

Projet 7 : Implémentation d'un modèle de scoring

OCTOBRE 2022



PARTIE 1 :

PRÉSENTATION DE LA MISSION



Société : Prêt à dépenser

- Ma mission en tant que data Scientist chez l'entreprise financière « Prêt à dépenser »:
 - Mise en place d'un outil « scoring crédit » pour calculer la probabilité qu'un client rembourse son crédit.
 - Classification de la demande en crédit accordé ou refusé.

PARTIE 2 :

PROPOSITION DE SOLUTION



ELABORATION DU MODELE

- Nous avons utilisé le modèle lightGBM permettant de :
 - ✓ réaliser une classification qui peut fournir la probabilité d'appartenance à une classe (predict_proba)
 - ✓ d'entraîner le modèle sur un jeu de données :
 - split en training et testing
 - enregistrement du modèle sous-format pickle pour ne pas ré-entraîner le modèle.

TECHNOLOGIE MISE EN ŒUVRE

- Hébergement de l'application (VPS, Virtual Private Server, avec OVH)
 - en raison de sa capacité pour 1Go de RAM
 - en 40 Go de mémoire SSD (très fluide).
- Création d'une API pour fournir les données au front via Flask.
- Création d'une interface web qui met à disposition un dashboard - interactif disponible sur la machine VPS (port 80)
- Outil de versioning GIT utilisé pour échanger les codes avec le serveur OVH et pour enregistrer des versions du code tout au long du projet.

PARTIE 3 :

DESCRIPTION DES MODULES



API

- Création d'une API pour fournir les données au front via Flask.
 - récupération du modèle pickle enregistré
 - création de routing générique et dynamique dépendant d'un id client.
- Création d'un dossier dash et API indépendants qui communiquent via « **requests** » comme ci-après:

```
CURRENT_CLIENT_ID = 193423
req=requests.get(f"http://127.0.0.1:5000/get_score/{CURRENT_CLIENT_ID}").json()
info_client=requests.get(f"http://127.0.0.1:5000/get_info_client/{CURRENT_CLIENT_ID}").json()
info_client=pd.Series(info_client['info_client']).to_frame().T.iloc[:,2:]
# print(info_client)
id_client=requests.get("http://127.0.0.1:5000/get_id_client").json()
```


Description du dashboard

- Mise en place d'une communication interactive entre le dashboard et l'API en utilisant « request api » pour récupérer les données et ensuite utiliser le système de callback pour rendre l'application dynamique.
- L'utilisation de callback simple d'un formulaire pour mettre à jour l'application de manière dynamique.
- Soucieux d'avoir une volonté de transparence pour accorder les prêts, j'ai décidé de :
 - faire un dashboard interactif disponible via une interface web (afin que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d'octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement).
- La mise à disposition du code sur le Git.
- Lien du dashboard VPS OVH: <http://vps-c51efc30.vps.ovh.net/>

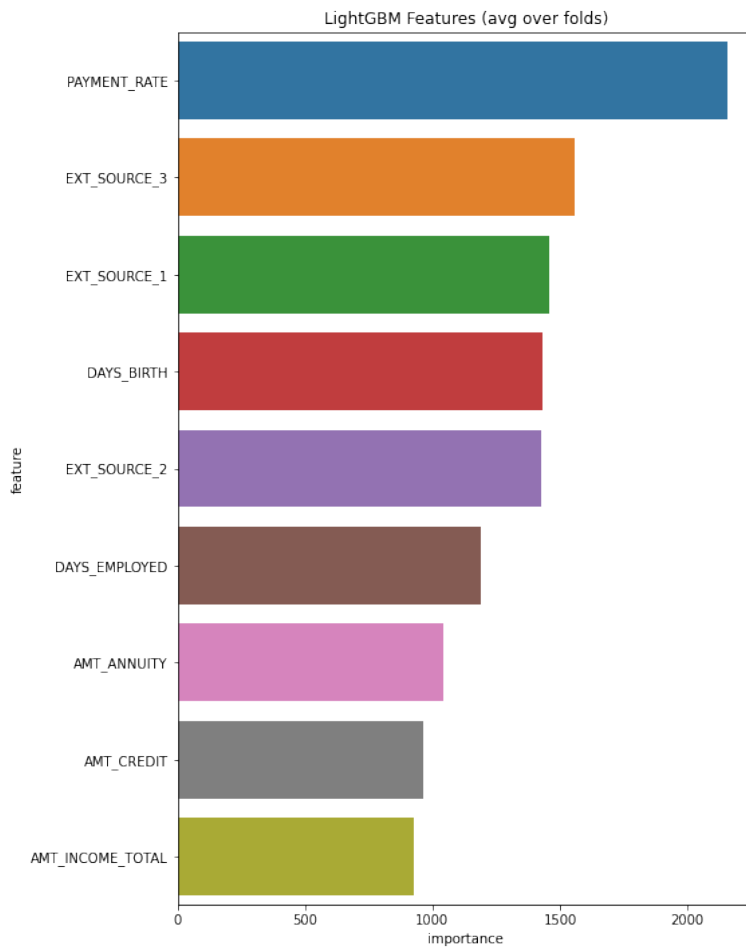
Disponibilité du dash et API 24/24

- Il est important que le fonctionnement du dash et l'API ne dépendent pas du terminal GNU/Linux pour le rendre disponible, d'où sa mise à disposition sur le VPS OVH pour qu'ils soient disponibles de manière permanents avec une **continuité d'activité du serveur** via la commande « **nohup python** » avec **une tâche de fond**.
 - Raison de non disponibilité des informations sur le dash et API sur le PC individuel : erreur humaine, interruption de service, ordinateur en veille, arrêté, tombé en panne, mémoire insuffisant RAM ou SSD).
- La mise à disposition du code sur le Git.
- Lien de l'API : http://15.237.51.88/get_score/111767 (sur le serveur EC2 Amazon) avec tâches de fonds.
- Lien du dashboard VPS OVH: <http://vps-c51efc30.vps.ovh.net/>

Exemple du graphique de dash en ligne

<http://vps-c51efc30.vps.ovh.net/>

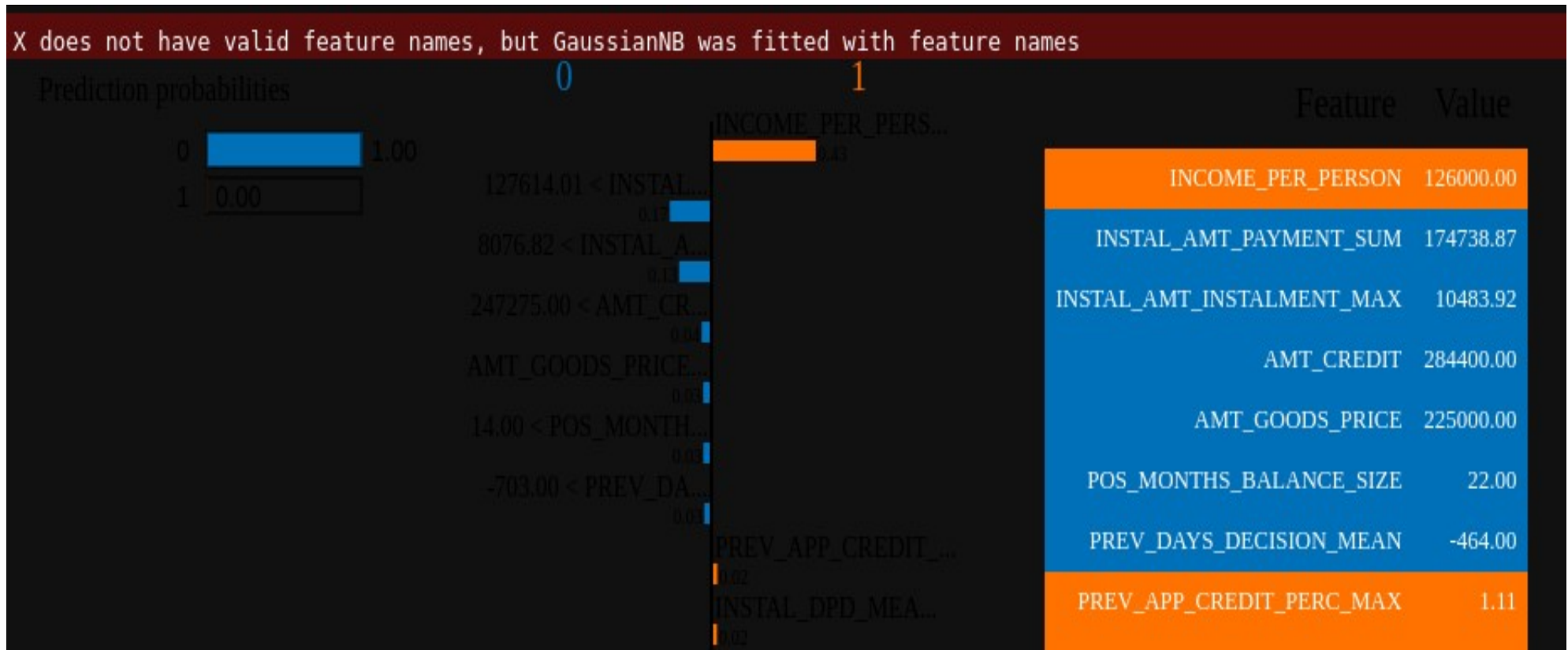
Features importance globale



Prédiction du score de solvabilité client



Features locales

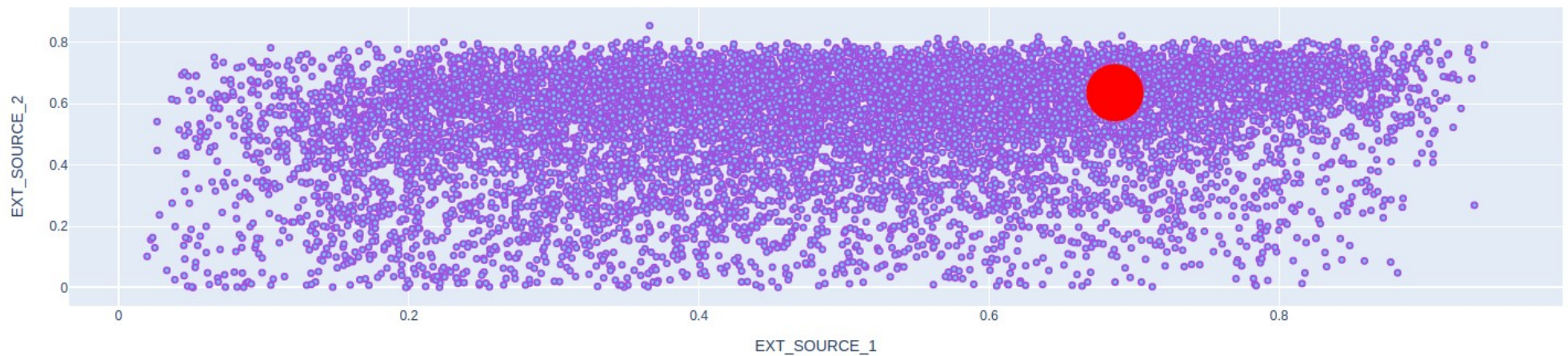


Les valeurs de Shap sont représentées pour chaque variable dans leur ordre d'importance, chaque point représente une valeur de Shap , pour un exemple.

Exemple du graphique de dash en ligne

<http://vps-c51efc30.vps.ovh.net/>

Graphique de corrélation bivariée entre 2 features avec marker sur le score



**Je vous remercie pour votre
attention**



