

# Implémenter un modèle de scoring

-

**Entreprise Prêt à Dépenser**

---

Open Classrooms parcours Data Science - projet 7

Camille Besançon

# PROBLÉMATIQUE

- Entreprise “Prêt à dépenser” : Propose des crédits à la consommation pour des personnes ayant peu ou pas d’historique de prêt.
- Objectif : Mettre en place un modèle de scoring évaluant la probabilité de défaut de paiement pour aider à la décision
- Plusieurs contraintes :
  - Besoin de transparence des clients sur les conditions d’octroi de prêts.
  - Conseillers et clients ne sont pas spécialisés en data science / ML
  - **Besoin d’un modèle performant dont les résultats sont présentés clairement**

**En plus de l’entrainement d’un modèle adapté, mise en place de dashboards interactifs détaillant les résultats.**

# OBJECTIF ET APPROCHE

**Objectif : Préparer un modèle qui évaluera la capacité du client à rembourser son crédit et présenter ces résultats sous une forme claire et pédagogique**

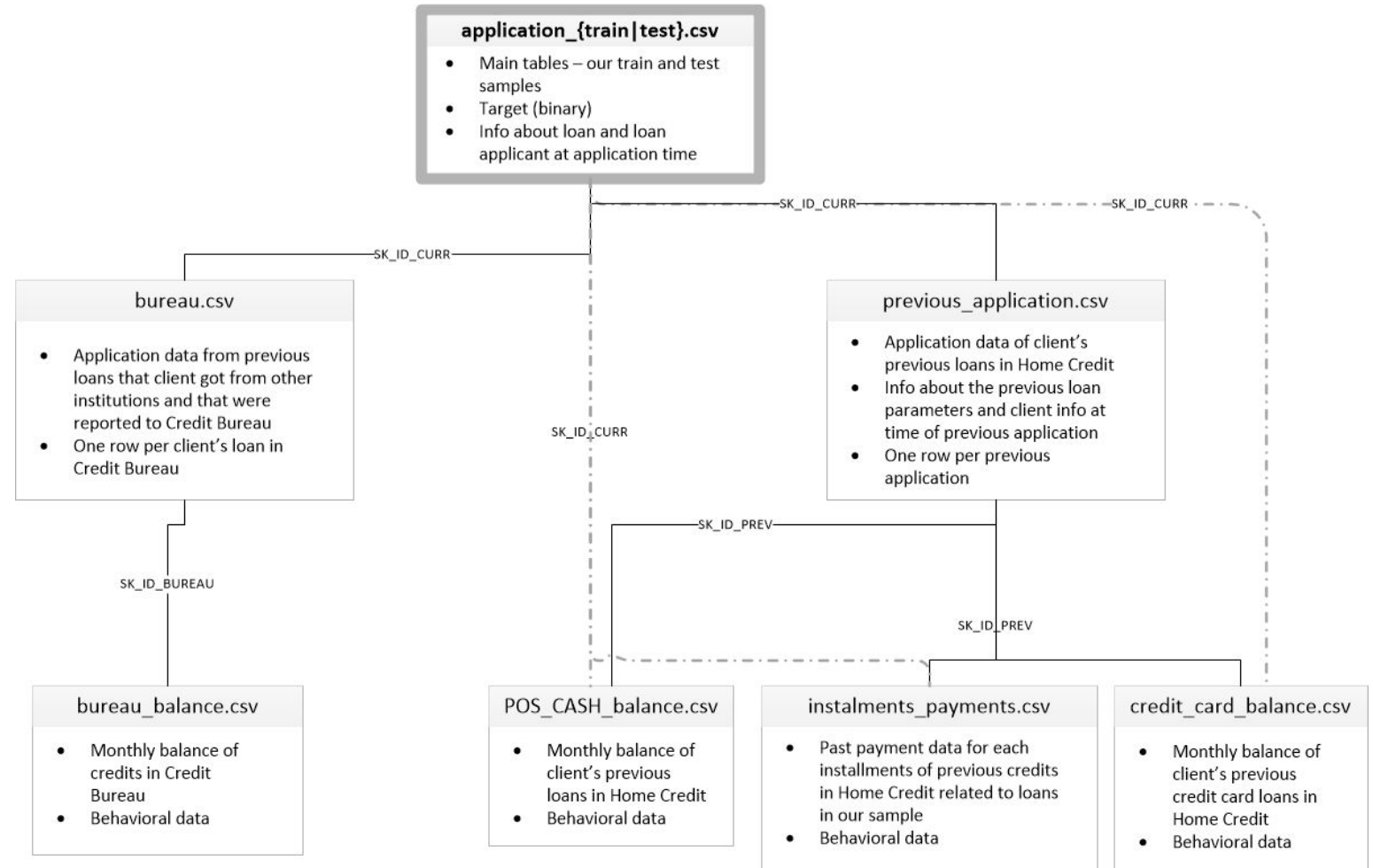
- Partie 1 : Construire un modèle de scoring qui évaluera la capacité d'un client à rembourser ou non son crédit
  - Pas d'historique de prêt : Données externes
  - Suivi des performances du modèle dans le temps
- Partie 2 : Construire un dashboard interactif destinés aux chargés de relation client
  - Interprétation des prédictions du modèle
  - Améliorer la connaissance des clients

# STRUCTURE DES DONNÉES

Plusieurs fichiers réunissant les informations sur le prêt / crédit demandé, les paiement précédents et les dossiers déposés précédemment (si applicable), informations sur le client lui-même, ...

Application\_train / application\_test : Fichiers principaux sur lesquels les autres données sont agrégées

**Cible : Variable TARGET**



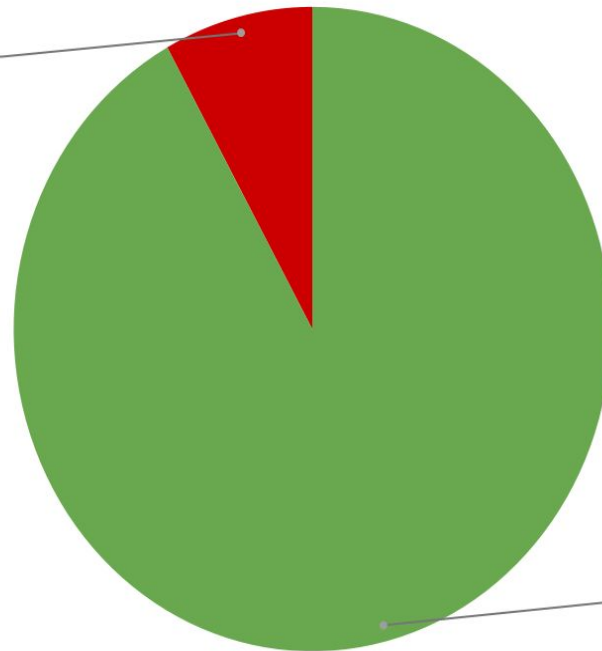
# STRUCTURE DES DONNÉES : CIBLE

- Clients classés en deux catégories :
  - 0 : 282 686 individus - Clients ayant remboursé leur crédit
  - 1 : 24 825 individus - Clients n'ayant pas remboursé leur crédit

Répartition des clients

1 (Mauvais client)

8,1%



0 (Bon client)

91,9%



**Les classes sont déséquilibrées :  
8% VS 92%**

**Risque de Surapprentissage du modèle :**

Modèle calibré uniquement sur les caractéristiques de la classe majoritaire 0 (bon clients) et n'aura pas vraiment "appris" des caractéristiques des mauvais clients

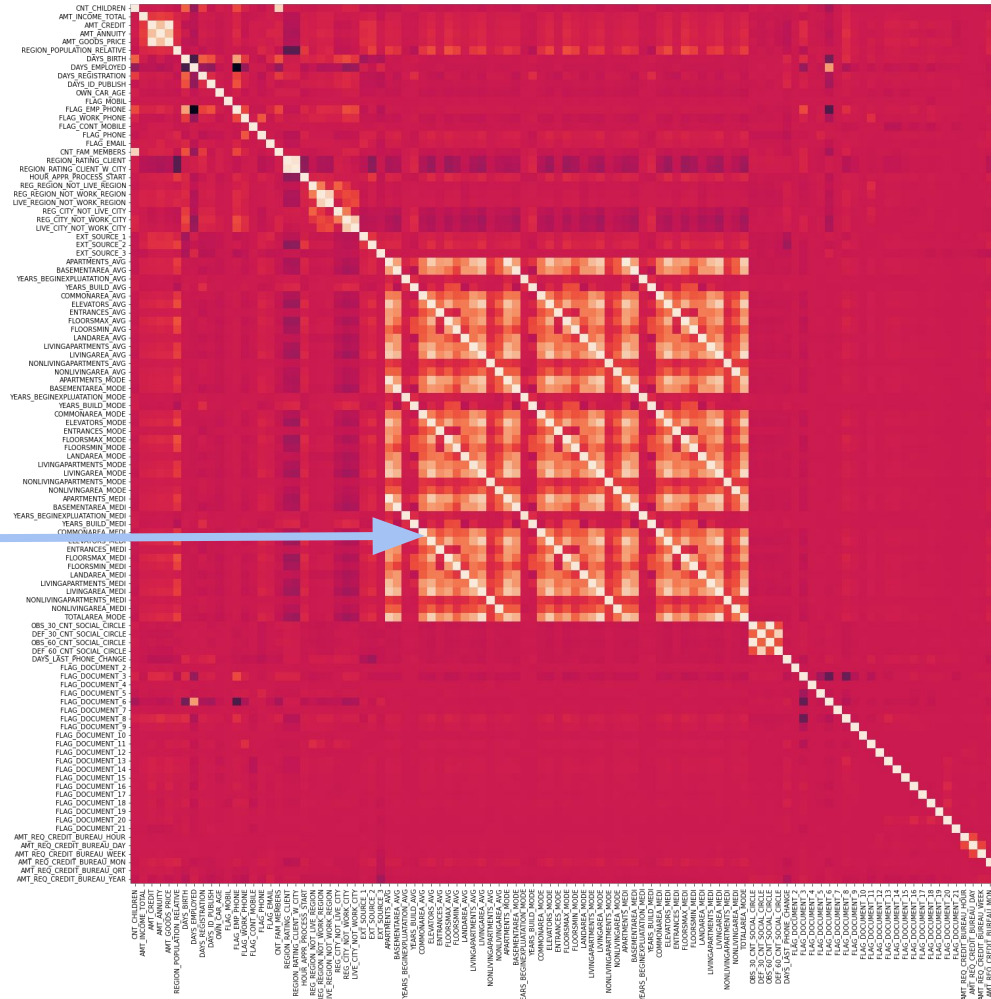
# SÉLECTION DES FEATURES IMPORTANTES

- Étape réalisée sur l'ensemble des données
- Mesure des corrélations :
  - Objectif : Sélectionner des variables indépendantes entre elles pour ne pas surestimer l'importance d'une variable
  - Sélectionner des variables qui sont indépendantes de la variable cible (catégorie du client)
- Le jeu de données initial comporte beaucoup de variables corrélées

# SÉLECTION DES FEATURES IMPORTANTES

Risque de surestimer  
l'importance de ces  
variables

Variables fortement  
corrélées



1 (Correlation)

0 (pas de corrélation)

-1 (Correlation)

# SÉLECTION DES FEATURES IMPORTANTES

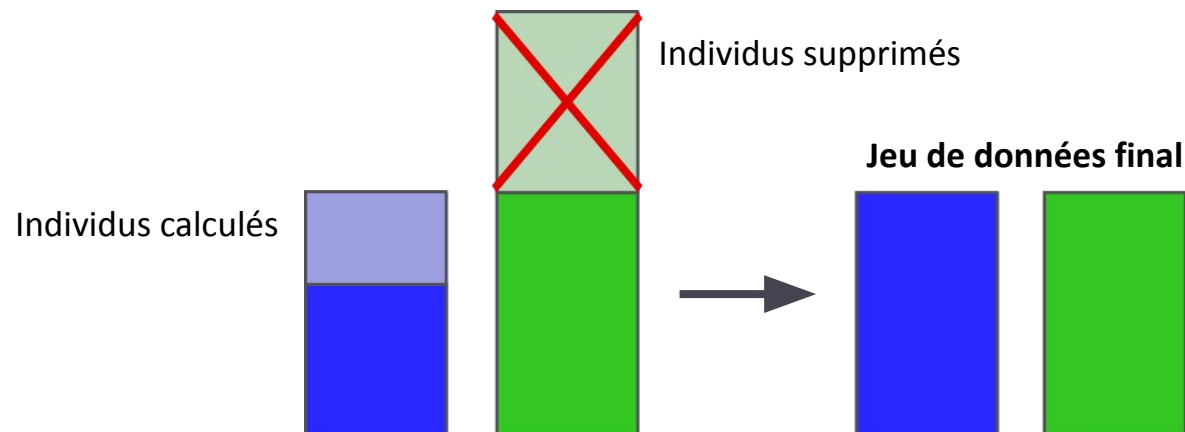
- Étape réalisée sur l'ensemble des données
- Mesure des corrélations :
  - Objectif : Sélectionner des variables indépendantes entre elles pour ne pas surestimer l'importance d'une variable
  - Sélectionner des variables qui sont indépendantes de la variable cible (catégorie du client)
- Le jeu de données initial comporte beaucoup de variables corrélées
  - Risque de surestimer l'importance de ces variables dans le modèle
  - On supprime donc une partie de ces variables
- Types de données mixtes : Variables continues et catégorielles
  - Encodage des variables catégorielles

Catégorie	Rouge	Bleu	Vert
[ Rouge ]	1	0	0
[ Bleu ]	0	1	0
[ Rouge, Vert ]	1	0	1



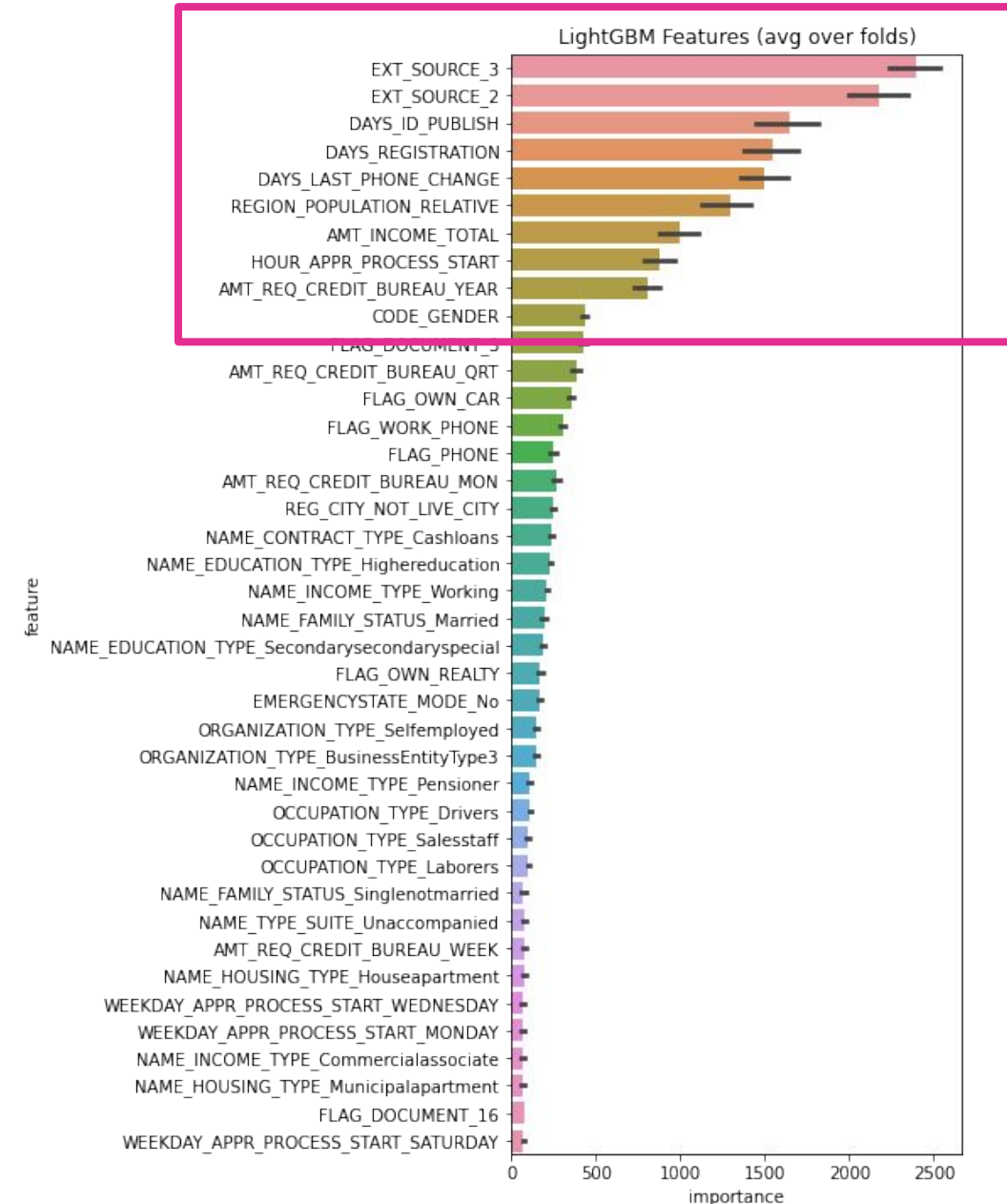
# ÉQUILIBRAGE DES DONNÉES

- Jeu de données déséquilibré : Méthode SMOTE-NC
  - Création de nouveaux individus **à partir des caractéristiques des individus réels**
  - Attention : Créer trop d'individus = Majorité de données artificielles !
- Classe minoritaire : De 24 825 à 28 268 individus.
  - Calcul de 3 443 nouveaux individus ( ~ 14% de la classe minoritaire, 10% de la classe majoritaire)
- Suppression aléatoire des individus excédentaires dans la classe majoritaire
  - Ajouter des individus dans la classe minoritaire = limiter la suppression d'individus de la classe majoritaire et de conserver un maximum de données
- Données finales : 56 536 individus



# SÉLECTION DES FEATURES

- Entraînement d'un premier modèle utilisant toutes les variables
  - Identification des variables les plus pertinentes
  - Sélection des 10 premières variables ayant la plus forte contribution au modèle
- A partir de ces variables :
  - Entraînement de différents types de modèles et optimisation des hyper-paramètres
  - "Seulement" une dizaine de variables = gain de temps, moins d'informations à collecter



# ENTRAÎNEMENT DE MODÈLES

- Méthode :

- Entraînement de 6 modèles sur les mêmes données
- Evaluation des performances : Temps d'entraînement, AUC ou RMSE, Score métier

**AUC :**

Evalue la capacité du modèle à distinguer les classes à prédire.

0,5 = Classification aléatoire. 1 = Classement "parfait"

**RMSE :**

Basé sur la moyenne des écarts entre les valeurs prédites par le modèle et les valeurs réelles. Plus le RMSE est faible plus les prédictions sont proches des valeurs réelles.

- Modèles testés :

- Deux modèles de régression (prédiction sous forme de variable continue)
- Quatre modèles de classification (Prédiction sous forme de variable catégorielle)

- Score métier : Evaluer la capacité du modèle à éviter les faux négatifs

- Clients jugés "bons" qui sont en fait de mauvais clients : Risque de perte de capital

$$\text{Score métier} = ( 10 \times \text{FN} + \text{FP} ) / \text{total de clients}$$

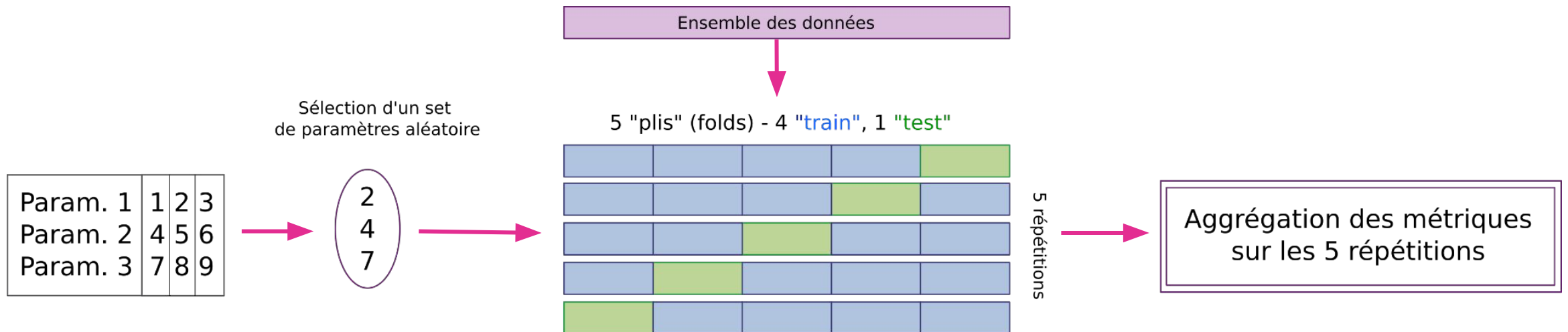
FN = Faux négatifs. FP = Faux positifs (bons clients refusés, perte d'opportunités / de clients pour l'entreprise)

On juge les FN plus grave car ils entraînent une perte sèche de capital

**Plus ce score est bas, plus le modèle est efficace**

# ENTRAÎNEMENT DE MODÈLES

- Entraînement des modèles :
  - Basé sur les 10 variables sélectionnées.
  - On veut éviter le sur-aprentissage ( "overfitting" )
  - On veut optimiser les paramètres pour avoir le meilleur résultat possible
- Méthode basée la technique du "k-fold" et une grille de paramètres
  - maximisation de l'utilisation des données (utilisation en validation et en test)
  - Réduction de la variabilité et des risques d'overfitting



# ENTRAÎNEMENT DE MODÈLES

- Modèles testés :
  - Deux modèles de régression (prédiction sous forme de variable continue)
  - Quatre modèles de classification (Prédiction sous forme de variable catégorielle)
- Tracking des performances avec MLFlow

~~Régression - Régression linéaire~~

~~Régression - ElasticNet~~

~~Classification - Random Forest~~

Classification - Régression logistique

Classification - K-Neighbors

Classification - SGD

Duration	Source	Models	AUC	best_cv_score	rmse	score metier
4.6s	ipykern...	sk-learn-R.../8, 2 more	-	-3.526	0.625	3.573
7.3s	ipykern...	sk-learn-R.../8, 2 more	-	-4.725	0.693	4.751
7.9min	ipykern...	sk-learn-C.../8, 2 more	0.728	-1.561	-	1.575
12.0s	ipykern...	sk-learn-C.../14, 2 more	0.686	-1.731	-	1.738
3.1min	ipykern...	sk-learn-C.../8, 2 more	0.699	-1.665	-	1.658
13.3s	ipykern...	sk-learn-C.../12, 2 more	0.67	-1.713	-	2.238

# SUIVI DU MODÈLE


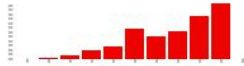

















- Risque de dégradation des performances du modèle : Data drift
  - Phénomène où les données réelles des clients s'éloignent des données d'entraînement
  - Le modèle devient moins performant (basé sur des données obsolètes)
  - Exemple : En cas de crise financière, l'importance des revenus des clients changera
- Données "application\_test.csv"
  - Données plus récentes et sans "TARGET" (variable cible) - pas utilisées pour l'entraînement
  - Utilisées pour l'analyse de data drift
- Librairie Evidently :
  - Librairie / plateforme d'outils pour l'évaluation et le suivi de modèles
  - Création d'un dashboard d'analyse de data drifting...

# SUIVI DU MODÈLE

- Data drifting détecté sur 60% des variables (6/10)

**Il faudra prévoir d'actualiser / réentraîner régulièrement le modèle avec des données plus récentes.**

**Mise en place d'un système d'alerte basée sur l'analyse du data drift**

Column	Type	Reference Distribution	Current Distribution	Data Drift
DAYS_LAST_PHONE_CHANGE	num			Detected
EXT_SOURCE_2	num			Detected
EXT_SOURCE_3	num			Detected
AMT_INCOME_TOTAL	num			Detected
DAYS_ID_PUBLISH	num			Detected
AMT_REQ_CREDIT_BUREAU_YEAR	num			Detected
REGION_POPULATION_RELATIVE	num			Not Detected
DAYS_REGISTRATION	num			Not Detected
CODE_GENDER	num			Not Detected
HOURL_APPR_PROCESS_START	num			Not Detected



# MISE EN PRODUCTION

- Dépôt sur GitHub des scripts et du modèle
  - Plateforme d'hébergement basée sur Git, un logiciel de gestion de version gratuit
  - Centralisation du projet : Facilité de collaboration, suivi des versions et partage des fichiers
- Clonage du répertoire sur une instance EC2 d'Amazon Web Services (AWS)
  - Elastic Compute Cloud - Service de location de serveurs pour déploiement d'applications
  - Permet d'accéder au code et de mettre à jour facilement le code ou le modèle

The screenshot shows a GitHub repository for 'Ubuntu' with the following components:

- File List:** A table of files and their commit history.
- Activity:** A list of recent commits with a pink arrow pointing to the 'Activity' link in the repository sidebar.
- Repository Sidebar:** Contains links to 'About', 'Readme', 'Activity', 'Releases', 'Packages', and 'Languages'.

File	Description	Commit Date
__pycache__	added tracking of inputs and predictions	5 days ago
final_data	added tracking of inputs and predictions	5 days ago
pipeline	updates on dashboard	last month
venv	added tracking of inputs and predictions	5 days ago
README.md	Update README.md	last month
app.py	mods	5 days ago
dashboard.py	added tracking of inputs and predictions	5 days ago
deploy.py	mods	5 days ago
deploy.yml	mods	5 days ago
essai.py	mods	5 days ago
my_dashboard.conf	config files	2 weeks ago
projet7.ipynb	Added notebooks and pipeline pickle	2 months ago
projet7_MLFlow.ipynb	Added notebooks and pipeline pickle	2 months ago
requirements.txt	fixed packages versions	2 weeks ago
rerun.sh	mods	5 days ago
test_unit.py	mods	5 days ago

**Activity Feed:**

- added tracking of inputs and predictions**  
camillesancon pushed 1 commit to main • ecf84b2...82f7ab2 • 4 days ago
- prettify the code**  
camillesancon pushed 1 commit to main • 9a34e58...ecf84b2 • 4 days ago
- mods**  
camillesancon pushed 1 commit to main • cbe4d95...9a34e58 • 4 days ago
- correctifs**  
camillesancon pushed 1 commit to main • 5c9ca0d...cbe4d95 • 5 days ago
- code cleaning**  
camillesancon pushed 1 commit to main • ea507bd...5c9ca0d • 6 days ago



# MISE EN PRODUCTION

- Script de déploiement et tests unitaires
  - Script de déploiement : Importe les packages nécessaires à l'exécution des scripts à partir d'un fichier spécifiant les packages et leur version
  - Lancement des scripts de l'API et du dashboard
  - Exécute ensuite des tests unitaires pour vérifier le fonctionnement correct de l'API
- Tests unitaires :
  - Pratique de développement pour tester individuellement le fonctionnement des éléments d'un logiciel (ici, une application)
  - Tests réalisés : API joignable, validité des résultats pour des données normales en entrée, ...
  - ... Si les données entrées sont invalides (réponse manquante), vérification que l'API renvoie une erreur
- Mise à disposition *via* un dashboard interactif...

# DASHBOARD INTERACTIF

- Accessible depuis n'importe quel navigateur
  - Ouverture des requêtes html de l'instance EC2
  - Facilité d'utilisation car pas de connexion directe à l'instance
- Démo...
- Caractéristiques du dashboard interactif :
  - Permet d'entrer les informations du client et de visualiser où il se situe pour chaque variable par rapport aux deux populations du jeu d'entraînement (bon client / mauvaise client)
  - Permet de lancer la prédiction (requête à l'API) et affiche les résultats sous forme d'un "compteur" affichant la probabilité estimée que le client rembourse l'emprunt
  - Accessibilité : Informations visuelles dédoublées avec du texte et/ou graphiques en couleurs adaptées aux personnes daltoniennes, motifs des lignes différents

# Implémenter un modèle de scoring

-

Entreprise Prêt à Dépenser

**Merci pour votre attention**

---

Open Classrooms parcours Data Science - projet 6

Camille Besançon