



Parcours OpenClassRooms

Data Scientist

P7 Déployer un modèle via API et interface web

Pictures used for educational purpose only

Sommaire

I. Introduction

II. L'approche de modélisation

III. L'API

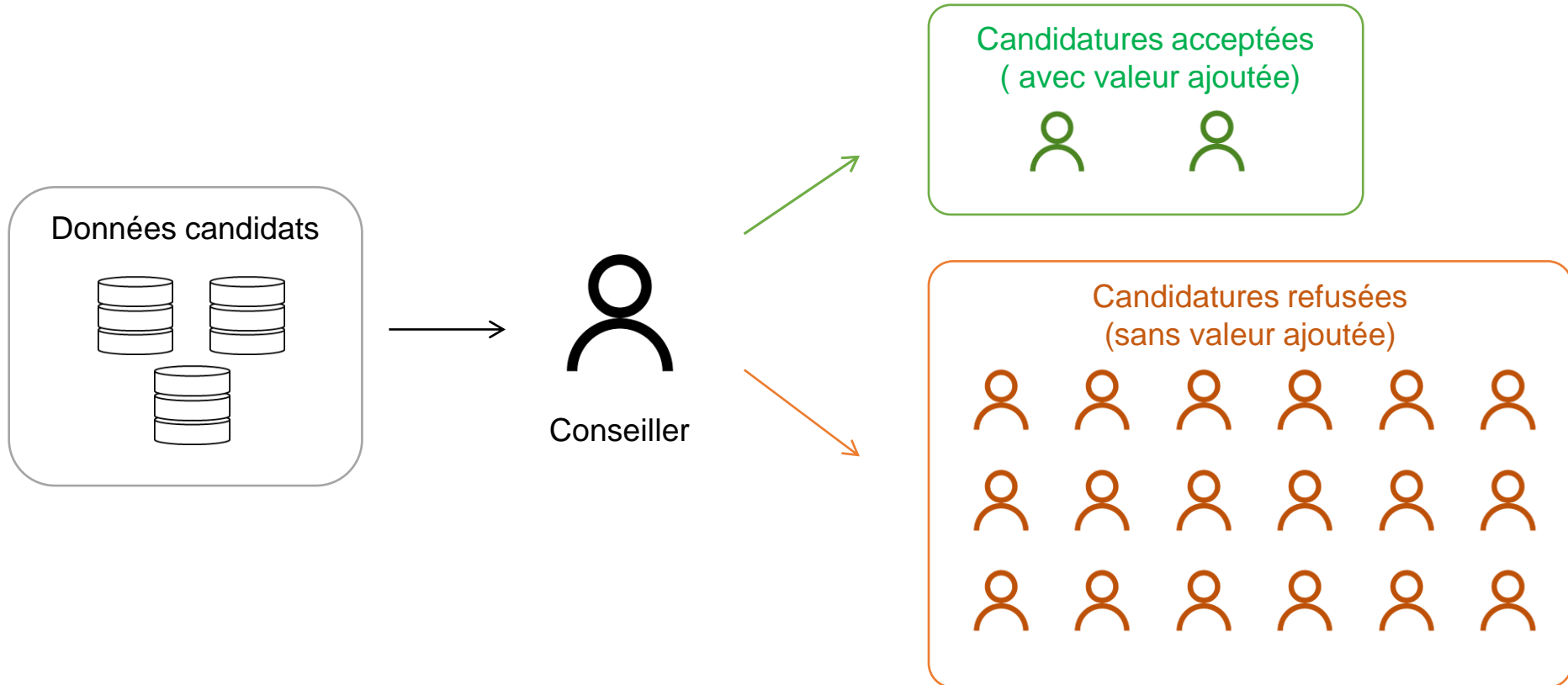
IV. Le dashboard

V. Bilan et perspectives

I. Introduction

A. L'entreprise

Organisme de crédit spécialisé dans l'**allocation de prêts**.



Candidatures refusées = travail **sans valeur ajoutée**.

→ fournir un algorithme qui **automatise la décision** et qui **l'explique** au conseiller, pour lui éviter d'étudier chaque candidature dans le détail.

I. Introduction

B. Le projet

Aider la décision des conseillers par l'intermédiaire :

- d'un **algorithme de prédiction**
- d'une **interface web** interactive expliquant la prédiction de l'algorithme



Données candidats



Algorithme de
prédiction



Interface web



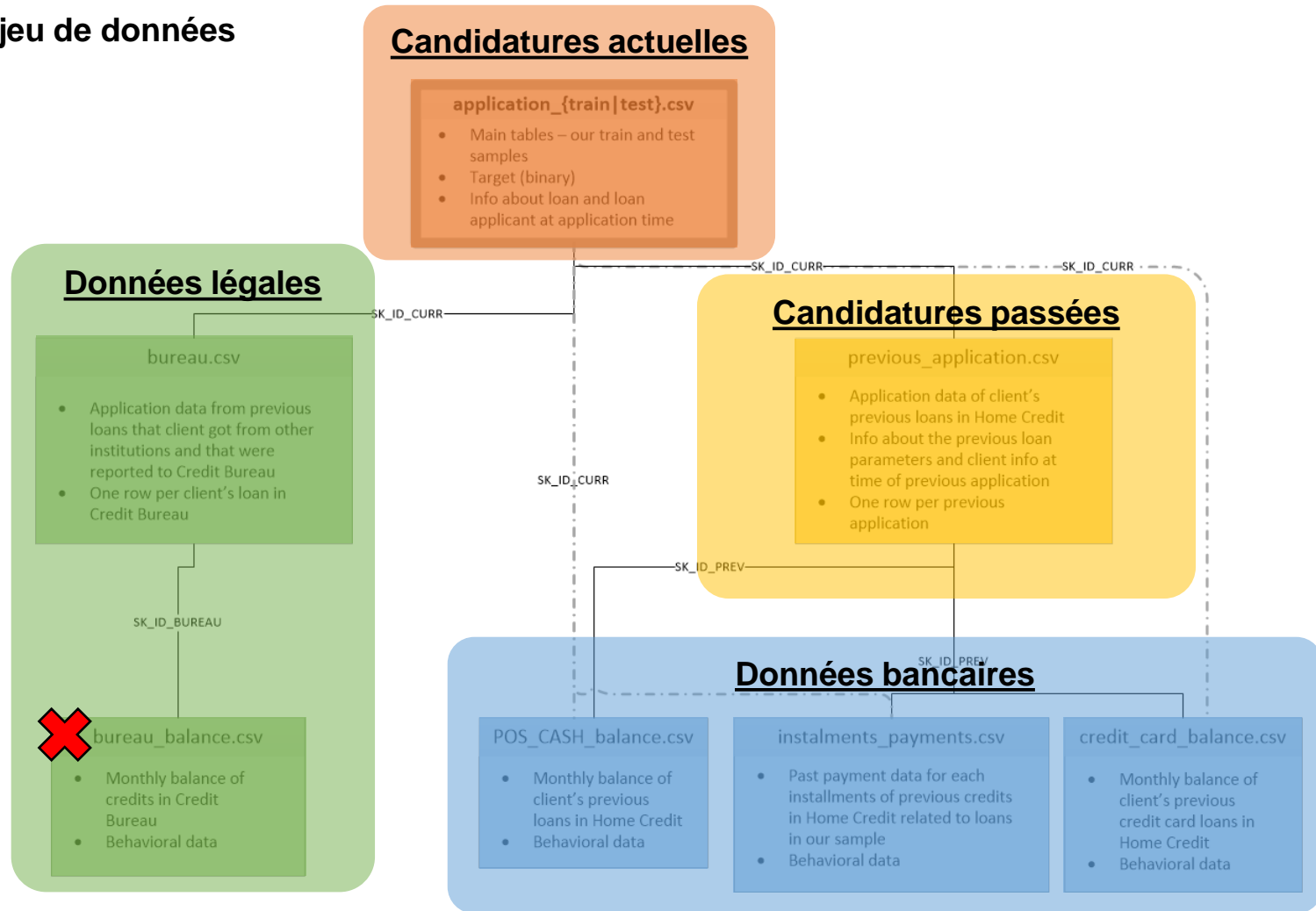
Conseiller

Candidature
acceptée

Candidature
refusée

I. Introduction

C. Le jeu de données



→ S'approprier les données par une bonne compréhension du métier du crédit
→ Définir les fusions utiles

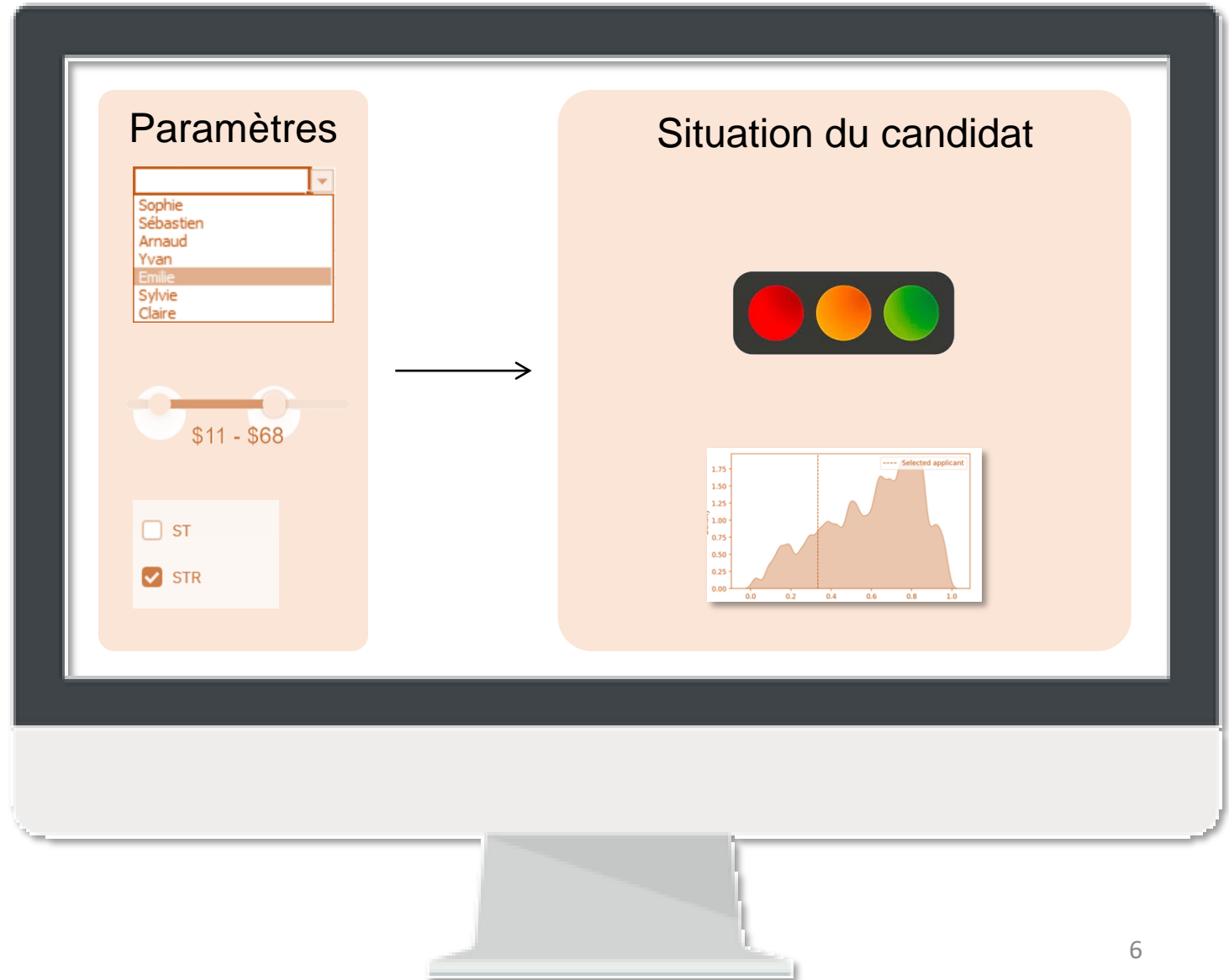
I. Introduction

D. Le livrable recherché : un dashboard

Synthétique

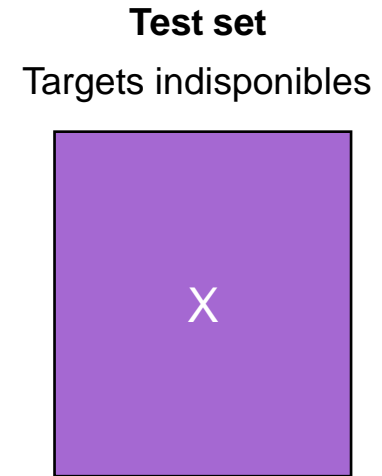
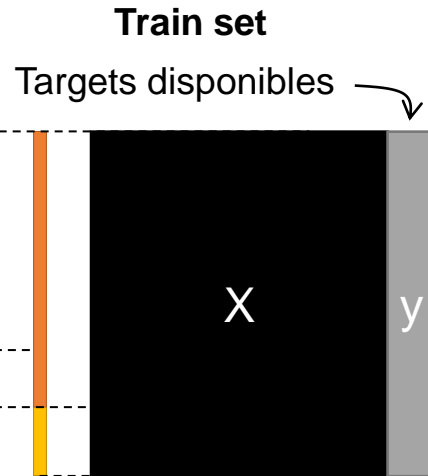
Aide à la
compréhension

Aide à la
décision



II. L'approche de modélisation

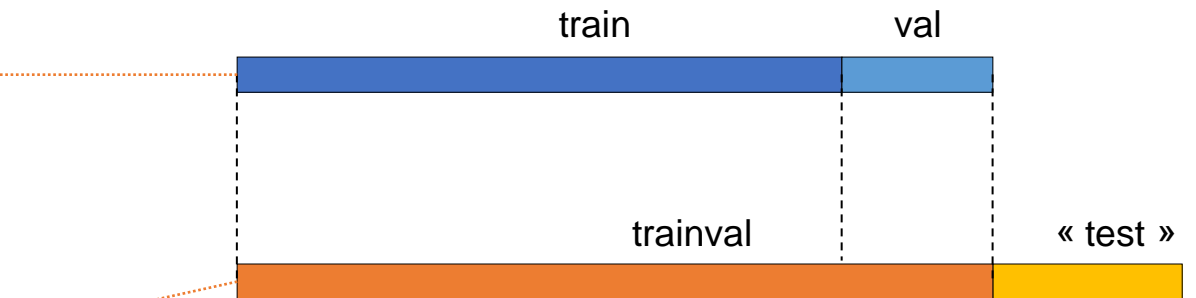
A. La démarche



1) Recherche des hyperparamètres

2) Comparaison des algorithmes

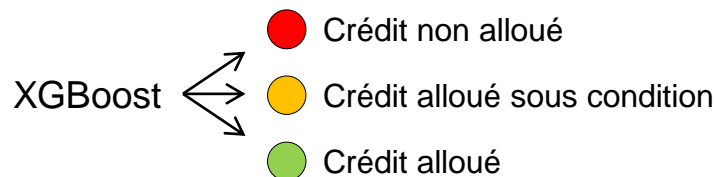
3) **Prédictions** sur le test set final



RandomForest

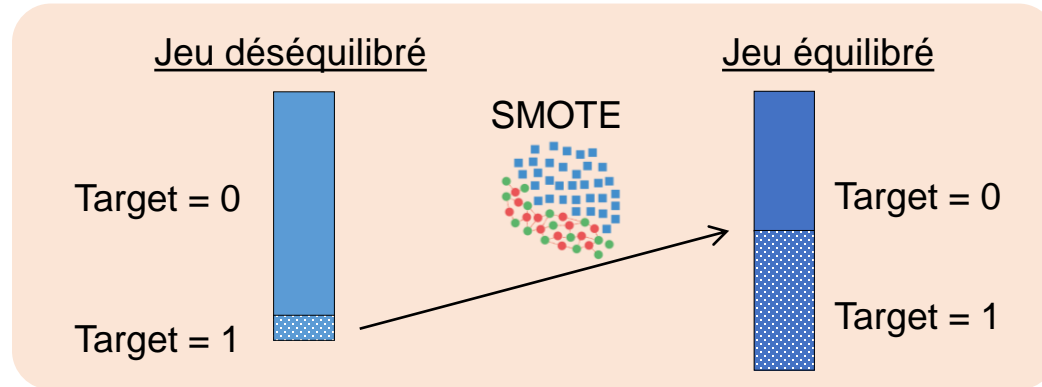
GradientBoosting

XGBoost



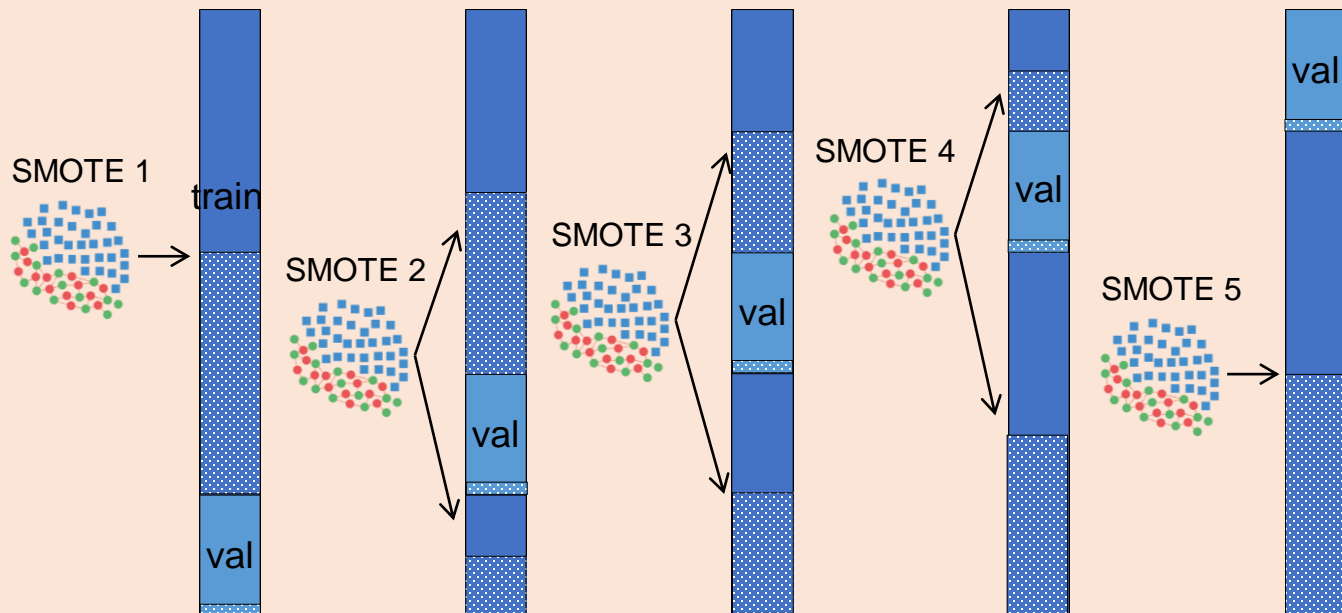
II. L'approche de modélisation

B. Jeu de données déséquilibré et SMOTE

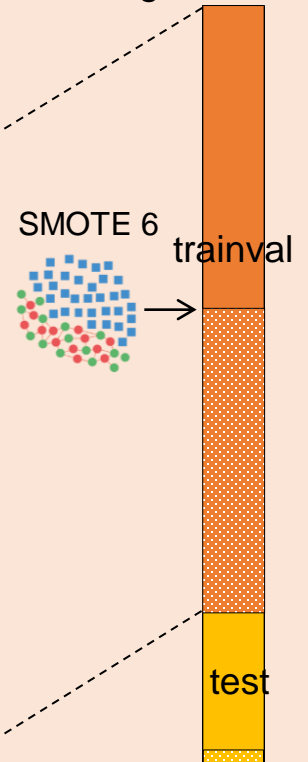


→ SMOTE ne s'applique pas aux *val* sets et au *test* set

1) Recherche des **hyperparamètres**

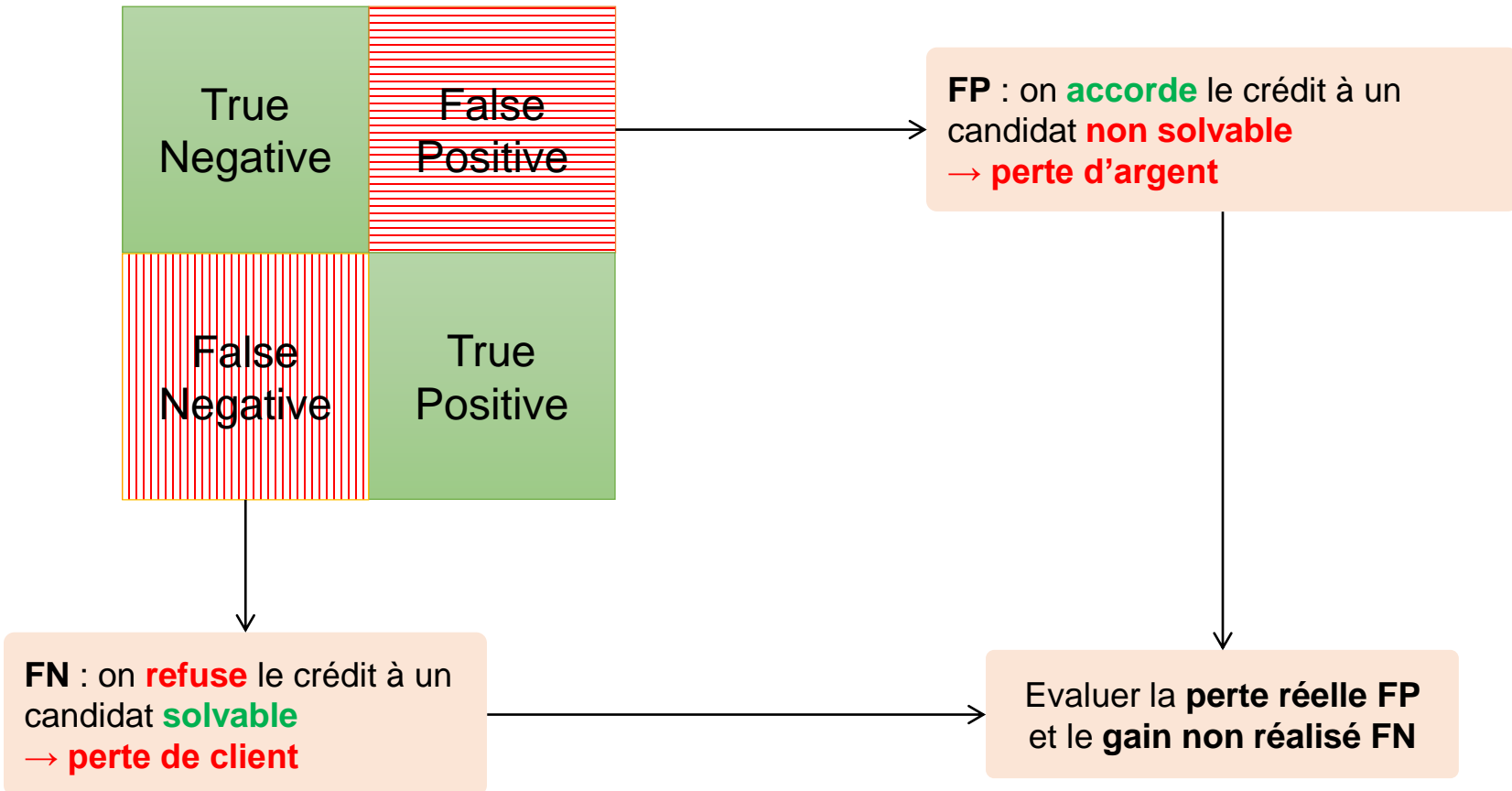


2) **Comparaison** des algorithmes



II. L'approche de modélisation

C. La fonction coût



- Meilleur compromis pour la fonction coût : **AMT_ANNUITY**, les intérêts annuels du candidat
- Cette fonction coût permettra d'évaluer les erreurs des algorithmes et d'en déduire le meilleur.

II. L'approche de modélisation

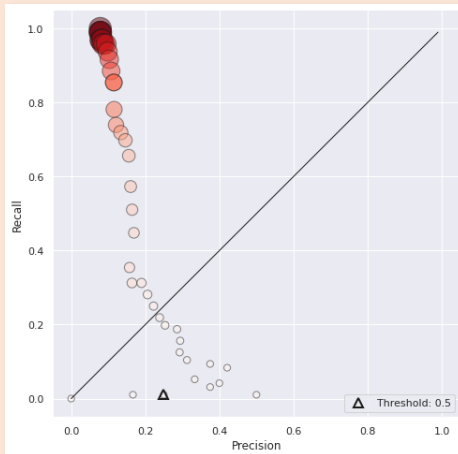
D. Choisir l'algorithme

→ sur 2% du jeu de données

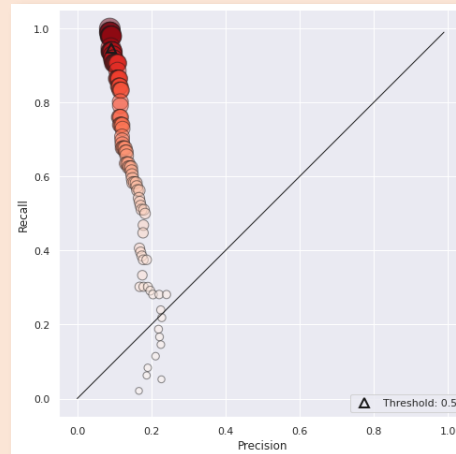
Courbes
précision /
rappel

● Coût élevé
○ Coût faible

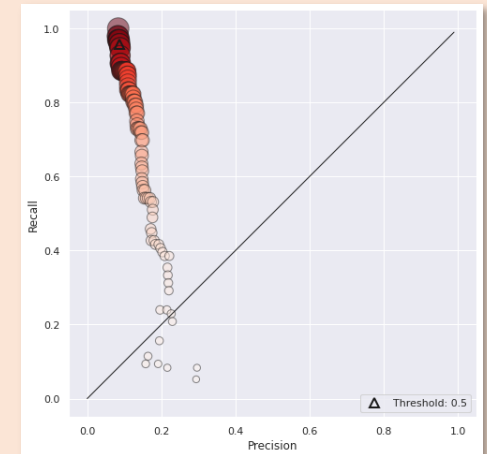
Random Forest



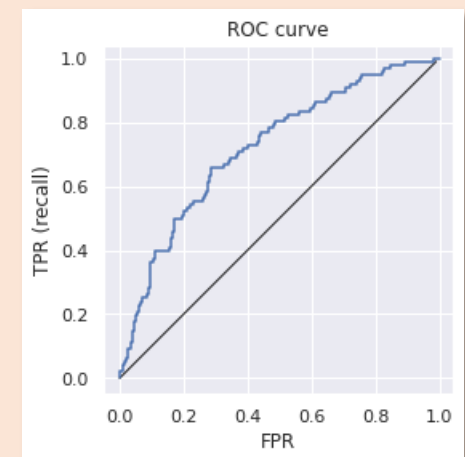
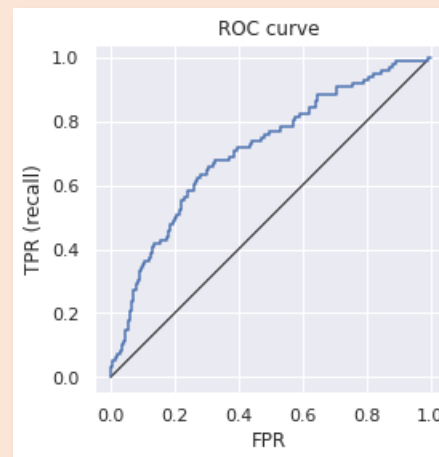
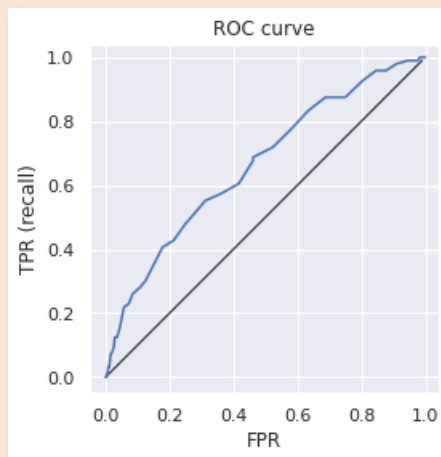
Gradient Boosting



XG Boost



Courbes
ROC

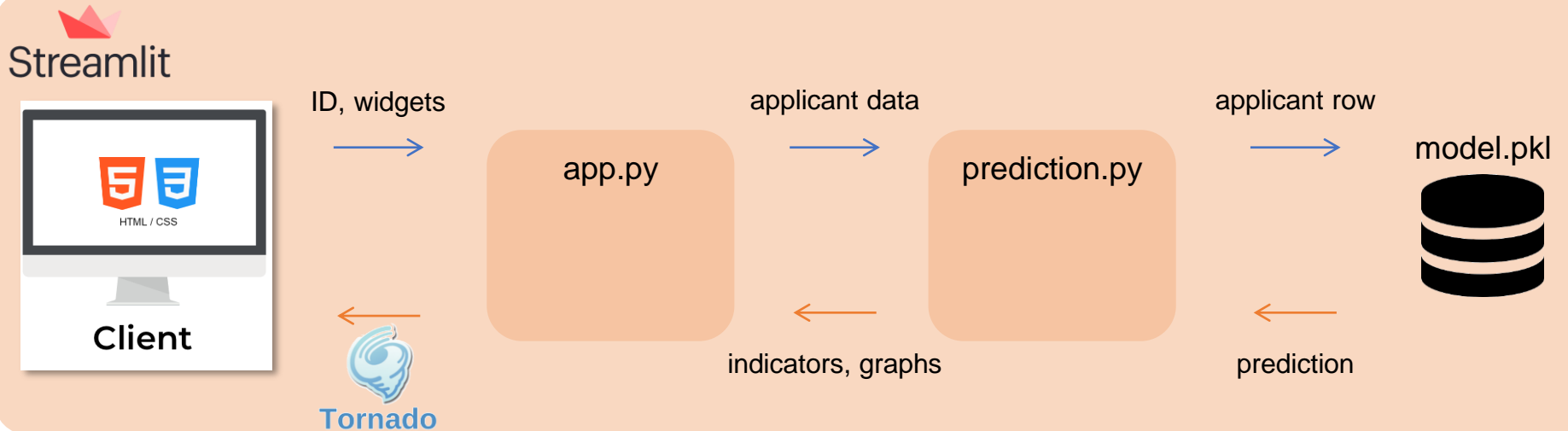
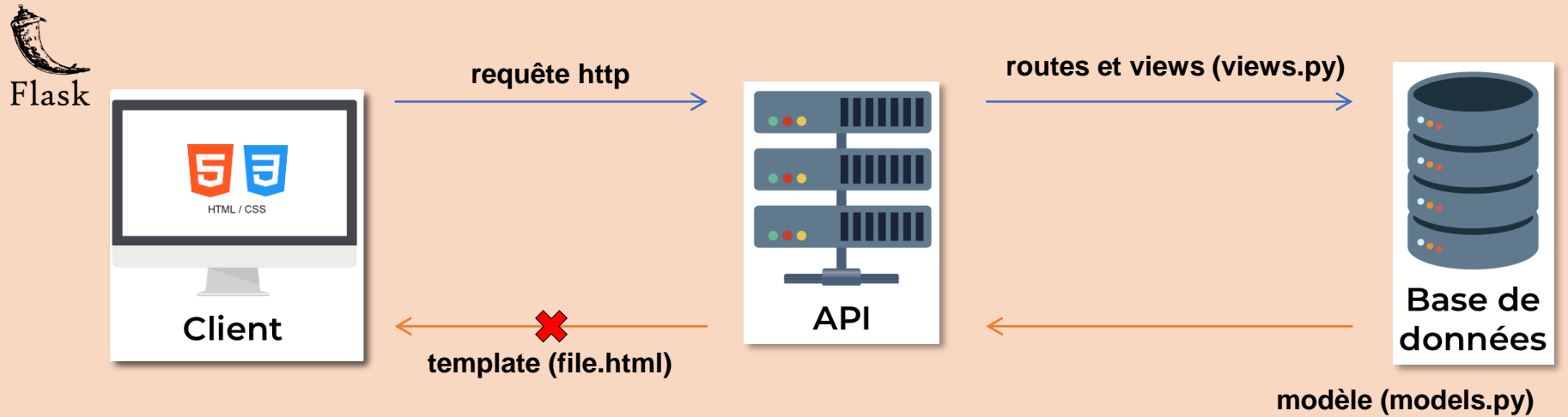


XG Boost donne le coût le moins élevé.

→ Gradient Boosting donne un coût similaire, mais XG Boost est bien plus rapide.

III. L'API

A. Organisation du projet

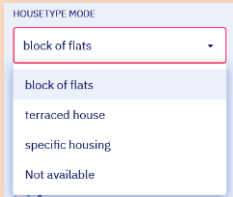


III. L'API

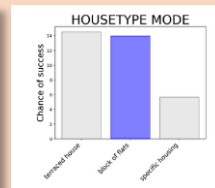
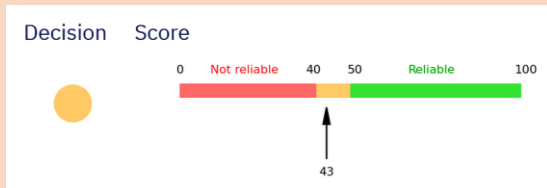
B. Les scripts

app.py

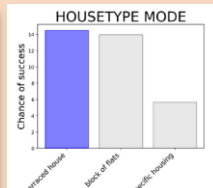
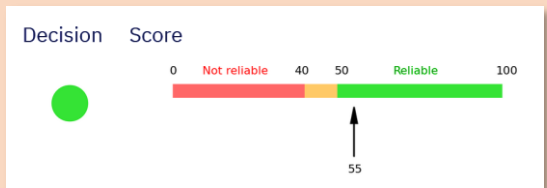
- Affichage des widgets



- Affichage des graphiques



- Gestion des modifications du conseiller



Applicant
data

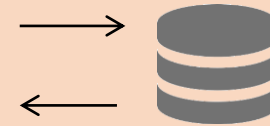


Indicators,
graphs

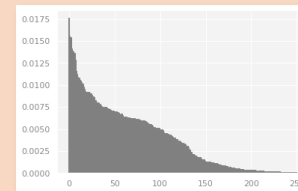


prediction.py

- Demande de prédiction au modèle entraîné

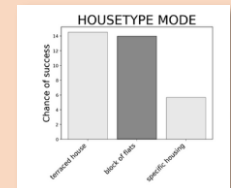
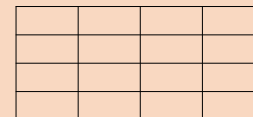


- Importances des caractéristiques



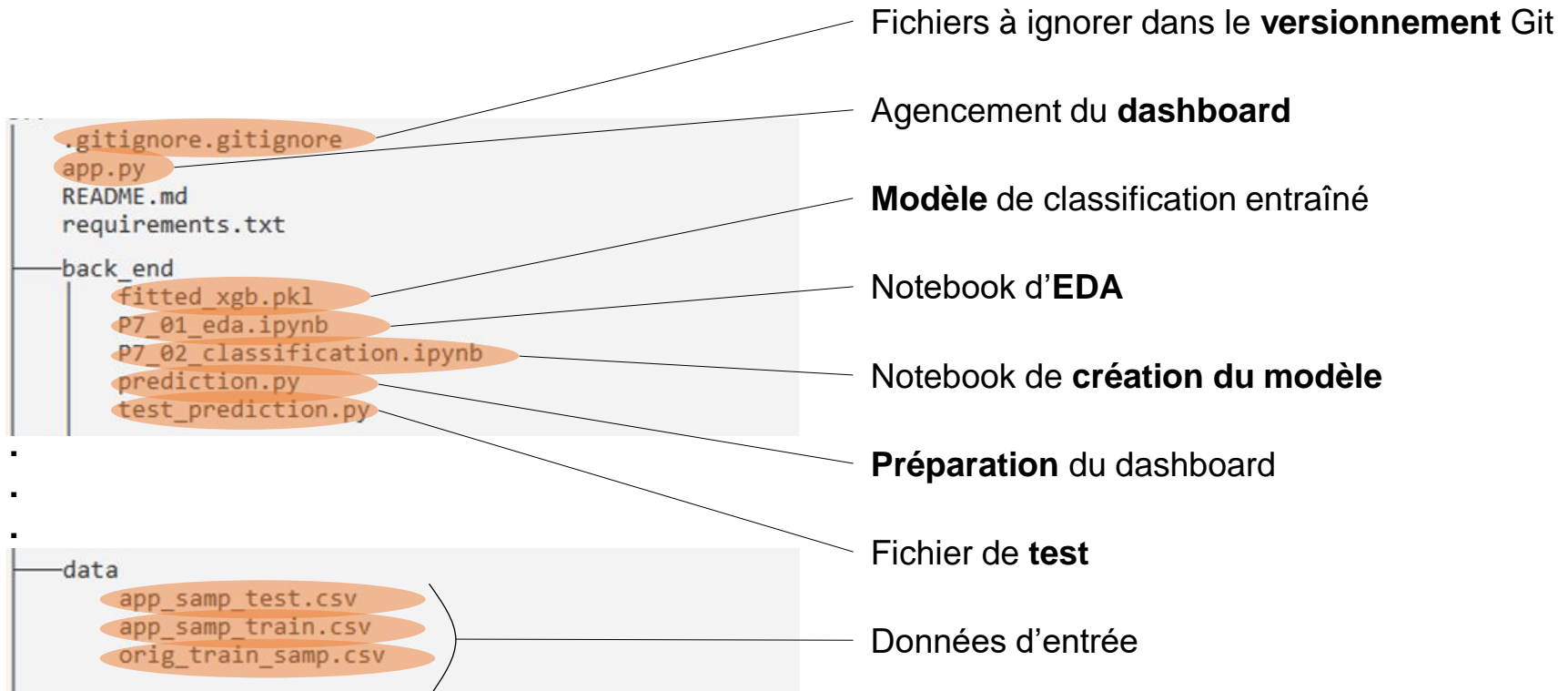
carac_1
carac_2
carac_3
carac_4
carac_5
carac_6

- Création des graphiques



III. L'API

C. Arborescence du projet



III. L'API

D. Le déploiement Web



**En local avec
Streamlit**



Dashboard fonctionnel



GitHub

Piste non aboutie

Difficultés à ouvrir le modèle entraîné (formats .pkl et .json)

Dernier bug :

Please compile with DMLC_USE_S3=1 to use S3

Lien : https://github.com/Benoit-78/credit_loan_scoring_model



heroku

Heroku

Piste non aboutie



Flask

Flask

Piste non explorée

Pas de fichier HTML / CSS

→ Déploiement en local avec Streamlit



III. L'API

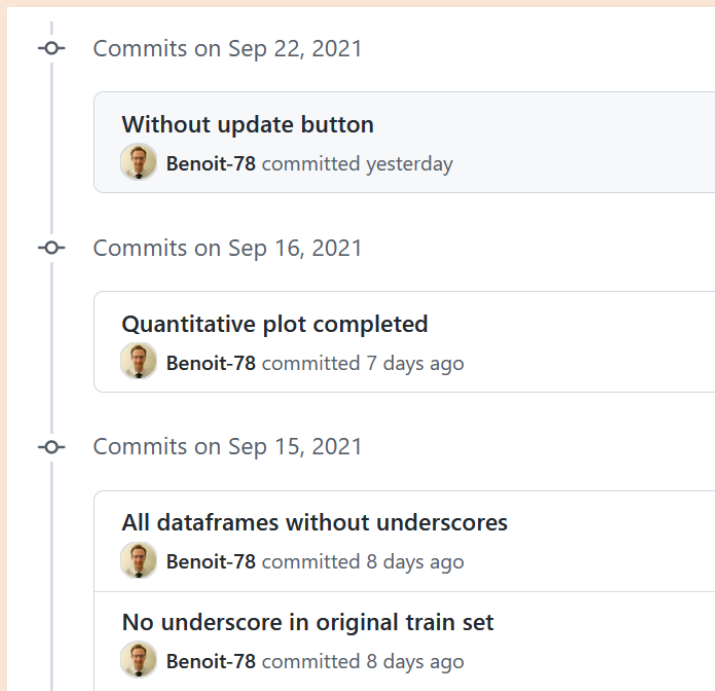
E. Le versionnement



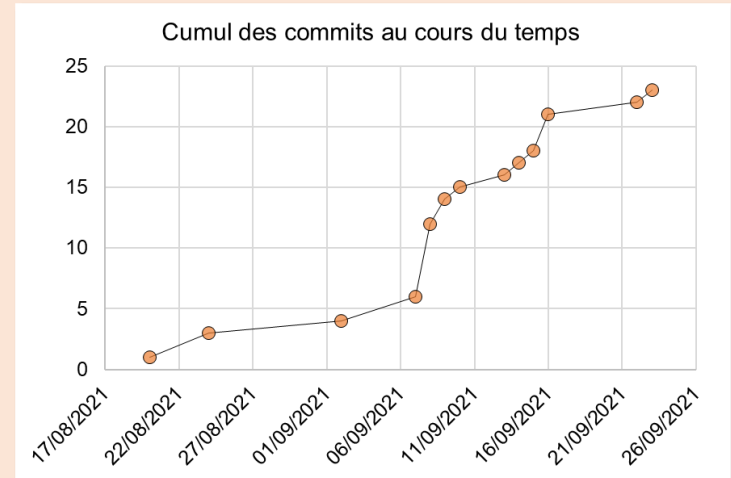
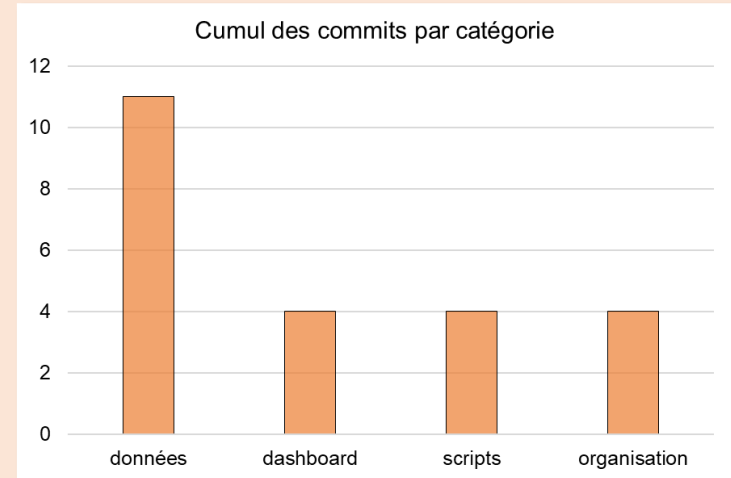
Git



GitHub



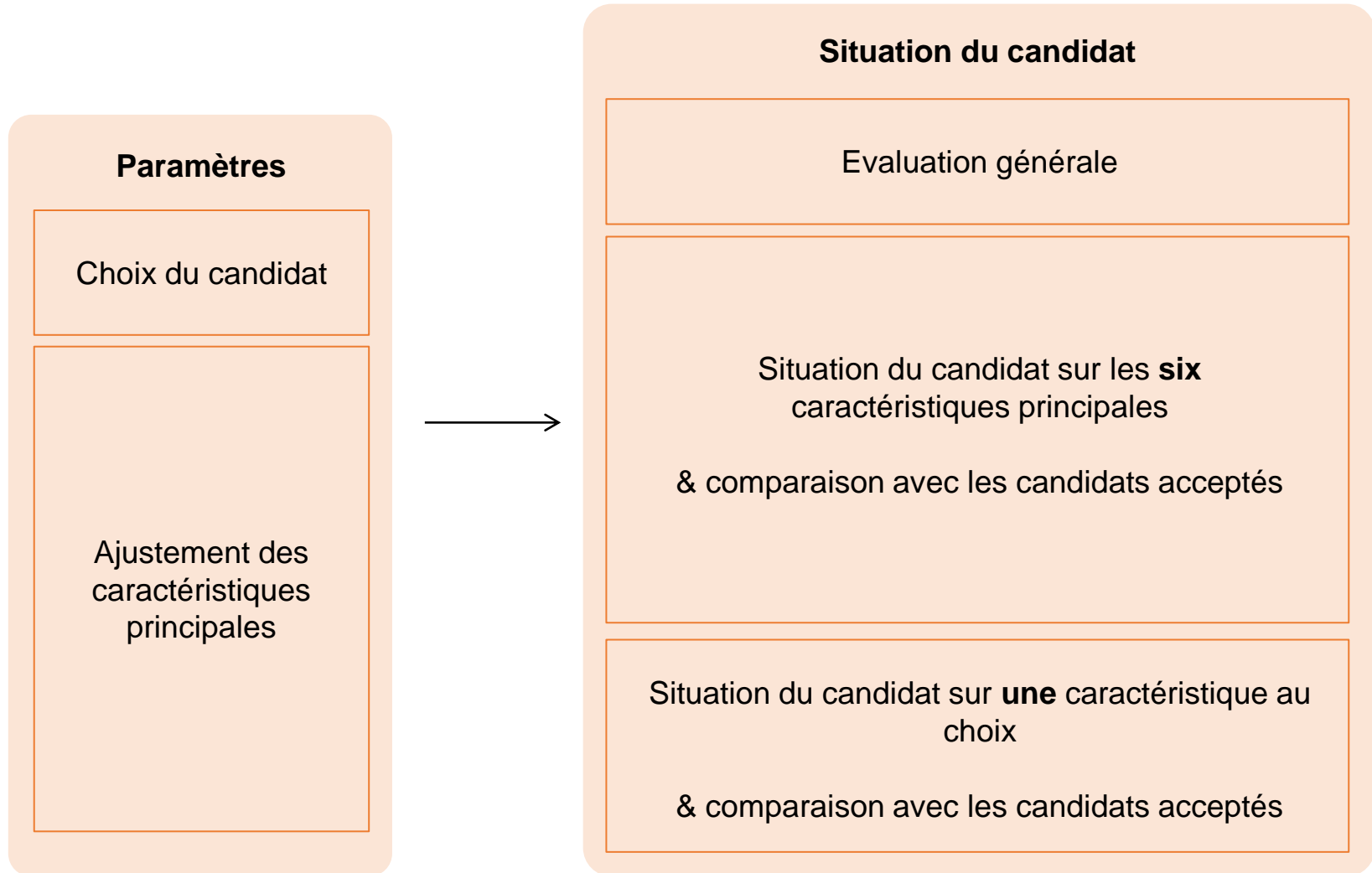
Analyse des commits



→ Certaines améliorations du dashboard auraient pu faire l'objet de branches pour limiter la taille de la branche principale

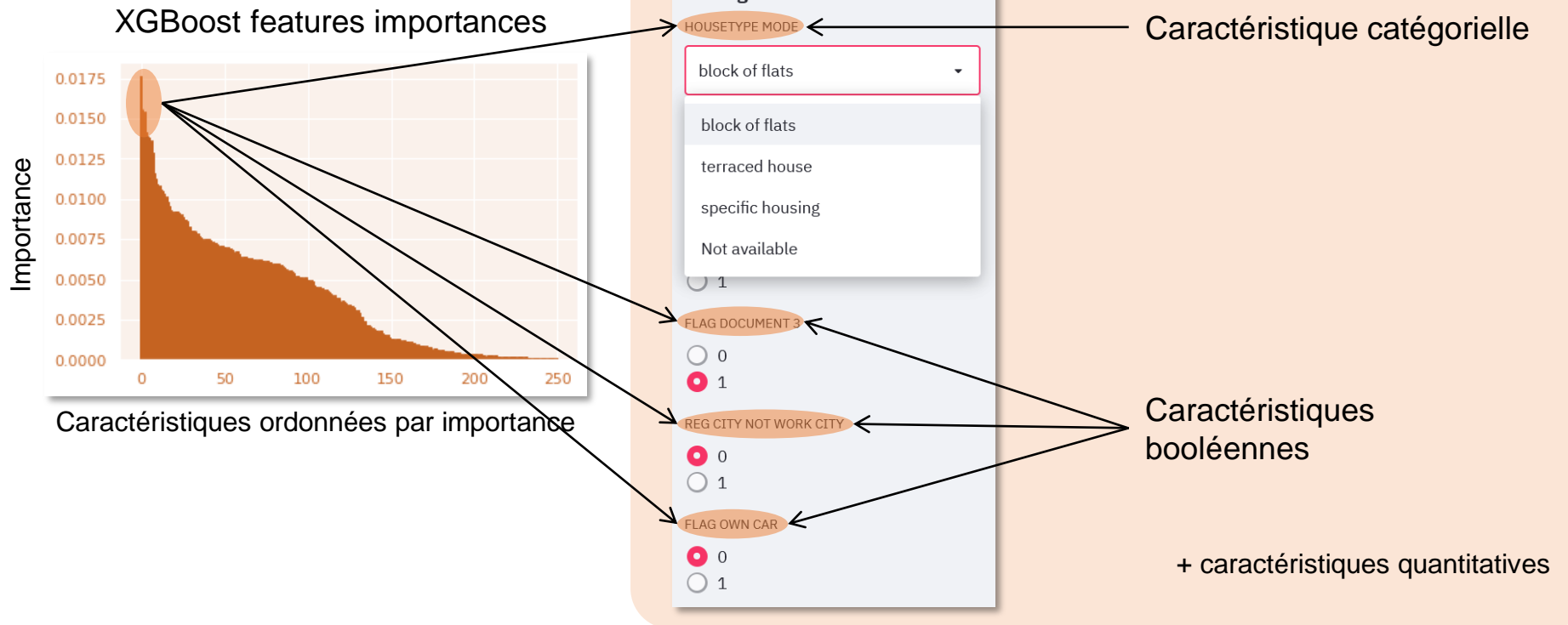
IV. Le dashboard

A. L'agencement général



IV. Le dashboard

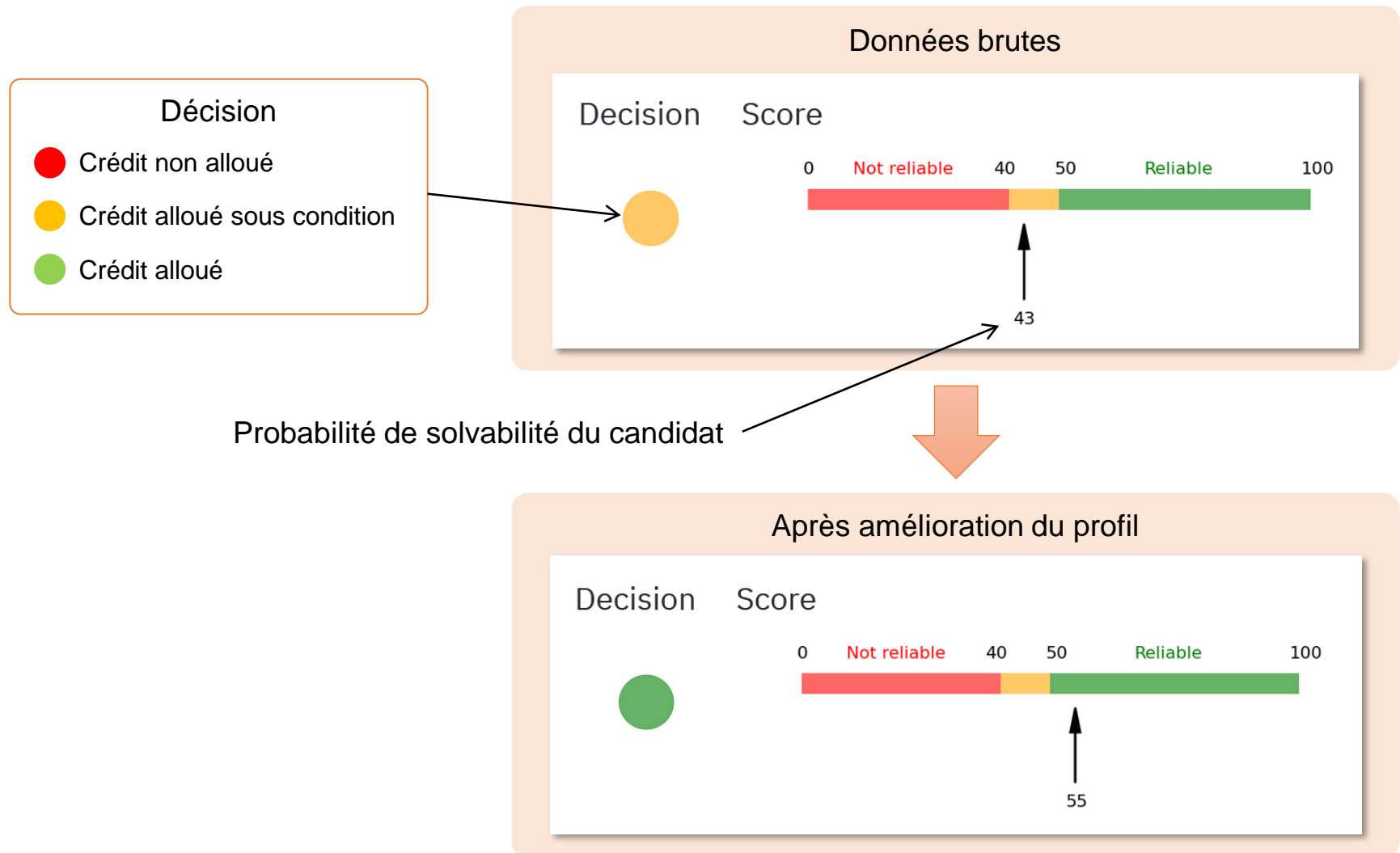
B. Les paramètres



→ Les caractéristiques d'un candidat peuvent ensuite être modifiées pour améliorer son profil de risque.

IV. Le dashboard

C. L'évaluation générale du candidat



→ Le conseiller peut jouer sur les paramètres pour améliorer en temps réel le profil du candidat

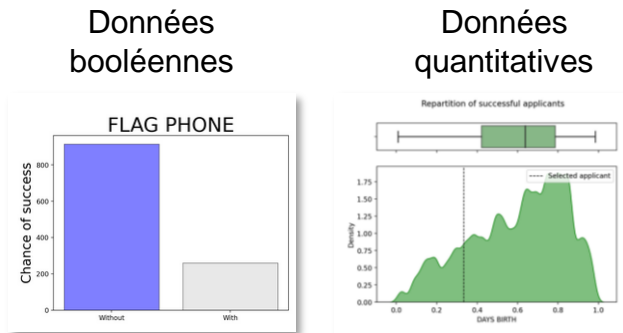
IV. Le dashboard

D. Les caractéristiques principales

- On affiche la **situation** du candidat parmi la **population** de candidats **acceptés**.
- À chaque type de caractéristique son propre graphique : catégorielle, booléenne ou quantitative.



Ajustement de caractéristiques

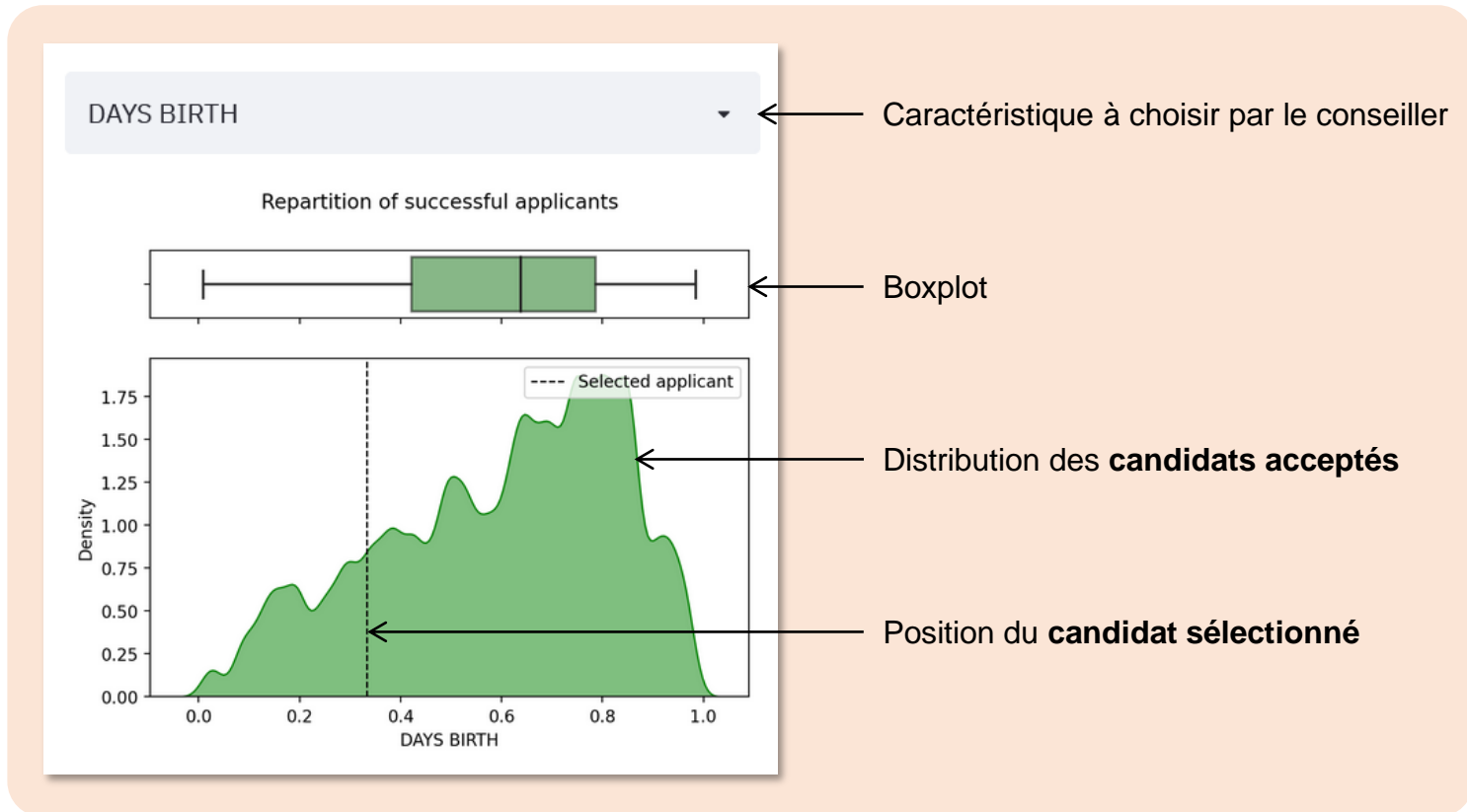


→ Les graphiques permettent de repérer rapidement les améliorations possibles

IV. Le dashboard






E. L'analyse au choix

Le conseiller peut choisir une caractéristique de son choix et y repérer la position du candidat. Exemple ci-dessous avec une caractéristique quantitative.



Ici, le candidat se trouve en dessous du quartile inférieur
→ candidat jeune par rapport aux candidats acceptés.

Bilan et perspectives

Sujet	Commentaire	
	Le jeu de données	<ul style="list-style-type: none">• Jeu de données important, certaines tables non exploitée par manque de puissance de calcul• SMOTE permet de rééquilibrer les classes accepté / refusé• Nécessité de s'intéresser au métier du crédit
	Le modèle	<ul style="list-style-type: none">• Fonction coût qui mérite d'être affinée avec plus de données• XGBoostClassifier l'emporte en précision et rapidité• On peut identifier des caractéristiques principales
	L'API	<ul style="list-style-type: none">• Fonctionnel en local avec Streamlit• Nombreuses pistes explorées sans succès pour le déploiement web (GitHub, Heroku, ...), mais sources d'apprentissage
	Le dashboard	<ul style="list-style-type: none">• Fonctionnel• Répond aux besoins d'un conseiller en crédit
	Perspectives	<ul style="list-style-type: none">• Augmenter la puissance de calcul pour traiter les tables de grande taille et affiner la fonction coût• Ajouter un commentaire général sous forme de texte• Placer les menus déroulants sous chaque graphique

Fin de la présentation



Merci pour votre attention