Prêt à dépenser

Projet 7

Implémentez un modèle de scoring

Par **Paul Smadja**
Mentoré par
Walid Ayadi
Février 2021

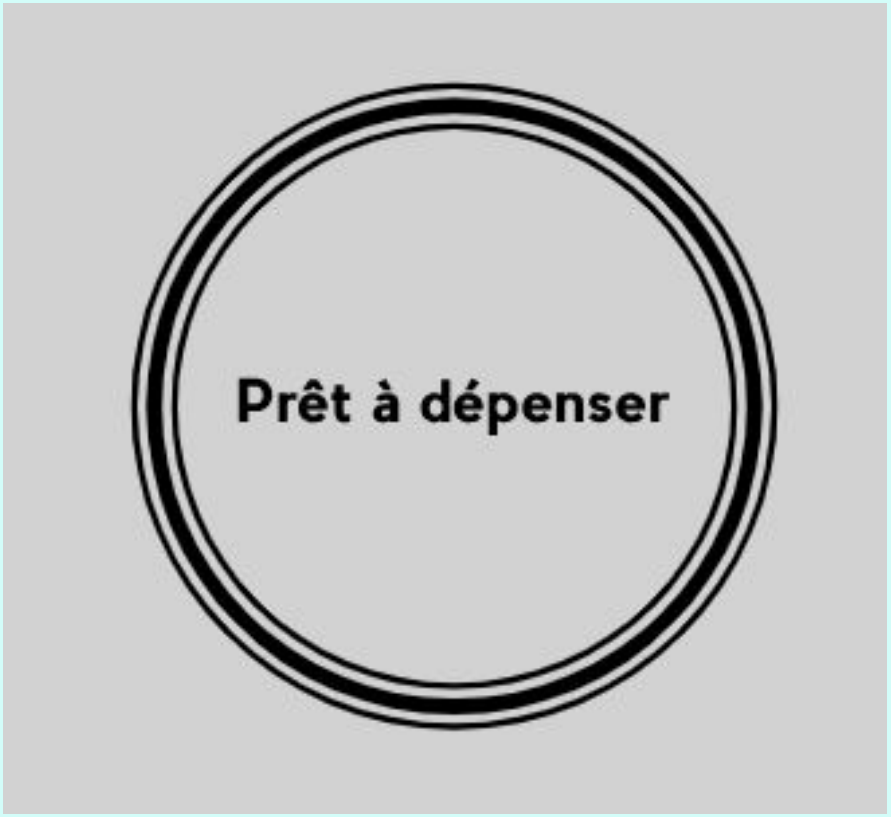
Sommaire

Partie 1 (5 min) – Problématique & Présentation du jeu de données

Partie 2 (7 min) –Explication de l'approche de modélisation

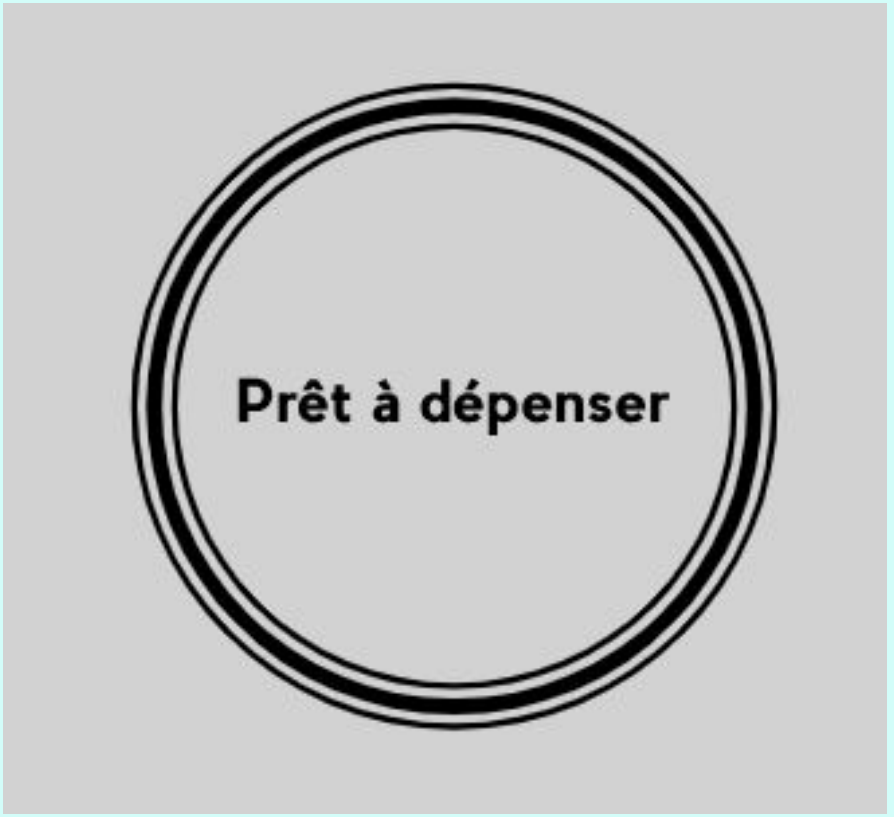
Partie 3 (15 min) – Présentation du tableau de bord

Partie 4 (5 à 10 min) – Questions & Réponses

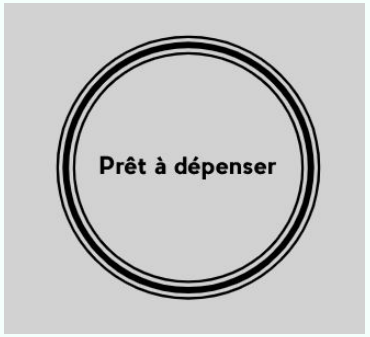


Partie 1

Problématique & Présentation du jeu de données



Contexte & Mission



Prêt à dépenser : société financière de **prêts à la consommation**

⇒ personnes ayant **pas ou peu d'historique** de prêts

Missions :

Développement d'un modèle de *scoring*

⇒ **probabilité de défaut de paiement** du client

⇒ déploiement du modèle sous forme d'**API**

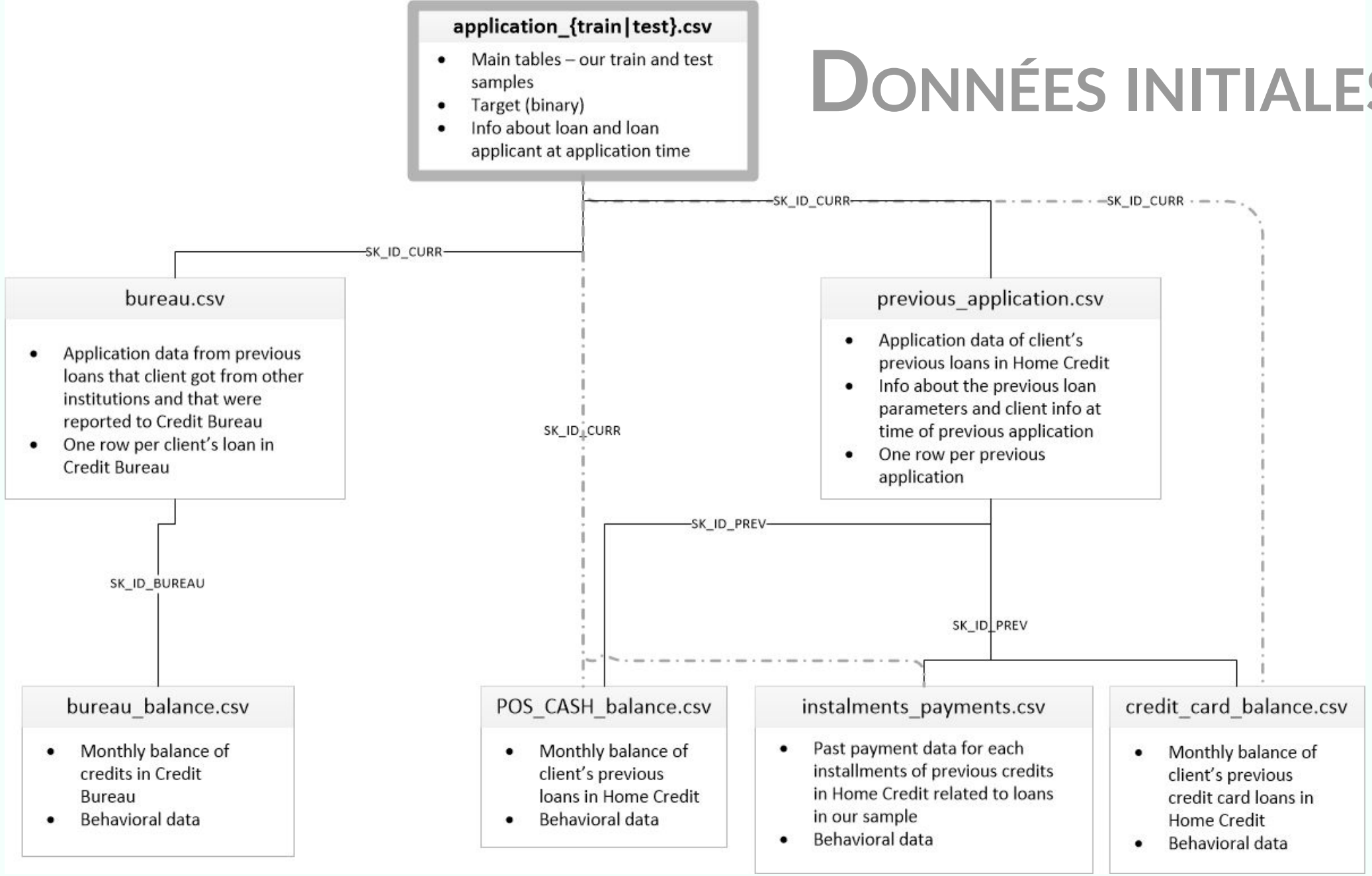
⇒ réalisation d'un **tableau de bord interactif**

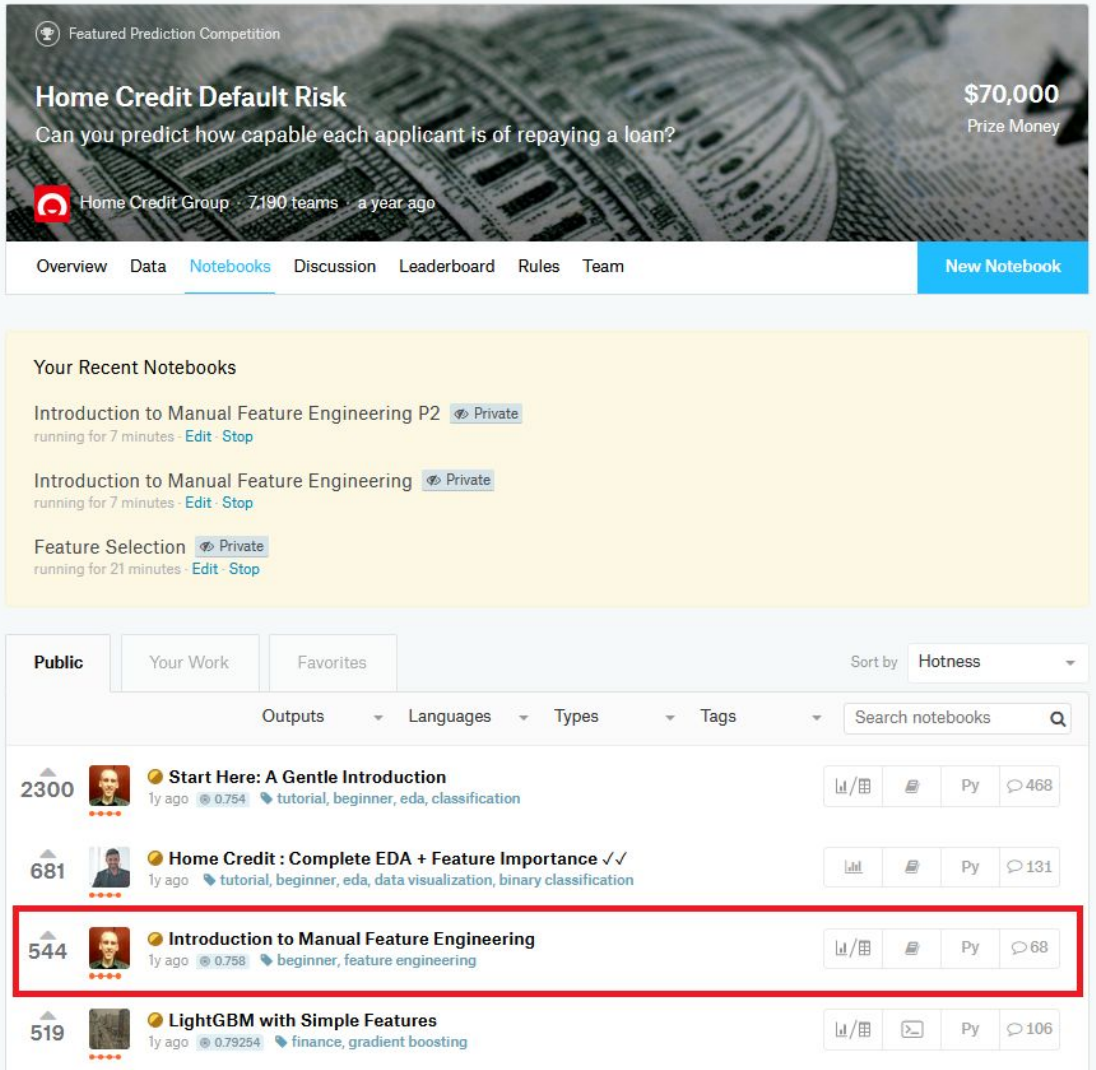
⇒ **interprétabilité** du modèle

⇒ rédaction d'une **note méthodologique**

⇒ utilisation d'un **outil de versionnage**

DONNÉES INITIALES





Kernels **Kaggle** de Will Koehrsen

- ⇒ *Start Here: A Gentle Introduction*
- ⇒ *Introduction to Manual Feature Engineering*
- ⇒ *Introduction to Manual Feature Engineering P2*
- ⇒ *Feature Selection*

*=> Taille des données : **289'000 x 215***

PRESENTATION DU NOTEBOOK KAGGLE

Data train
Data test

- Rappel : "test.csv" est le dataset que nous utilisons pour simuler un nouveau client dans la base. Toutefois il convient que ces deux datasets aient la même structure à l'issu du feature engineering.

Valeurs manquantes

- Traitement par imputation de la médiane

Encodage variables

- Label encoding pour les variables à 2 catégories.
- One Hot Encoding pour les variables à plus de deux catégories.

Alignement datasets

- Alignement des datasets "train" et "test" pour conserver des structures identiques.

Création de variables

- Remplacement des outliers par des valeurs nulles. Ensuite les valeurs sont imputées par la médiane dans le Preprocessing.
- Ajout d'une "flag feature" pour identifier les lignes qui contiennent les outliers.

Hypothèses

- Création de deux hypothèses de feature engineering :
  - *"Weighted Features"*
  - : Amélioration de la corrélation des variables EXT SOURCES avec la target
  - *"Domain Features"* : Construction de variables s'appliquant plus au domaine de la banque comme :
    - "CREDIT_INCOME_PERCENT"
    - "ANNUITY_INCOME_PERCENT"
    - "CREDIT_TERM"
    - "DAYS_EMPLOYED_PERCENT"

Partie 2

Explication de l'approche de modélisation

Méthodologie

Métriques pour notre modèle de classification :

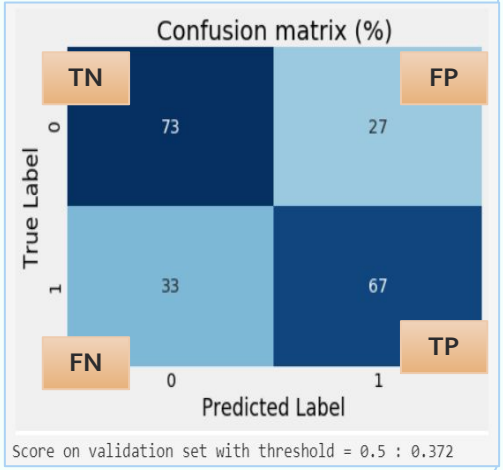
- $gain = TP \cdot TP\_value + TN \cdot TN\_value + FP \cdot FP\_value + FN \cdot FN\_value$
- $max\_gain = N \cdot TN\_value + P \cdot TP\_value$
- $baseline = (TN + FP) \cdot TN\_value + (TP + FN) \cdot FN\_value$

$$=> score = \frac{gain - baseline}{max\_gain - baseline} \in [0; 1]$$

$$=> model\_score = \max_{threshold \in [0;1]} [score] \in [0; 1]$$

La matrice de confusion :

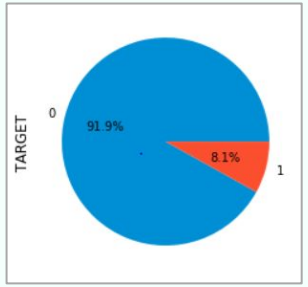
La matrice de confusion consiste à compter le nombre de fois où des observations de la classe 0 ont été rangées dans la classe 1. Par exemple, si nous voulons connaître le nombre de fois où le classifieur à bien réussi à classer une classe 1, on examinera la cellule à l'intersection de la ligne 1 et de la colonne 1.



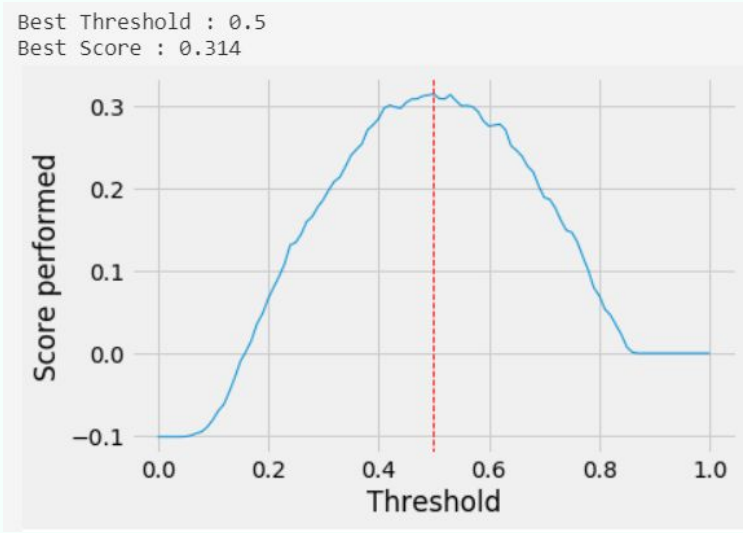
Explication des targets / Déséquilibre de la population:

Nous avons à faire à un problème de classification binaire où la population est fortement déséquilibrée.

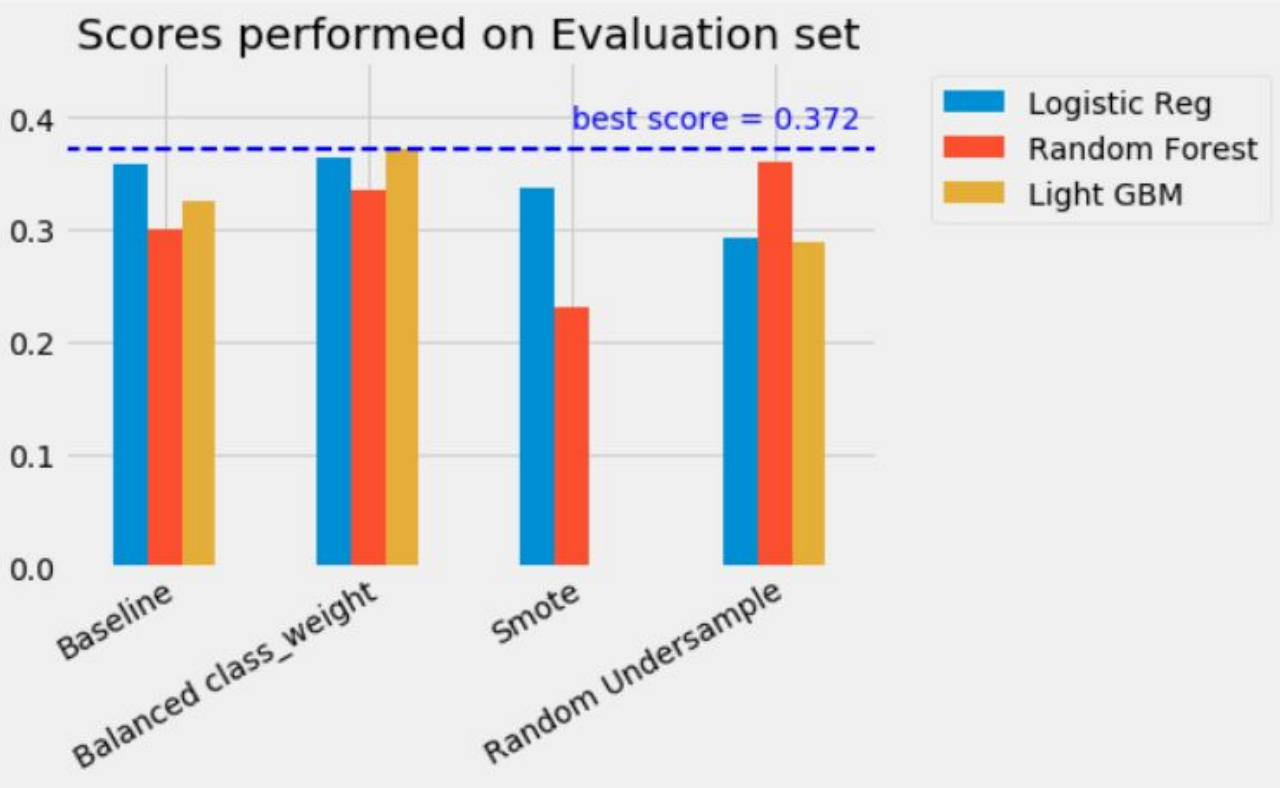
Plusieurs méthodes de rééquilibrage ont été mise en oeuvre.

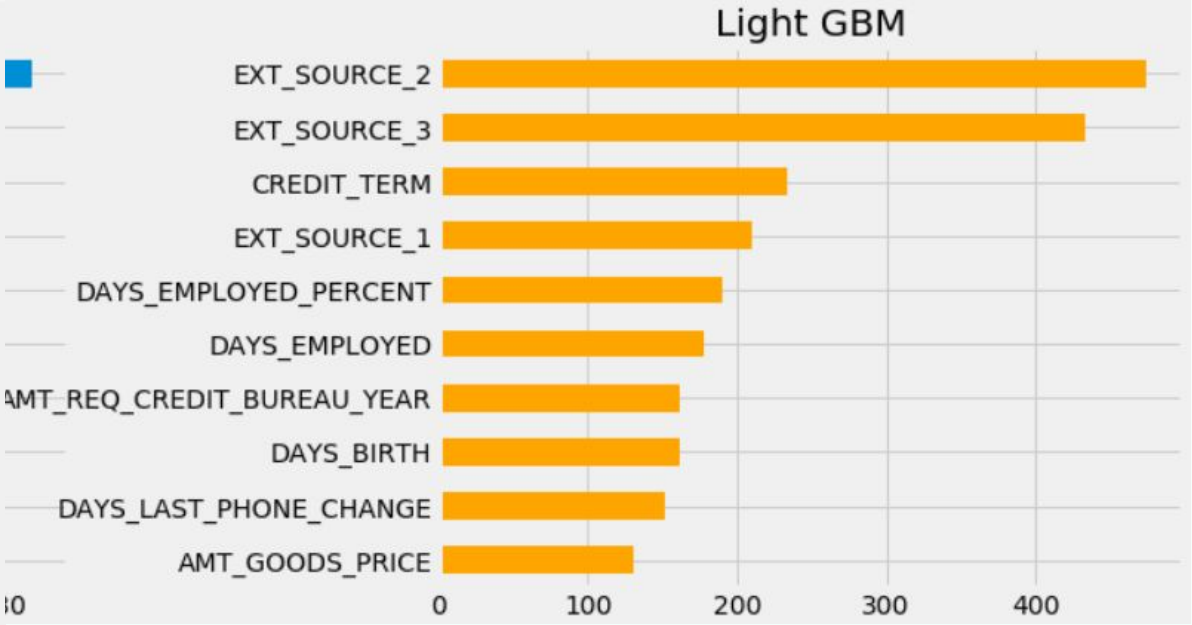


Sélection du Threshold:

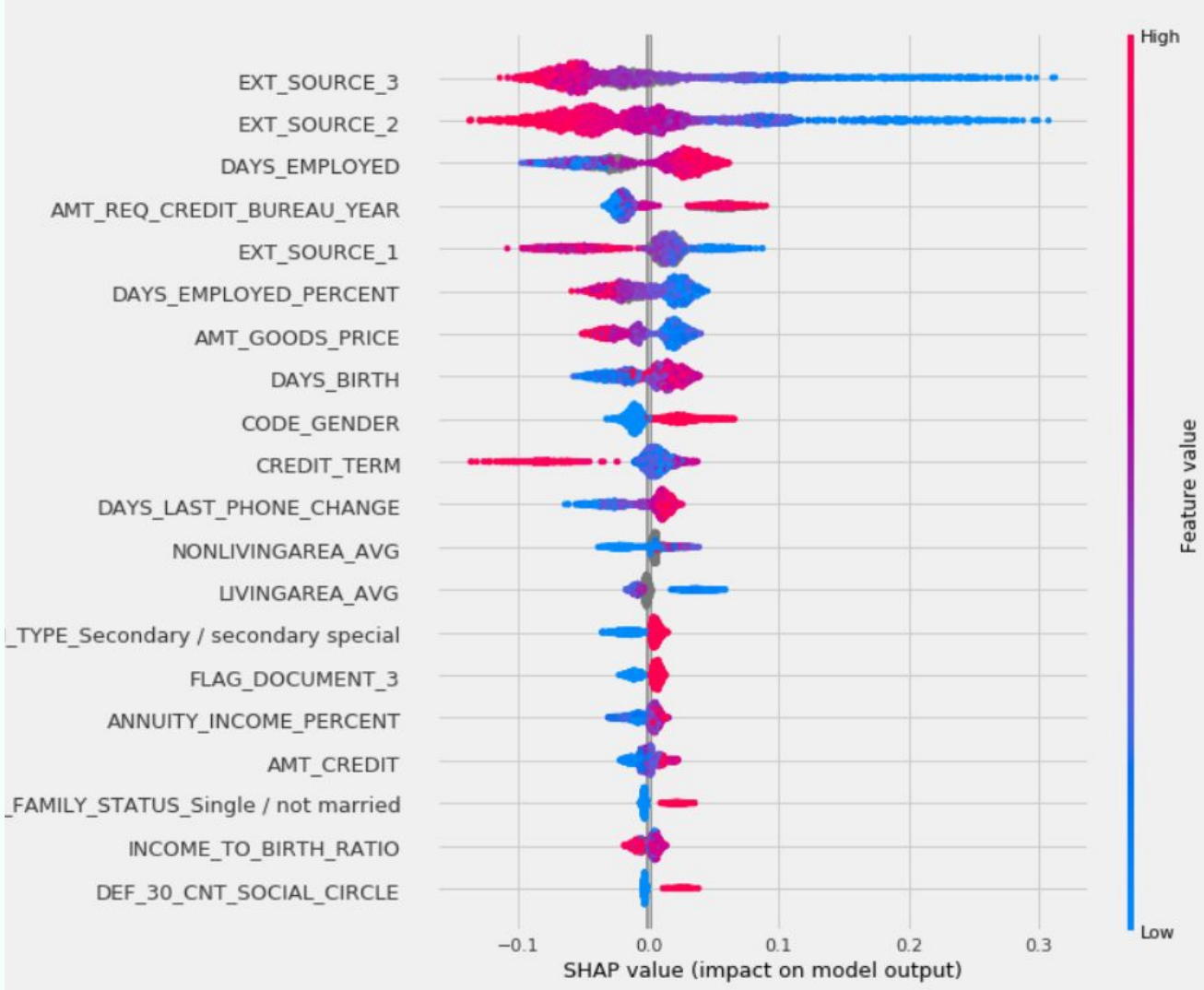


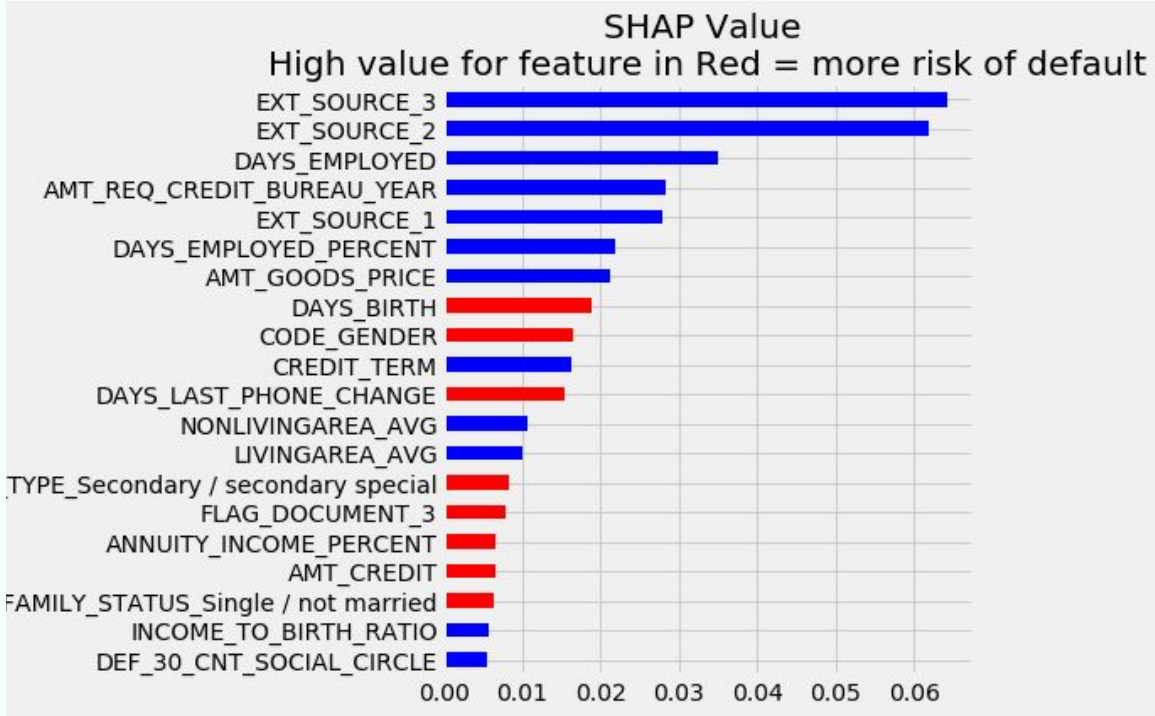
Résultat


Scores performed on Evaluation set
best score = 0.372
Logistic Reg
Random Forest
Light GBM
0.4
0.3
0.2
0.1
0.0
Baseline
Balanced class_weight
Smote
Random Undersample

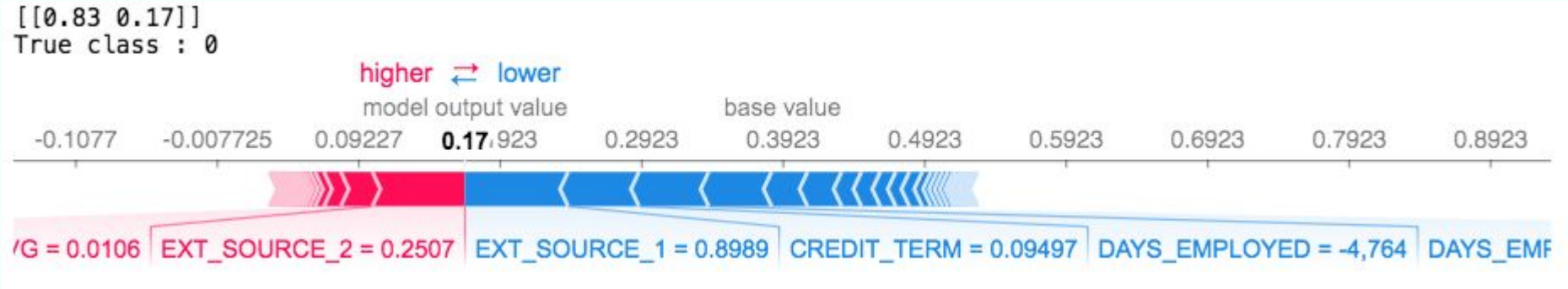

Light GBM
EXT_SOURCE_2
EXT_SOURCE_3
CREDIT_TERM
EXT_SOURCE_1
DAYS_EMPLOYED_PERCENT
DAYS_EMPLOYED
AMT_REQ_CREDIT_BUREAU_YEAR
DAYS_BIRTH
DAYS_LAST_PHONE_CHANGE
AMT_GOODS_PRICE
0
100
200
300
400

Interpretation Globale





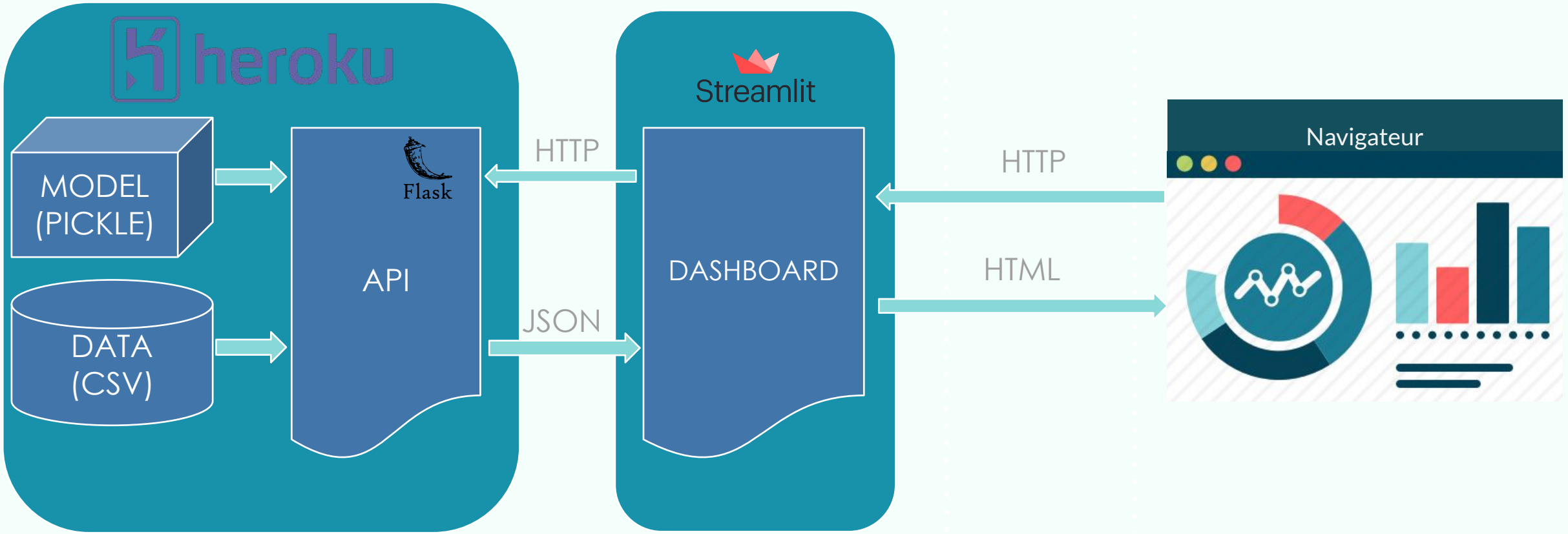
Interpretation Locale



Partie 3

Présentation du tableau de bord

ARCHITECTURE API/DASHBOARD



Click Me to see the dashboard !!!

RESUME

CLASSIFICATION

- Construction d'un modèle de classification binaire à partir d'un Kernel de départ téléchargé sur Kaggle.
- Une population fortement asymétrique (92% - 8%)

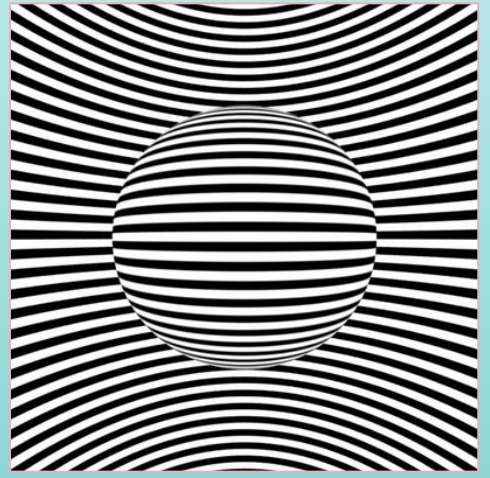
API / DASHBOARD

- Création d'une API web avec Flask pour le côté serveur, et Streamlit pour le côté dashboard.

AXES D'AMELIORATION

- Une recherche de performances de prédiction plus approfondie, avec réseaux de neurones par exemple.
- Une optimisation plus fine en étudiant plus en détails chaque hyperparamètre.

PROFIL GitHub



L'ensemble des fichiers de ce projet ont été stockés sur mon compte GitHub :
Code API : https://github.com/SmadjaPaul/Projet7_API
Code DashBoard : https://github.com/SmadjaPaul/Projet7_Dashboard
Analyse : https://github.com/SmadjaPaul/DATASCIENCE-PROJECT/tree/master/OPC_P7_SCORE

Merci !

Des Questions ?

Par **Paul Smadja**
Mentoré par
Walid Ayadi
2021