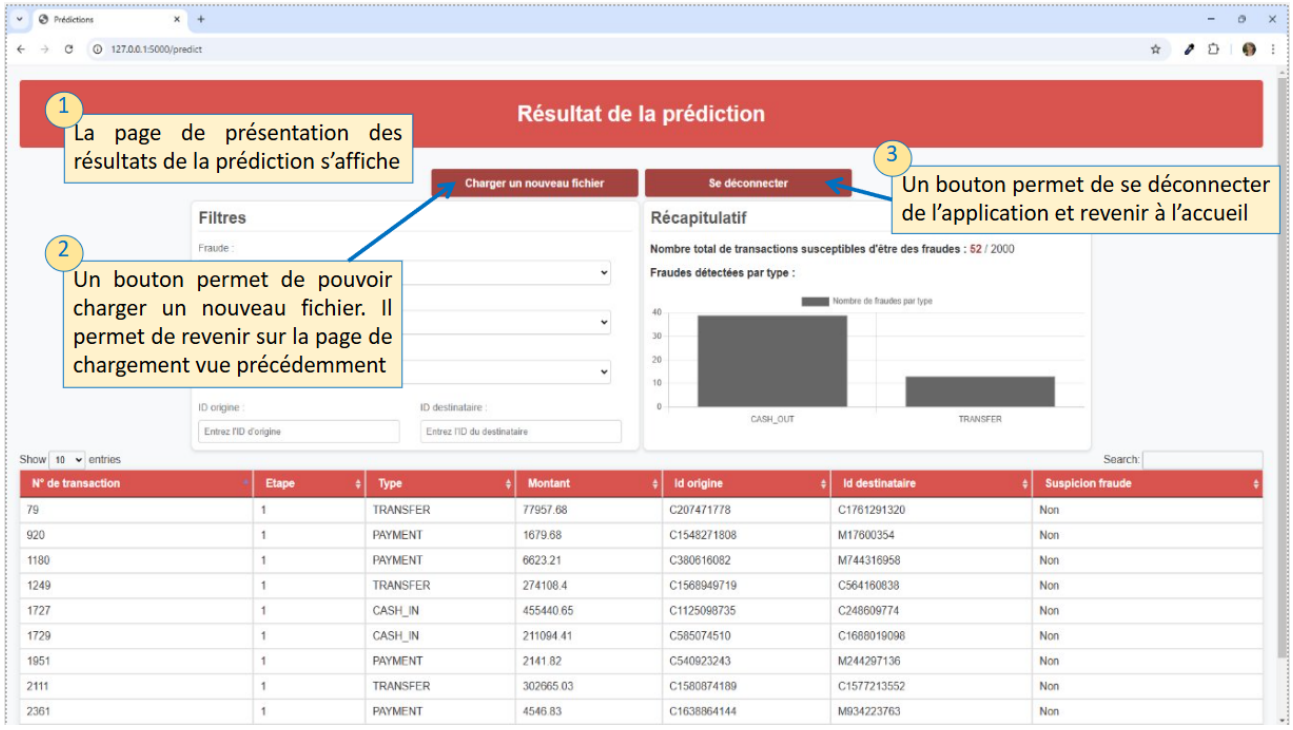
- Lecture du fichier : Le fichier CSV envoyé est chargé et stocké temporairement.
- Prétraitement des données : Normalisation et encodage des données pour assurer leur compatibilité avec le modèle.
- Prédiction : Le modèle préalablement entraîné est appliqué sur les données transformées.
- Filtrage et stockage des résultats :
- Des variables spécifiques sont générées (ex. : type, fraude ou non) pour permettre un filtrage des résultats sur l'interface utilisateur.
- Les prédictions sont insérées dans la table "predictions" de la base de données.

- Affichage des résultats : Les résultats sont envoyés à la page predict.html via le moteur de templates Jinja.

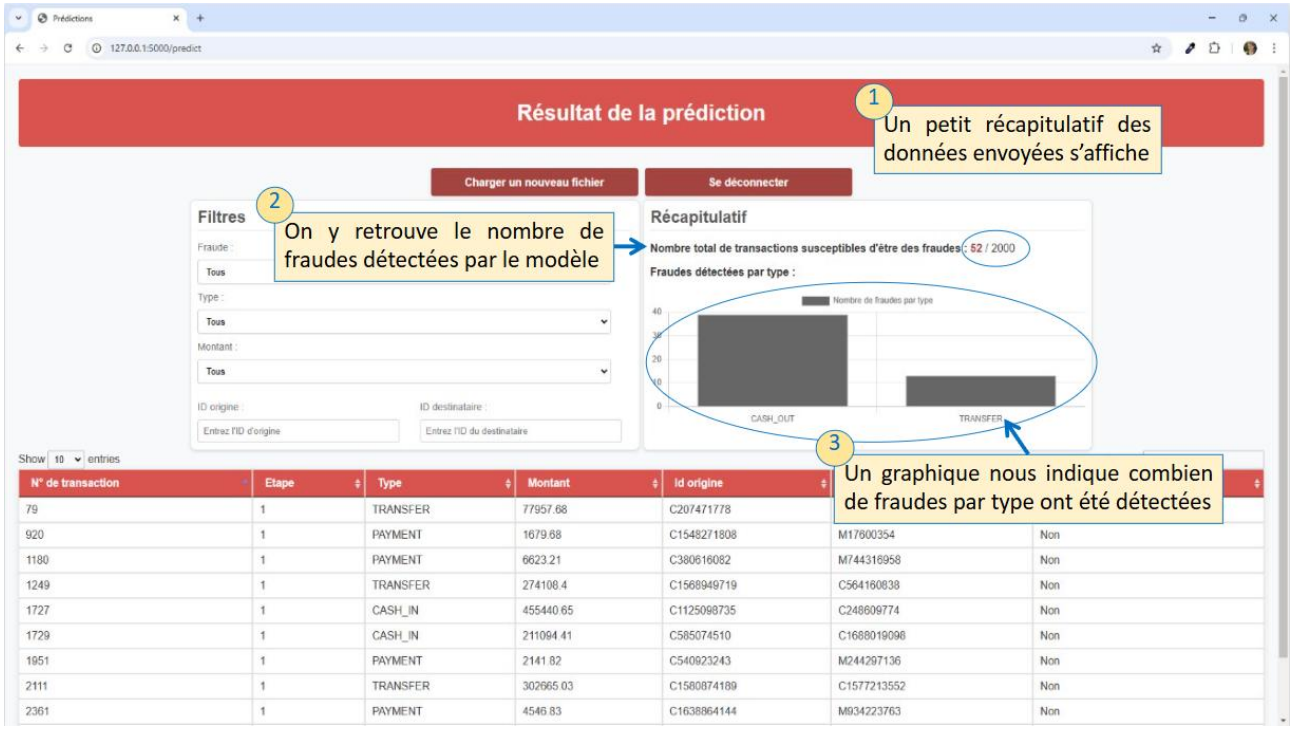



Charger un fichier CSV
Se déconnecter
Choisir un fichier :
Choisir un fichier
Aucun fichier choisi
Charger
4
Suite au clic sur « choisir un fichier » cela ouvre une fenêtre pour sélectionner un fichier

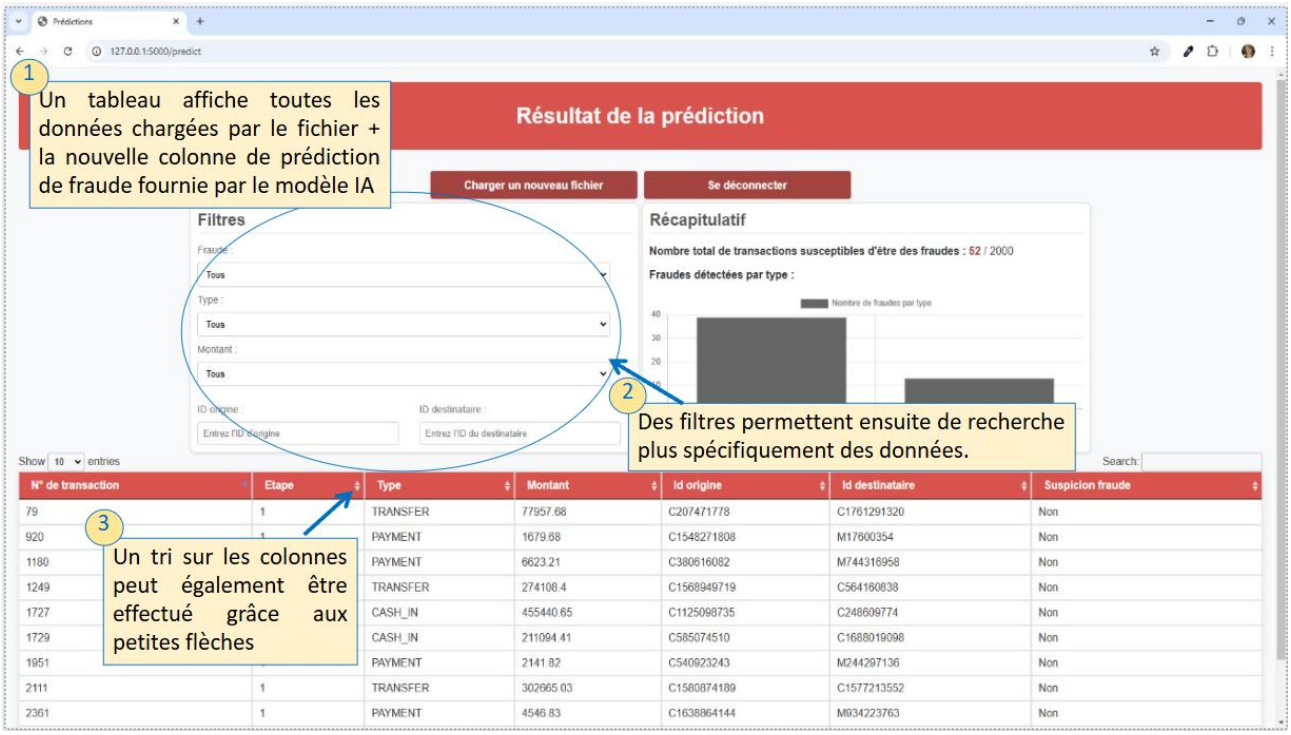

Charger un fichier CSV
Se déconnecter
Choisir un fichier :
Choisir un fichier
sample_test_5.csv
Charger
5
Le fichier sélectionné s'affiche dans le formulaire
6
L'utilisateur clique sur le bouton pour envoyer le document vers le modèle de prédiction



N° de transaction	Etape	Type	Montant	Id origine	Id destinataire	Suspicion fraude
79	1	TRANSFER	77957.68	C207471778	C1761291320	Non
920	1	PAYMENT	1679.68	C1548271808	M17600354	Non
1180	1	PAYMENT	6623.21	C380616082	M744316958	Non
1249	1	TRANSFER	274108.4	C1568949719	C564160838	Non
1727	1	CASH_IN	455440.65	C1125098735	C248609774	Non
1729	1	CASH_IN	211094.41	C585074510	C1688019098	Non
1951	1	PAYMENT	2141.82	C540923243	M244297136	Non
2111	1	TRANSFER	302665.03	C1580874189	C1577213552	Non
2361	1	PAYMENT	4546.83	C1638864144	M934223763	Non



N° de transaction	Etape	Type	Montant	Id origine	Id destinataire	Suspicion fraude
79	1	TRANSFER	77957.68	C207471778	C1761291320	Non
920	1	PAYMENT	1679.68	C1548271808	M17600354	Non
1180	1	PAYMENT	6623.21	C380616082	M744316958	Non
1249	1	TRANSFER	274108.4	C1568949719	C564160838	Non
1727	1	CASH_IN	455440.65	C1125098735	C248609774	Non
1729	1	CASH_IN	211094.41	C585074510	C1688019098	Non
1951	1	PAYMENT	2141.82	C540923243	M244297136	Non
2111	1	TRANSFER	302665.03	C1580874189	C1577213552	Non
2361	1	PAYMENT	4546.83	C1638864144	M934223763	Non



V. Difficultés rencontrées

a) Laetitia MONNIER

Dès le début du projet, nous avons été confrontés à un manque de méthodologie, car nous n'avions pas encore reçu de cours sur la gestion de projet. J'ai donc dû me documenter et improviser en m'appuyant sur mon expérience passée.

Un autre défi a été la gestion de l'équipe. Un membre n'était pas motivé, ce qui nous a contraints à travailler principalement à deux. Ensuite, lorsque Merwan est tombé malade, nous avons dû adapter notre organisation en travaillant à distance selon ses disponibilités et son état de santé. Cette situation a ajouté une pression supplémentaire pour respecter les délais.

Nous avons également rencontré des difficultés dans l'interprétation des attentes du projet. Le cahier des charges était peut-être trop léger, et nous avons dû interpréter certaines consignes à notre manière, sans être certains de répondre exactement aux attentes.

Sur le plan technique, j'ai éprouvé des difficultés avec l'Exploratory Data Analysis (EDA). J'ai eu du mal à savoir par où commencer et à interpréter correctement les résultats obtenus. De plus, je manque de formation sur la création de visualisations en Python, ce qui a compliqué cette étape.

b) Patrick-Sébastien DAILLIEZ

Travail de groupe.

c) Merwan BOUDRIAS

Pour ma part, je n'ai pas rencontré de grandes difficultés dans le projet.

Cependant, quelques préoccupations ont été levée lors des différentes phases du projet :

- EDA : Je me suis rendu compte de toute l'importance d'une bonne compréhension du problème et du jeu de données car elle peut rapidement devenir chaotique si elle n'est pas bien cadrée, entraînant une perte de direction.
- Feature Engineering : La crainte d'éliminer des caractéristiques essentielles qui pourraient avoir un impact significatif sur le modèle final.
- Modèle IA : beaucoup de modèle, de techniques d'échantillonnage, d'hyperparamètres à tester d'où la nécessité de connaître chaque modèle, ses qualités et ses défauts, et l'évaluation à l'overfitting.

De plus, je suis tombé malade durant 2 jours lors du projet et j'ai dû m'adapter en travaillant à distance afin de ne pas pénaliser le groupe.

VI. Perspectives d'évolutions

Bien que le cahier des charges ait été respecté, nous avons identifié des opportunités d'évolution pour chacune des phases du projet. Voici les détails :

Phase	Evolutions possibles
MCD – MLD – BDD	Lien entre l'utilisateurs et la table des prédictions pour connaître qui sauvegarde les prédictions. Ajouter des renseignements les tables origine et destinataire (attention à la RGPD)
EDA	Plus approfondir les données
Features Engineering	Garder les colonnes « montants » du destinataire et tester par rapport à cela.
Modèle IA	Affinage et réglage des hyperparamètres et des paramètres d'équilibrage suite à la validation du modèle (grâce à la learning curve)
API	• Faire un formulaire mot de passe oublié • Sécuriser la connexion • Ajouter un formulaire de saisie manuelle d'une ligne de transaction • Ajouter le compte qui utilise l'application • Changer les couleurs

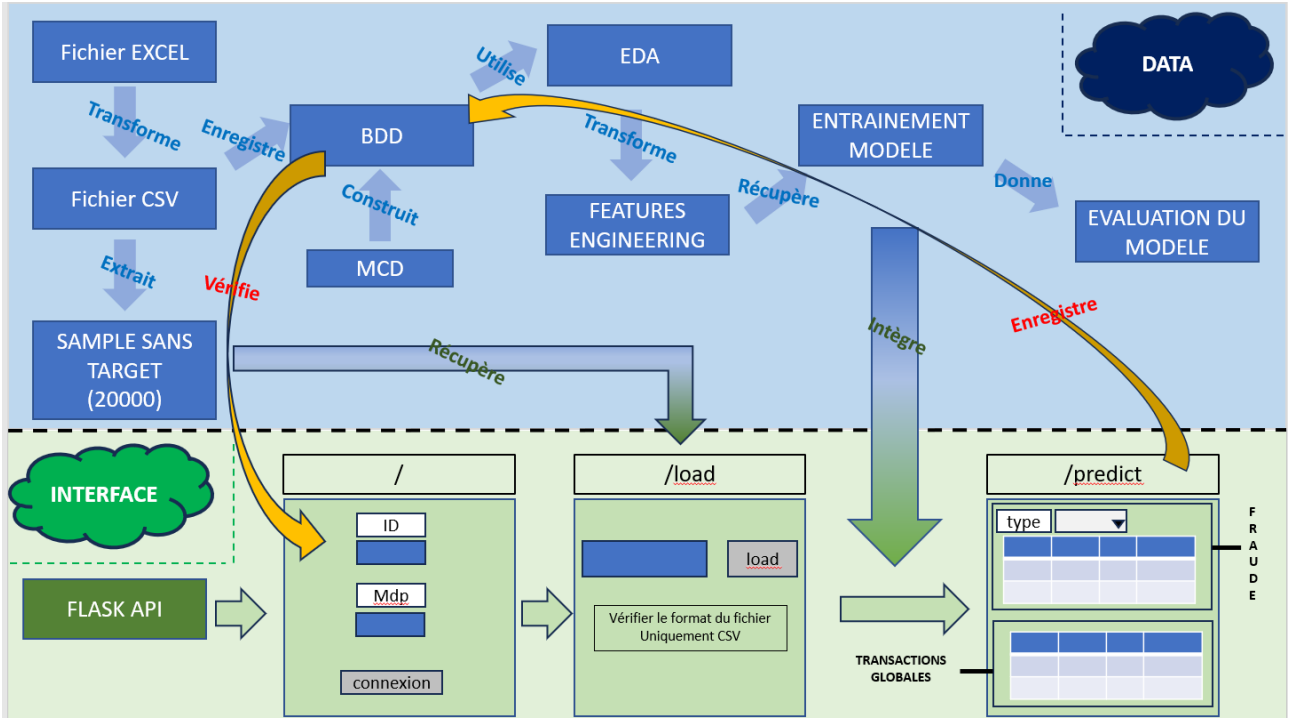
VII. Conclusion

Ce dossier a mis en évidence l'apport du machine learning dans la détection des fraudes, en détaillant les étapes clés du processus, de l'exploration des données à l'optimisation des modèles. L'**analyse exploratoire des données (EDA)** a permis de mieux comprendre la structure des transactions, tandis que des techniques comme le **Feature Engineering** et la **normalisation** ont contribué à améliorer la qualité des données utilisées pour l'entraînement des modèles.

L'approche par **classification** a été centrale dans l'identification des transactions frauduleuses, avec une optimisation des modèles via des méthodes comme **GridSearchCV**. Toutefois, des défis tels que l'**overfitting** peuvent impacter la capacité du modèle à bien généraliser sur de nouvelles données, ce qui souligne l'importance d'un ajustement rigoureux des **hyperparamètres** et d'une évaluation continue des performances.

L'intégration d'outils comme **Pandas, Seaborn** et **Matplotlib** a facilité l'analyse et la visualisation des données, rendant l'interprétation des résultats plus accessible. De plus, l'usage de métriques comme le **Recall** a permis d'évaluer la capacité du modèle à détecter efficacement les fraudes.

En conclusion, l'intelligence artificielle représente un atout majeur dans la lutte contre la fraude, mais elle nécessite une mise à jour régulière des modèles pour s'adapter aux nouvelles stratégies frauduleuses. Une veille technologique et une amélioration continue des algorithmes restent essentielles pour garantir des résultats fiables et efficaces.



VIII. Bilan de groupe

Ce qui s'est bien passé	**Ce qui s'est mal passé**
- Le travail à distance - L'adaptation aux imprévus	
A garder	**A jeter**
- L'écoute - Le lancement et l'approche du sujet - La compréhension de l'autre - Outil de gestion de projet : Asana - (Patrick-Sébastien)	- Les multiples dossiers et fichiers de travail

Suite à la réalisation de notre projet, nous avons identifié deux axes d'amélioration pour nos futurs travaux :

- Premièrement, nous avons constaté qu'un découpage plus fin des tâches en gestion de projet aurait permis une meilleure répartition du travail au sein de l'équipe. Cela aurait facilité le suivi de l'avancement.
- Deuxièmement, l'utilisation d'un système de gestion de versions comme Git pour le code source aurait été bénéfique. Un tel outil aurait simplifié la collaboration, le suivi des modifications et la gestion des différentes versions du code, contribuant ainsi à une meilleure organisation et à la réduction des risques d'erreurs.

IX. Bilans personnels

a) Laetitia MONNIER

Ce projet de classification de fraudes a été une expérience particulièrement enrichissante pour moi, notamment parce que c'était la première fois que j'endossais le rôle de chef de projet. Ne sachant pas comment structurer un projet au départ, j'ai dû me renseigner par moi-même et m'appuyer sur les expériences que j'avais pu observer auparavant. Malgré cela, j'ai le sentiment d'avoir bien géré cette responsabilité et d'avoir su orienter l'équipe dans la bonne direction.

J'ai particulièrement apprécié le travail en équipe, car cela m'a permis d'avoir des retours sur mon travail et d'échanger des idées pour améliorer nos analyses. J'ai aussi beaucoup aimé travailler avec Merwan, avec qui j'ai partagé la même vision sur le lancement du projet. Nous avons pris le temps de structurer notre approche plutôt que de nous lancer directement dans le code, ce qui nous a permis d'avoir une meilleure compréhension globale des étapes à suivre.

La répartition des tâches s'est faite de manière pragmatique, en veillant à ce que chacun puisse toucher à différentes étapes du projet. Malgré les difficultés rencontrées, nous avons su mener le projet à son terme et j'en suis fière. J'ai aussi la satisfaction de savoir que Merwan et moi avons compris toutes les étapes du projet, ce qui était un point essentiel pour moi.

Enfin, cette expérience m'a permis de développer des compétences en gestion de projet, en organisation et en adaptation aux imprévus. Elle met également en lumière les domaines techniques que je dois encore approfondir, notamment l'Exploratory Data Analysis (EDA) et la visualisation des données en Python.

b) Patrick-Sébastien DAILLIEZ

Le travail d'équipe c'est un support à la motivation. Pour ma part je trouve que Laetitia a bien géré en tant que chef de projet malgré le fait que ce soit la première fois à ce poste. J'ai trouvé que Merwan a été un bon technicien très investi. L'équipe a été à la hauteur de la tâche et a su relever le défi malgré mon absence et un intérêt peu poussé. Bonne continuation.

c) Merwan BOUDRIAS

Le travail en groupe a été une expérience enrichissante qui a renforcé mes compétences en collaboration et en communication au sein d'une équipe. La démarche structurée et claire du projet, avec ses étapes bien définies, m'a permis de développer mes compétences au sein d'une équipe. J'ai également amélioré mon efficacité dans toutes les phases du projet, de la planification à la réalisation, en passant par la conception et le développement. Enfin, j'ai acquis des connaissances solides et complètes sur les principes de classification et leurs applications concrètes dans différents contextes. Cette expérience a été très formatrice et m'a permis de progresser dans tous ces domaines.

X. Glossaire

Page 3

- **Asana** : Outil de gestion de projet collaboratif utilisé pour organiser les tâches et suivre leur avancement.
- **Fraude** : Transactions illégitimes détectées grâce aux modèles d'intelligence artificielle.

Pages 5-6

- **Base de données** : Structure permettant de stocker et d'organiser les transactions et autres informations du projet.

Pages 8-9

- **EDA - Analyse exploratoire des données (Exploratory Data Analysis)** : Processus d'analyse et de visualisation des données pour mieux comprendre leurs structures et caractéristiques.
- **Matplotlib** : Bibliothèque Python utilisée pour la visualisation des données.
- **Pandas** : Bibliothèque Python utilisée pour la manipulation et l'analyse des données.
- **Seaborn** : Bibliothèque Python utilisée pour la visualisation des données.

Pages 9-10

- **Features Engineering** : Processus de sélection et de transformation des variables pour améliorer la performance des modèles prédictifs.

Page 10, 15

- **Normalisation** : Transformation des variables pour qu'elles aient une échelle comparable.

Pages 12-15

- **Classification** : Technique d'intelligence artificielle utilisée pour catégoriser les transactions en frauduleuses ou non frauduleuses.
- **Fraude** : Transactions illégitimes détectées grâce aux modèles d'intelligence artificielle.
- **Machine Learning** : Approche d'intelligence artificielle utilisée pour entraîner des modèles capables de prédire les fraudes.
- **Modèle d'intelligence artificielle** : Algorithme utilisé pour identifier les transactions frauduleuses

Page 16

- **GridSearchCV** : Technique d'optimisation des hyperparamètres utilisée pour améliorer la performance des modèles d'apprentissage automatique.
- **Recall** : Mesure de performance qui indique la capacité du modèle à détecter les fraudes.

Pages 19

- **Hyperparamètres** : Paramètres ajustables qui influencent l'apprentissage des modèles de classification.

Pages 29

- **Overfitting (Sur-apprentissage)** : Situation où un modèle est trop adapté aux données d'entraînement et ne généralise pas bien aux nouvelles données.