Projet Classification

Auteurs :

Bart, Philippe, Thibaut, Salah, Hadjer

22/02/2024

A. Préambule

A.1. Contexte

Depuis des décennies l'utilisation des cartes de crédit pour réaliser de achats à augmenter de manière exponentielle. Le premier facteur fut la part de marché de plus en plus grande pour les achats en ligne. Ces derniers temps, ce fut une nouvelle accélération pour cause de pandémie. Les choses ont été rapides même trop rapide.

Vous travaillez au sein d'une société spécialisée dans la sécurité des systèmes bancaires. Vous disposez d'un simulateur de paiement Mobile Money. L'étude de cas de simulation de paiement Mobile Money est basé sur une entreprise réelle qui a développé une implémentation d'argent mobile qui offre aux utilisateurs de téléphones mobiles la possibilité de transférer de l'argent entre eux en utilisant leur téléphone comme un porte-monnaie électronique.

Une nouvelle vient d'arriver et les données sont dorénavant suffisantes pour mettre en place un système de prédiction des fraudes à la carte bancaire. La tâche à accomplir est de développer une approche qui détecte les activités suspectes qui sont révélatrices de fraude.

A.2. Planification

Pour la partie planification, nous avons décidé d'utiliser **Notion**, une application Web freemium de productivité et de prise de notes qui propose des outils d'organisation, notamment la gestion des tâches, le suivi des projets, les listes de tâches et la mise en favoris. Le tableau qui suit décrit notre organisation.

1. Planification de la réunion de lancement ;
2. Rédaction d'une proposition de projet ;
3. Recontextualisation du projet : *A partir de notre base de données (fichier* `credit_card_fraud.csv`), développer un modèle de prédiction qui permet determiner si une transaction est une fraude ou non ;
4. Réalisation d'un MCD/MPD ;
5. Construction de la base de données ;
6. Dataviz
7. Choix, construction et évaluation de quatres modèles de classification : `XGBoost`, `AdaBoost`, `KNN`, `RandomForest` ;

8. Construction et réalisation de la maquette.

I.

Guide Utilisateur

L'application que nous avons conçu est composé de trois pages HTML :

`index.html`

Page d'acceuil, qui permet de selectionner le type d'utilisateur (**client** ou **opérateur**). Elle permet à l'utilisateur de *créer son compte* ou de se *connecter*.

- **Inscription** – *si l'utilisateur choisi de s'inscrire, les informations fournies sont enregister dans la table* `utilisateur` *de notre base de donnée puis il est rédiriger directement sur la page correspondante à son role. Si l'utilisateur choisis de se réinscrire, son mot de passe est simplement mis à jour.*
- **Connexion** – *l'utilisateur est envoyé vers la page qui correspond à son statut.*

`client-home.html`

Page désigner pour un client. Elle lui permet de faire une transaction, en specifiant le **type**, **montant** et le **nom destinataire**.

Dû à l'incomplétude de données sur les utilisateurs, cette page web fait office de décor.

`operator-home.html`

En spécifiant le spectre entier des informations nécessaires pour le modèle, elle permet à un opérateur de **prédire** si la transaction fournie est une fraude ou pas. Elle permet aussi d'afficher l'ensemble des transactions *non-vérifié* ainsi que de *filtrer* pas le nom d'utilisateur.

II.

Documentation Technique

(*Description de l'analyse du fichier source*)

II.1 Base de donnée

II.1.1 Nettoyage du ficher source

Le fichier ressource qui nous avaient était fournis (`credit_card_fraud.csv`) avait eu besoin d'un petit nettoyage :

- Les guillemets on étaient supprimées ;
- Les virgules ont étaient replacés par des point, pour que les nombres soient proprement interprétés par la `DataFrame` ;
- Les points-virgules furent replacés simplement par des virgules ;
- Les attributs de types numérique, ont étaient convertis en `float` ou en `int` ;
- Les valeurs des champs `nameOrig` et `nameDest`, furent convertis en numérique avec le script :

```
1  def replace_first_letter(value):
2      if value.startswith('C'):
3          return '1' + value[1:]
4      elif value.startswith('M'):
5          return '2' + value[1:]
6      else:
7          return value
```

II.1.2 MCD

La construction de la base de données a commencé par la réalisation du MCD, et par le passage au MPD avec le logiciel **Looping**. Une fois celui-ci terminé, nous avons utilisé le module `mysql.connector` pour établir une connexion, et pour la construction de notre base de données depuis Python.

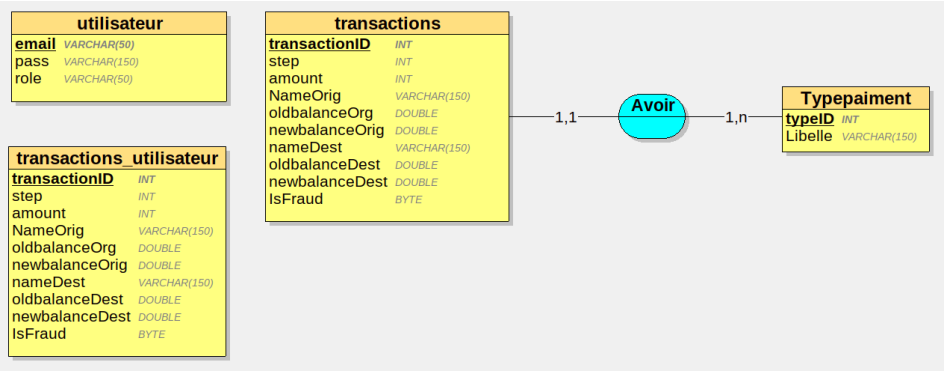

FIGURE II.1 – MCD à partir du fichier `credit_card_fraud.csv`

II.1.3 MLD

```
typepaiment = (
    typeID INT,
    libelle VARCHAR(150)
);

utilisateur = (
    email VARCHAR(50),
    pass VARCHAR(150),
    role VARCHAR(50)
);

transactions_utilisateur = (
    transactionID INT,
    step INT,
    amount INT,
    nameOrig VARCHAR(150),
    oldbalanceOrg DOUBLE,
    newbalanceOrig DOUBLE,
    nameDest VARCHAR(150),
    oldbalanceDest DOUBLE,
    newbalanceDest DOUBLE,
    isFraud BYTE
);

transactions = (
    transactionID INT,
    step INT,
    amount INT,
    nameOrig VARCHAR(150),
    oldbalanceOrg DOUBLE,
    newbalanceOrig DOUBLE,
    nameDest VARCHAR(150),
    oldbalanceDest DOUBLE,
    newbalanceDest DOUBLE,
    isFraud BYTE,
    #typeID
);
```

II.2 Construction des modèles

Après avoir étudié soigneusement les données fournies, nous avons considéré quatre modèles qui pouvaient convenir au dataset : `XGBoost`, `AdaBoost`, `KNN` ainsi que `RandomForest`.

II.2.1 Entrainement des modèles

L'entrainement de chaque modèles `XGBoost`, `KNN` et `RandomForest` a était fait par la méthode `GridSearchCV`, tandis que le modèle `AdaBoost` fut simplement entrainer avec `StratifiedKFold`. Voici la liste des hyperparamètres choisis pour chacun des modèles :

`XGBoost`

- $1000 \leqslant$ `n_estimators` $\leqslant 1400$
- $7 \leqslant$ `max_depth` $\leqslant 9$

Le meilleur des estimateur pour ce dataset a été :

`KNN`

- $2 \leqslant$ `n_neighbors` $\leqslant 11$
- `weights` $\in$ `['uniform', 'distance']`
- `p` $\in$ `[1, 2]`
- `algorith` $\in$ `['ball_tree', 'kd_tree']`

Le meilleur des estimateur pour ce dataset a été :

`RandomForest`

- $100 \leqslant$ `n_estimators` $\leqslant 300$
- $1 \leqslant$ `max_depth` $\leqslant 30$
- `criterion` $\in$ `['entropy', 'gini']`

Le meilleur des estimateur pour ce dataset a été : `n_estimators` $= 200$, `criterion` $=$ `'entropy'`, `max_depth` $= 20$.

II.2.2 Evaluation des modèles

Modèle	Accuracy	Precision	Recall	F1 Score	AUC
KNN					
XGBoost					
AdaBoost					
RandomForest					

Table II.1 – Métriques calculés pour les modèles entraînés

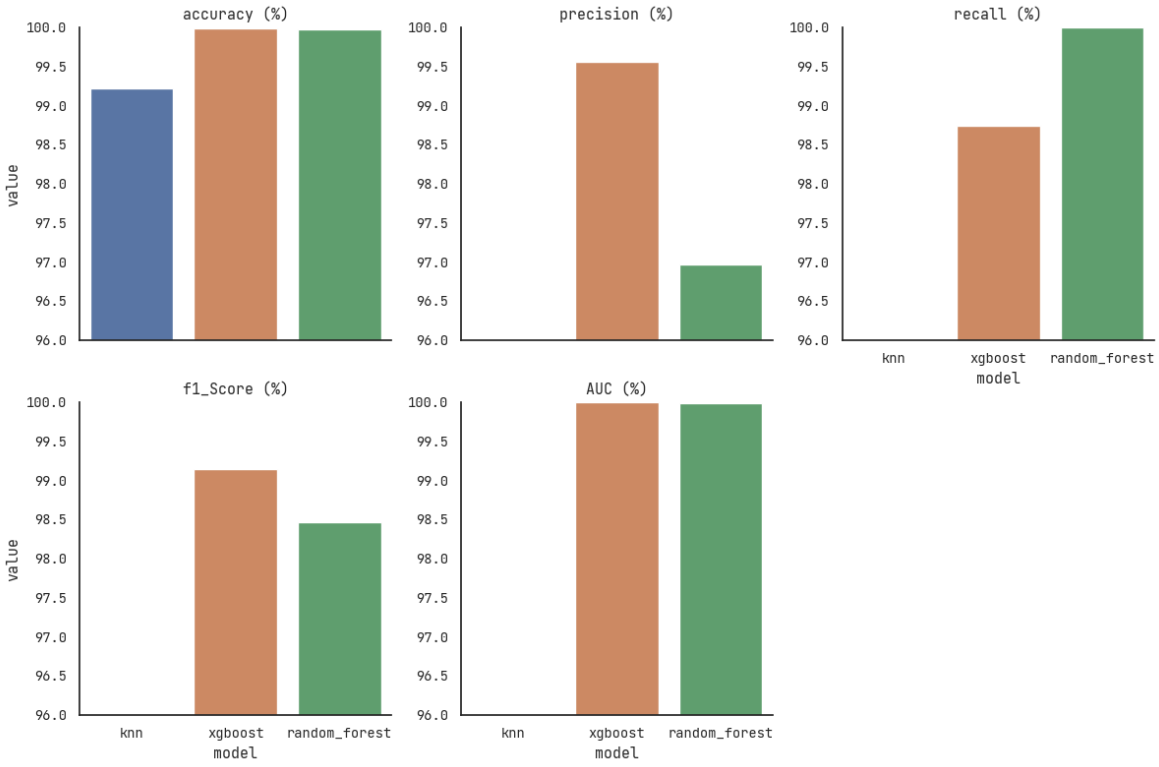

FIGURE II.2 – Comparaison des modèles

II.3 Realisation de l'application

La création de l'application a commencé bien évidemment par la création de la maquette, nous avons utilisé le logiciel **Figma** pour cette tâche. Une fois celle-ci réalisée, nous avons commencé par structurer notre projet.