

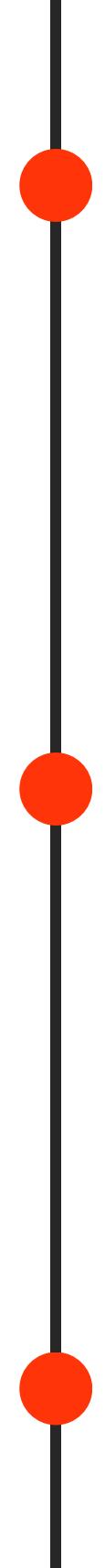
# IMPLÉMENTER UN MODÈLE DE SCORING

CORENTIN LAFI

Août 2024



# La mission



Créer un modèle de classification pour une entreprise de crédits à la consommation pour **identifier les bons et mauvais clients** à partir de leurs données disponibles

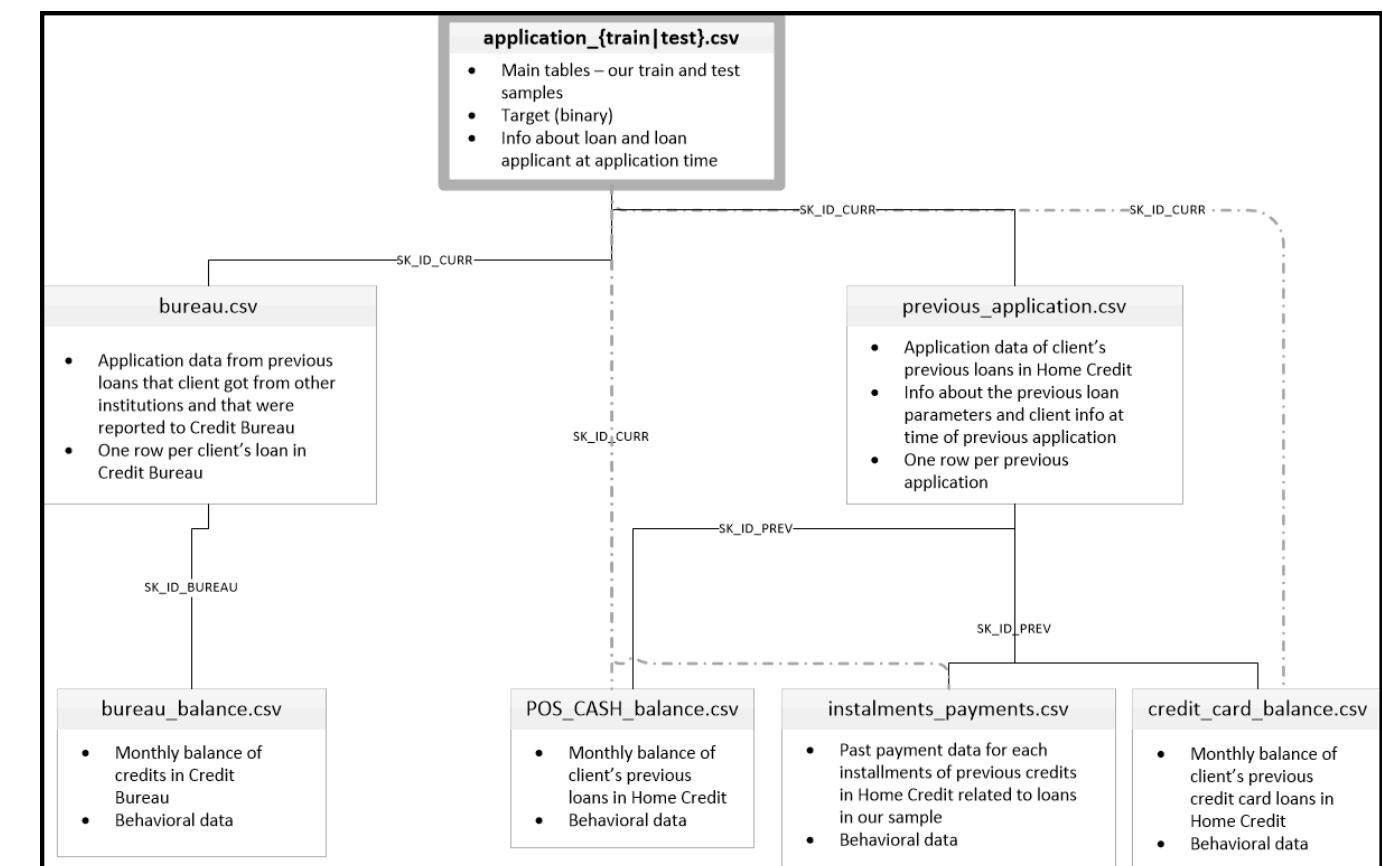
Mettre à disposition du métier le modèle grâce à une **API**

Mettre en place des process automatisés de gestion du **cycle de vie du modèle** : gestion des versions, déploiement sur le cloud, suivi des performances (métriques), surveillance du data drift

# Les jeux de données

---

- 9 jeux de données au format .csv
  - informations sur les **antécédents bancaires** du client
  - informations **actuelles** sur le client
  - descriptions des colonnes
- Table principale : **application\_train.csv**
  - Contient la variable cible
  - Une ligne = 1 prêt
  - **307 511 lignes, 122 variables**
- Variables :
  - sur la **personne** : sexe, statut marital, enfants, age, ...
  - **possessions** : logement et ses caractéristiques, voiture, ...
  - **travail** : diplômes, revenus, statut, ...
  - **localisation** : région et ses caractéristiques
  - 3 scores issus de sources externes

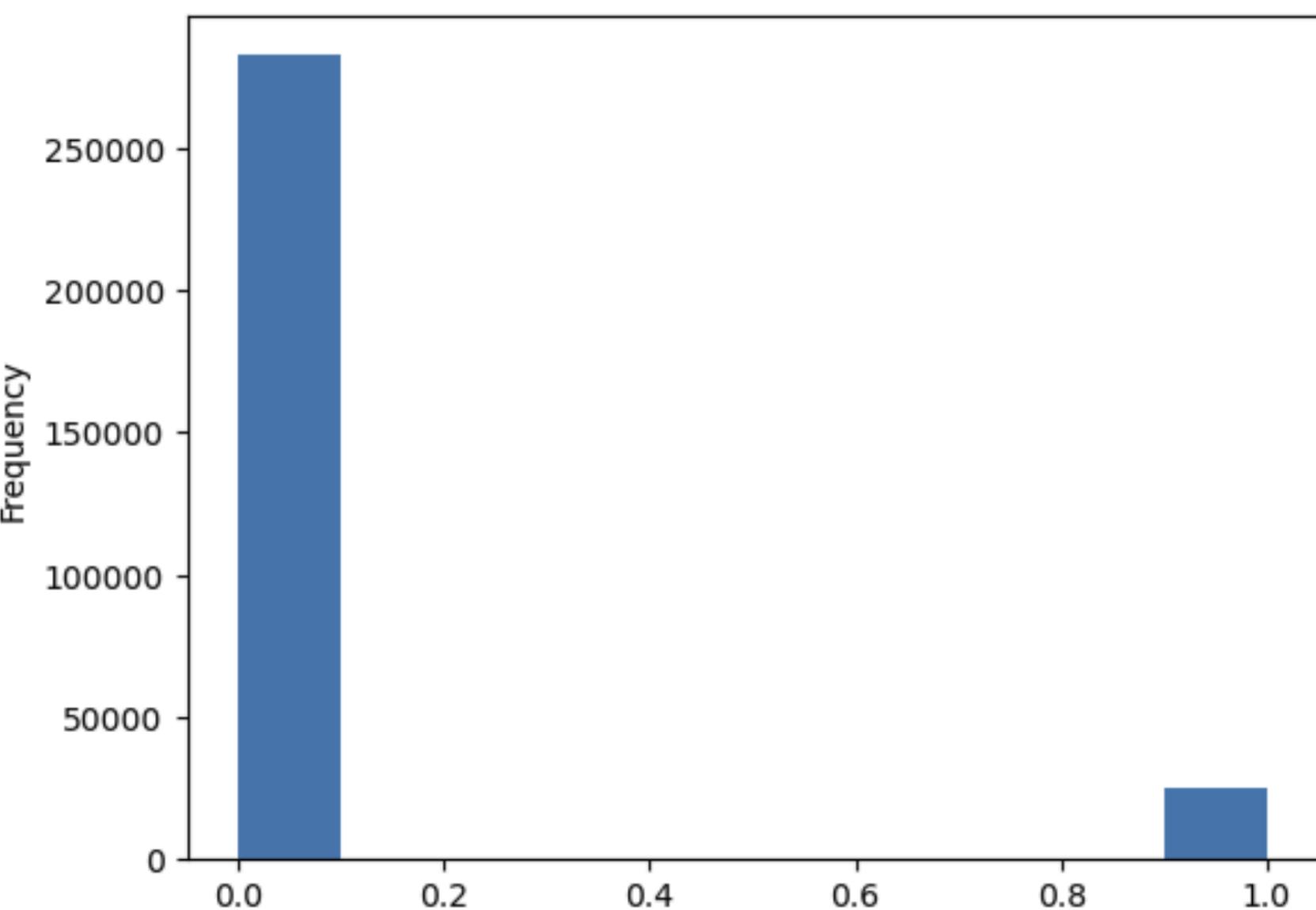


# 1. ANALYSE EXPLORATOIRE DE DONNÉES

*Réalisé à partir du travail antérieur*

# Un jeu déséquilibré

- Les **bons clients** (0) représentent environ **92%** du jeu
- Les **mauvais clients** (1) **8%** du jeu
- Crée des problèmes potentiels pour le bon entraînement des futurs modèles
  - Envisager un **rééquilibrage**



# Des valeurs manquantes

---

- Ci-contre : 20 variables ayant le plus de valeurs manquantes
- **67 colonnes** ont des valeurs manquantes
- Crée des problèmes potentiels pour le bon entraînement des futurs modèles
  - Envisager des **imputations**

	Missing Values	% of Total Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4
NONLIVINGAPARTMENTS_AVG	213514	69.4
FONDKAPREMONT_MODE	210295	68.4
LIVINGAPARTMENTS_MODE	210199	68.4
LIVINGAPARTMENTS_MEDI	210199	68.4
LIVINGAPARTMENTS_AVG	210199	68.4
FLOORSMIN_MODE	208642	67.8
FLOORSMIN_MEDI	208642	67.8
FLOORSMIN_AVG	208642	67.8
YEARS_BUILD_MODE	204488	66.5
YEARS_BUILD_MEDI	204488	66.5
YEARS_BUILD_AVG	204488	66.5
OWN_CAR_AGE	202929	66.0
LANDAREA_AVG	182590	59.4
LANDAREA_MEDI	182590	59.4
LANDAREA_MODE	182590	59.4

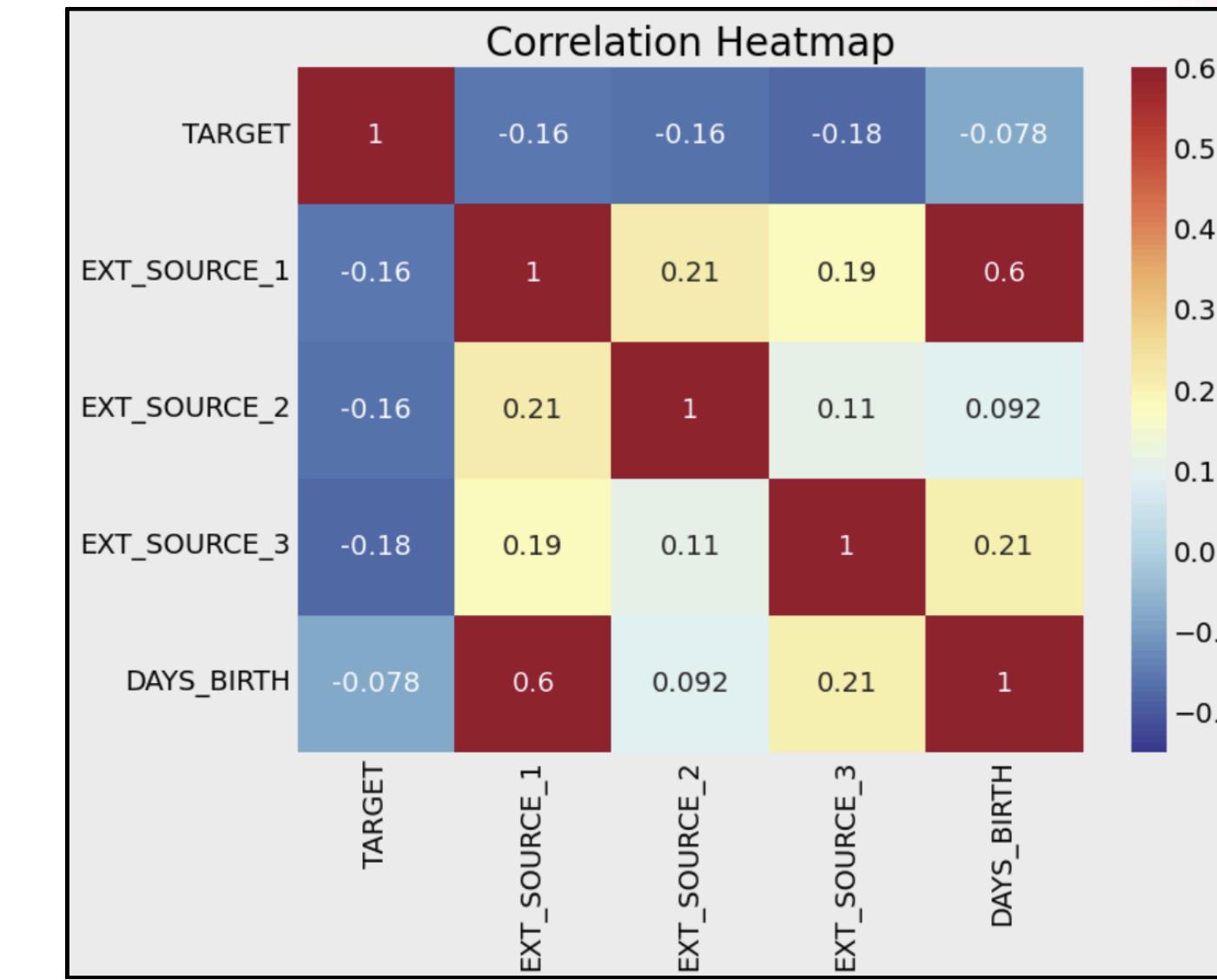
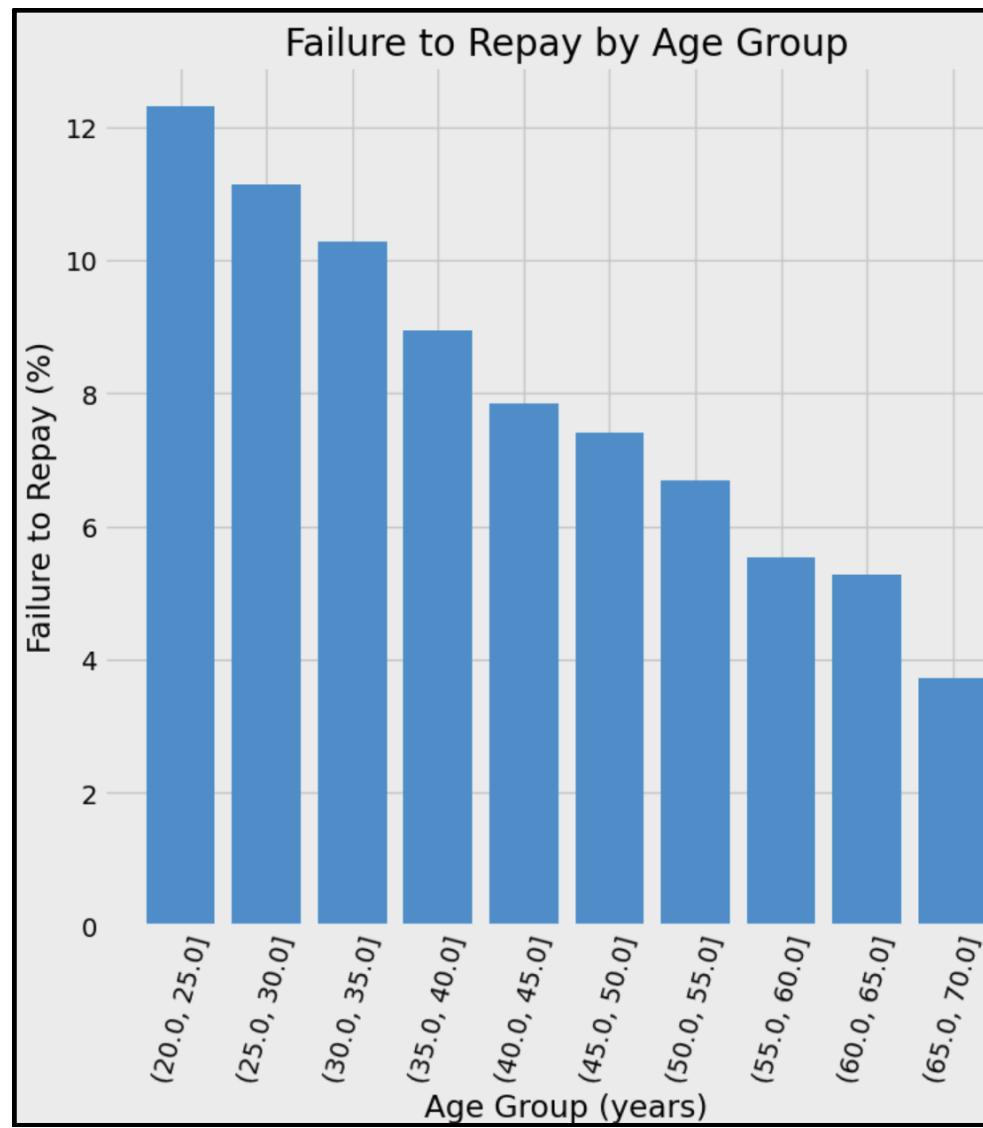
# Des variables catégorielles

- **16 variables catégorielles**
- La majorité multi-catégorielles
- Choix de l'encodage :
  - **One-hot encoder** pour les multi-catégorielles
  - **Label encoder** pour les binaires

NAME_CONTRACT_TYPE	2
CODE_GENDER	3
FLAG_OWN_CAR	2
FLAG_OWN_REALTY	2
NAME_TYPE_SUITE	7
NAME_INCOME_TYPE	8
NAME_EDUCATION_TYPE	5
NAME_FAMILY_STATUS	6
NAME_HOUSING_TYPE	6
OCCUPATION_TYPE	18
WEEKDAY_APPR_PROCESS_START	7
ORGANIZATION_TYPE	58
FONDKAPREMONT_MODE	4
HOUSETYPE_MODE	3
WALLSMATERIAL_MODE	7
EMERGENCYSTATE_MODE	2

# Recherche de corrélations

S'assurer qu'aucune variable n'est trop corrélée à la variable cible = risque de **data leakage**



# Feature engineering

---

- Construction de **4 variables “métier”** :
  - Pourcentage de montant du crédit / revenus annuels du client
  - Pourcentage de mensualité à remboursement / revenus mensuels du client
  - Durée de l'emprunt en mois (montant total / mensualité)
  - Pourcentage de nombre de jours travaillés / âge du client en jours
- Test sur des **baselines** avec et sans nouvelles variables :
  - LogisticRegression : 0.7456 --> 0.7483
  - RandomForest : 0.7081 --> 0.7116
- **Impact limité**

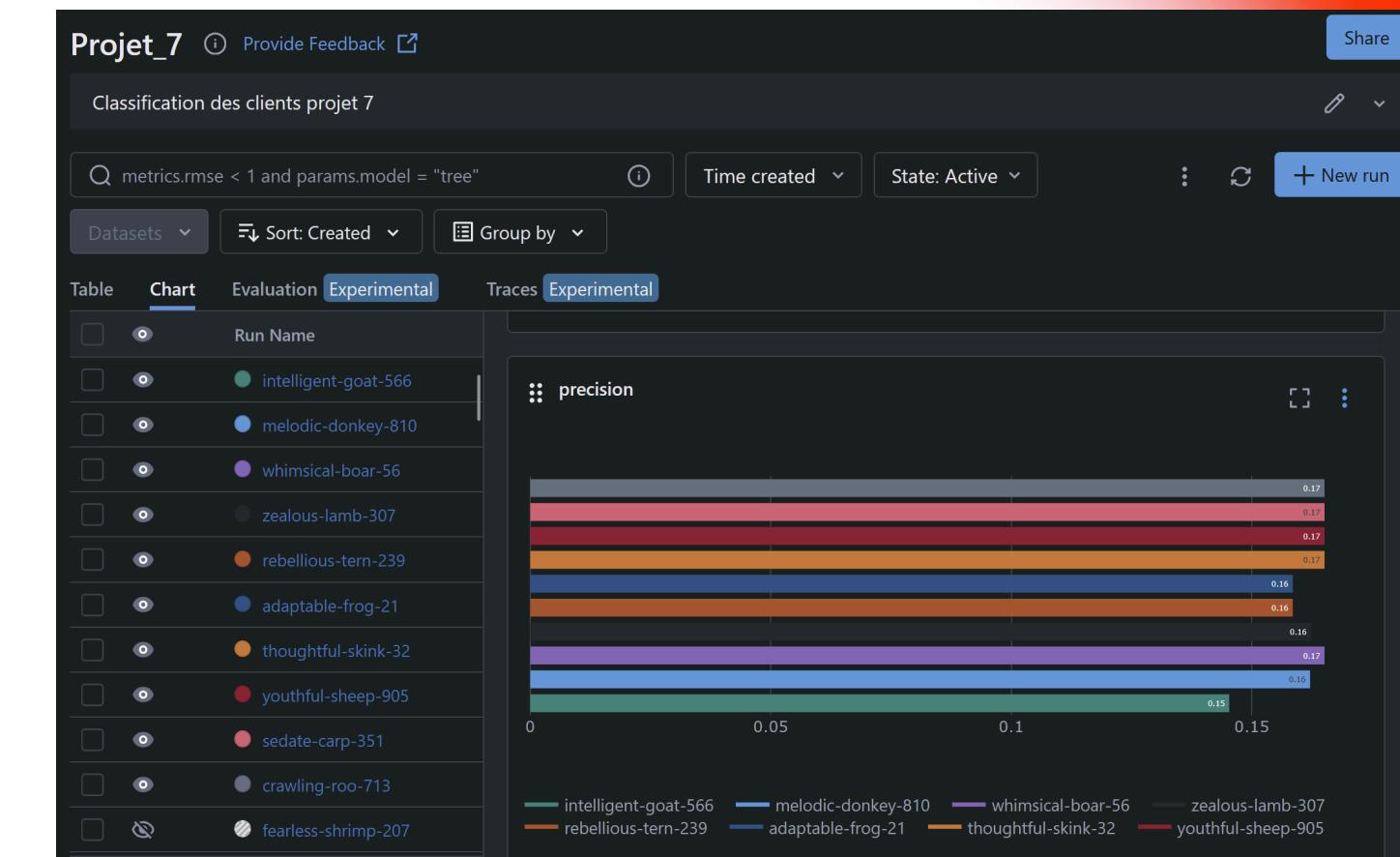
10

## 2. MODÈLES DE CLASSIFICATION



# Suivi des métriques avec MLFlow

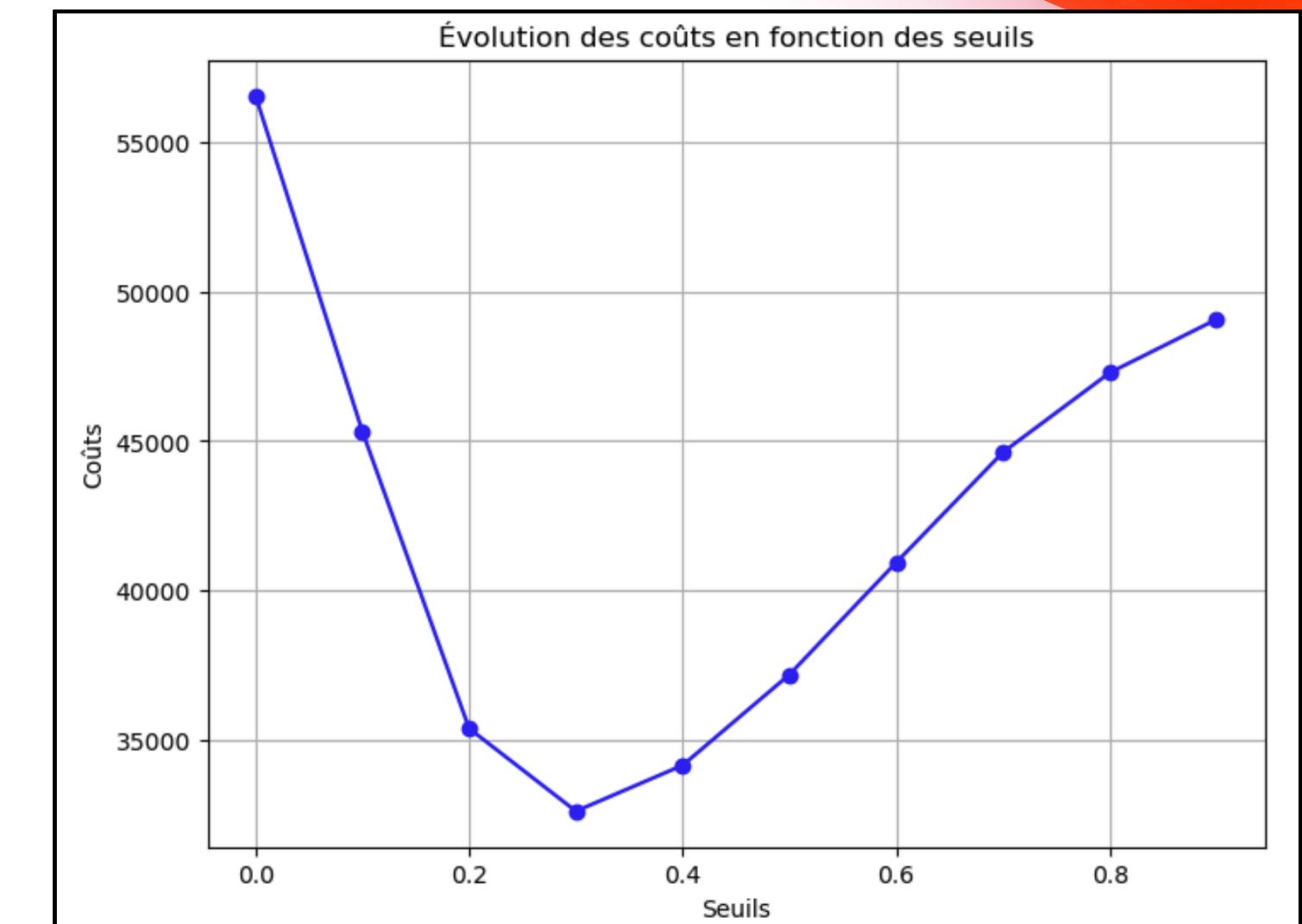
- Métriques surveillées :
  - **ROC AUC** : capacité du modèle à distinguer entre les classes, performance globale du modèle
  - **Précision** : nb de prédictions correctes (vrais positifs + vrais négatifs) / nb total de prédictions, minimiser les faux positifs
  - **Recall** : nb de vrais positifs / nb total de réels positifs (vrais positifs + faux négatifs), minimiser les faux négatifs
  - **Fbeta-score** : on pondère soit le recall soit la précision
    - **F1-score** : moyenne harmonique de la précision et du recall, idée globale de la performance du modèle



# Modèles et choix du protocole

---

- Plusieurs modèles testés : LogisticRegression, Light GBM, Catboost, XGBoost, RandomForest
- Sous-échantillonnage (92/8 à 71/29)
- Sans PCA
- Normalisation avec **MinMaxScaler**
- Randomized Cross Validation
  - Scoring : **fbeta\_score** avec **beta = 2**
- Choix du seuil de classification --> **fonction coût métier**
  - Hypothèse que faux négatifs coûtent **10x plus cher** que faux positifs ( $fp = 1$  ;  $fn = 10$ )
  - On **calcule le coût**  $fp + fn$  pour **chaque seuil** dans une gamme de seuil prédéfinie (entre 0 et 1 avec seuils de 0.1)
  - On retient le seuil associé au **coût le moins élevé**
  - On applique ce seuil aux probabilités issues du modèle testé pour **classifier les individus 0 ou 1**

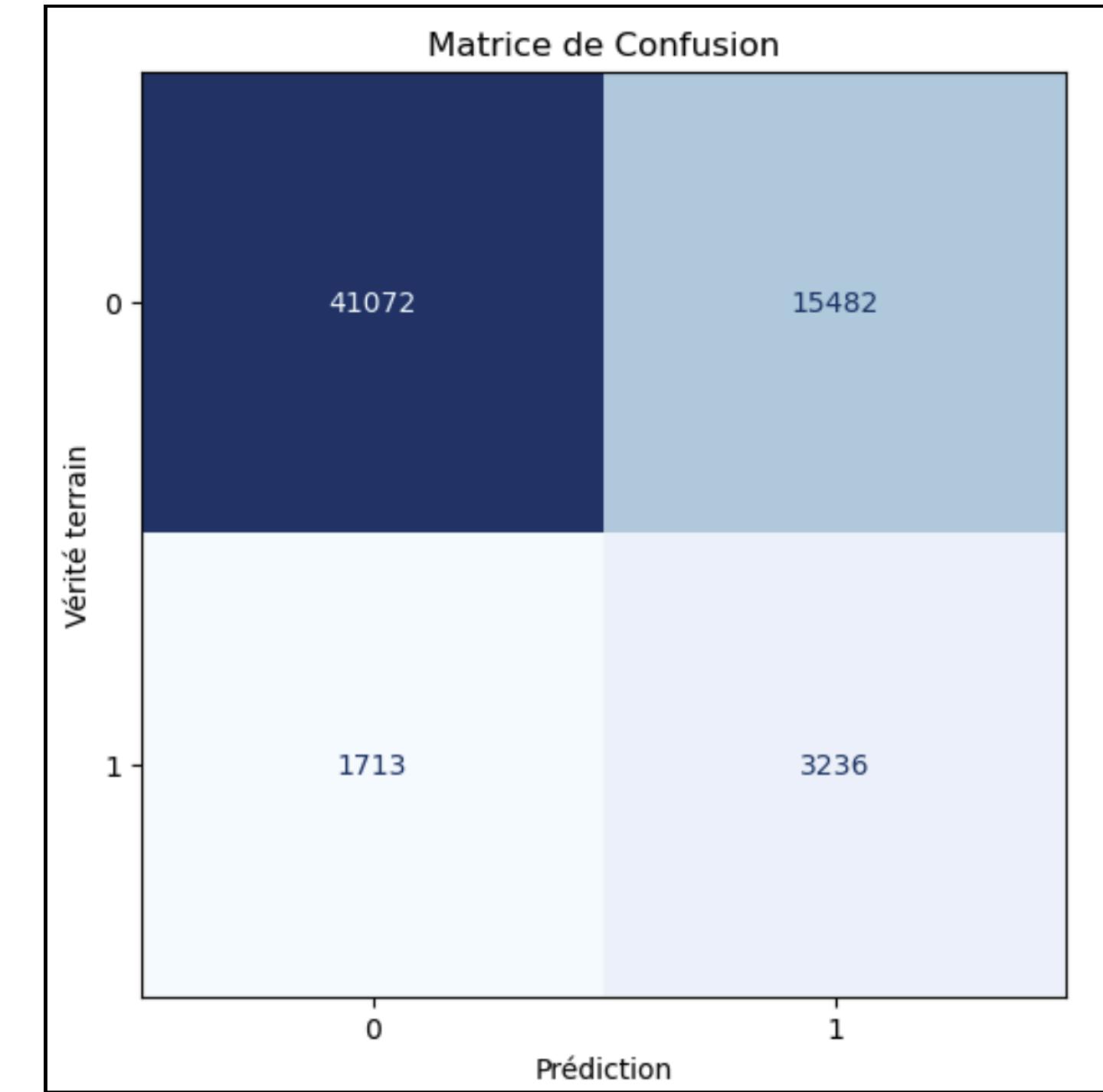


# CatBoost Classifier modèle retenu

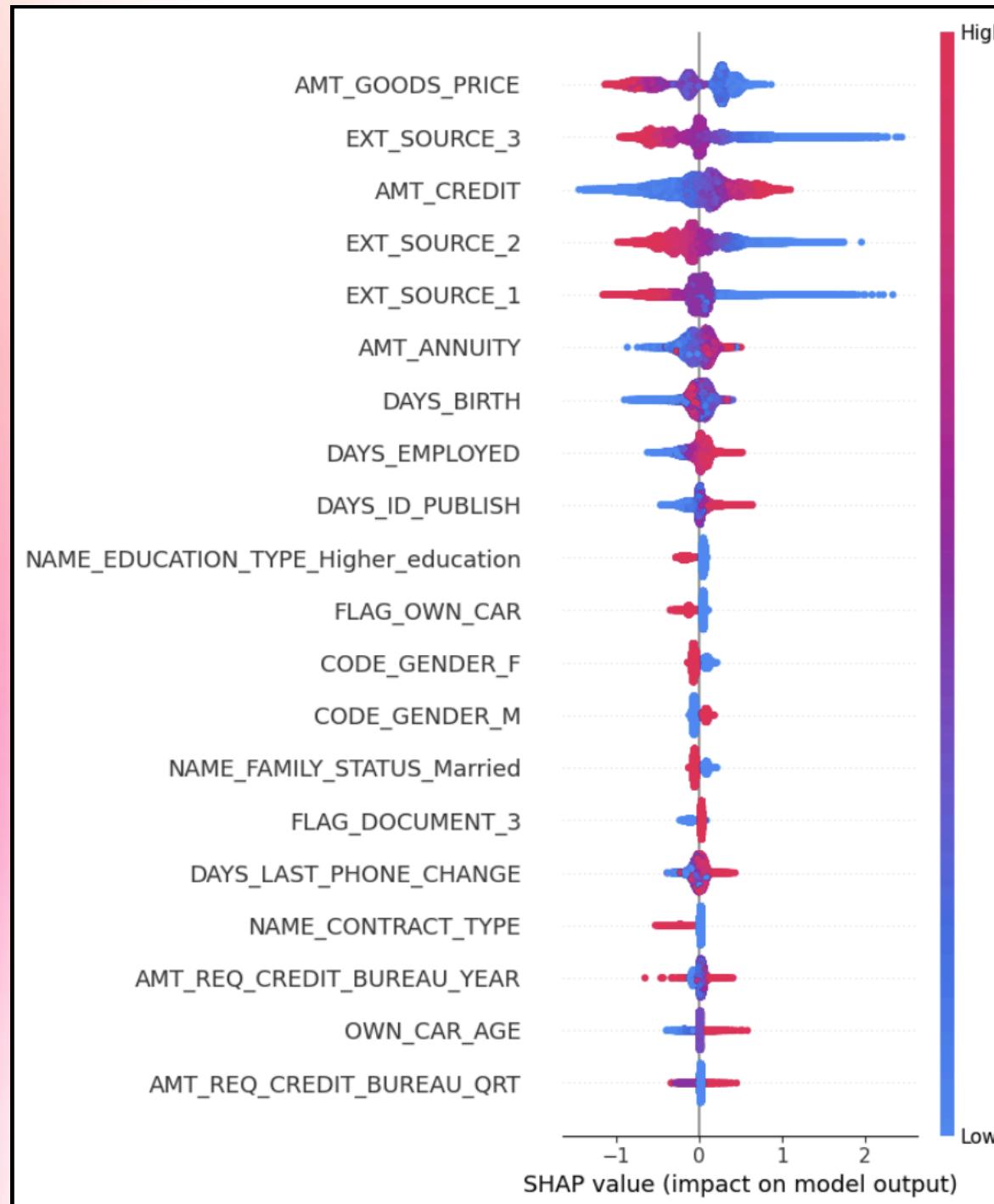
Metrics (4)

Q Search metrics

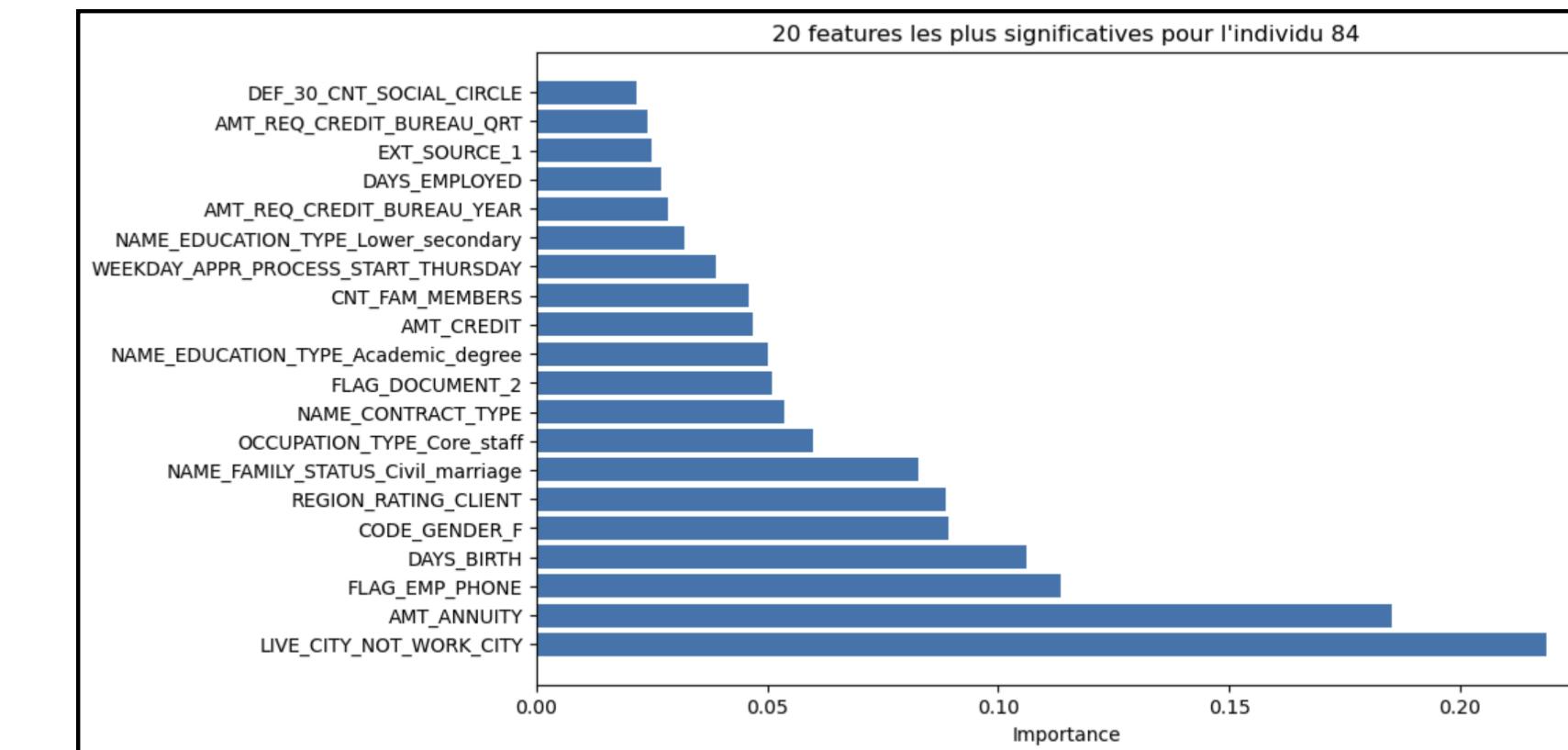
Metric	Value
f1	0.2973
precision	0.2595
recall	0.3479
roc_auc	0.7569



# Interprétabilité du modèle



- Feature importance globale retrouvée dans les feature importance locales
  - Ordre varie en fonction des individus



## 3. DÉPLOIEMENT AUTOMATISÉ

# Gestion des versions et push vers Github

Commits

```

master
Commits on Aug 22, 2024
- Test interface api.ipynb
  corentinlfi committed 6 hours ago · 2 / 2

Commits on Aug 17, 2024
- traitement valeurs manquantes api application_train_preprocessed.csv
  corentinlfi committed 5 days ago · 2 / 2
- mise à jour ci.yml pour dwl lfs
  corentinlfi committed 5 days ago · 2 / 2
- correction chemin acces test unitaires .py
  corentinlfi committed 5 days ago · 1 / 2
- test pytest automatisé
  corentinlfi committed 5 days ago · 1 / 2
- Ajout de application_train_preprocessed.csv à git lfs
  corentinlfi committed 5 days ago · 1 / 2
- Ajout tests unitaires et nouveau modele Catboost
  corentinlfi committed 5 days ago · 1 / 2

Commits on Aug 16, 2024

```



Projet7 Public

Commit	Message	Time Ago
corentinlfi Test interface api.ipynb	b93b414 · 6 hours ago	29 Commits
.github/workflows	mise à jour ci.yml pour dwl lfs	5 days ago
.ipynb_checkpoints	traitement valeurs manquantes api application_train_preproc...	5 days ago
catboost_info	Ajout tests unitaires et nouveau modele Catboost	5 days ago
mlruns	mise à jour du code	last month
.gitattributes	Ajout de application_train_preprocessed.csv à git lfs	5 days ago
Analyse exploratoire et feature engineering.ipynb	changement bdd csv colonne en trop (index)	last week
Classifications.ipynb	test pytest automatisé	5 days ago
Interface API.ipynb	Test interface api.ipynb	6 hours ago
Outils_Open_Source_MLOps.pdf	Ajout initial des fichiers	last month
SAUVEGARDE P7.ipynb	Optimisation modèles	last week
TEST MLFLOW.ipynb	Ajout initial des fichiers	last month
TEST PYCARET.ipynb	mise à jour du code	last month
TEST_FASTAPI.ipynb	Ajout main.py pour test API sur cloud	last month

<https://github.com/corentinlfi/Projet7>

# Tests unitaires et déploiement sur le cloud

The diagram illustrates a continuous integration and deployment (CI/CD) pipeline. It starts with a list of workflow runs on the left, which then leads to detailed logs for each step of the pipeline: testing and deployment.

**Left Panel: All workflows**

- Shows 33 workflow runs.
- Workflow runs include: "maj tests unitaires (encore)", "catboost nouveau + data drift", and "Test interface api.ipynb".
- Each run shows details like Event (master), Status (green checkmark), Branch (master), Actor (corentinlfi), and timestamp.

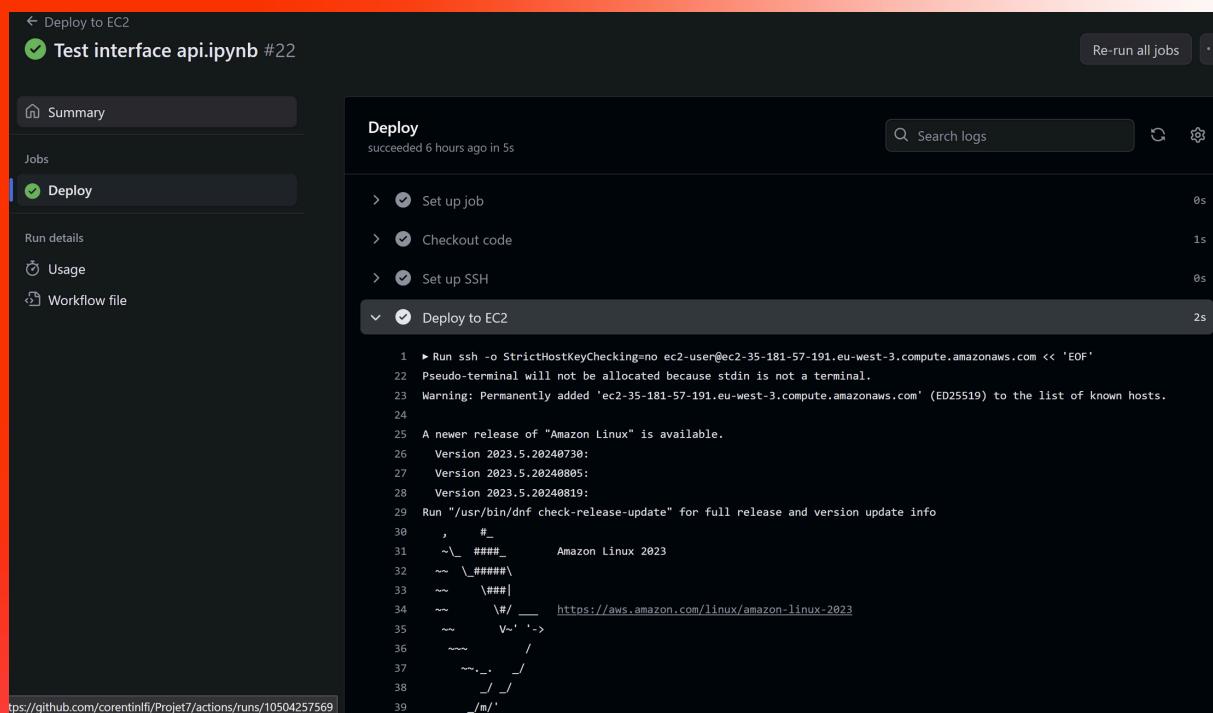
**Middle Panel: Test interface api.ipynb #7**

- Summary of the test job.
- Log details for the "test" job, showing steps like "Set up job", "Checkout code", "Usage", "Setup Python", "Install dependencies", and "Run tests".
- Log output for the "Run tests" step, showing pytest results: "collected 3 items", "3 passed in 11.21s".

**Right Panel: Deploy to EC2**

- Summary of the Deploy job.
- Log details for the "Deploy" job, showing steps like "Set up job", "Checkout code", "Usage", "Setup SSH", and "Deploy to EC2".
- Log output for the "Deploy to EC2" step, showing SSH commands and package updates: "Run ssh -o StrictHostKeyChecking=no ec2-user@ec2-35-181-57-191.eu-west-3.compute.amazonaws.com << 'EOF'", "Warning: Permanently added 'ec2-35-181-57-191.eu-west-3.compute.amazonaws.com' (ED25519) to the list of known hosts.", "A newer release of 'Amazon Linux' is available.", "Version 2023.5.20240730:", "Version 2023.5.20240805:", "Version 2023.5.20240819:", "Run /usr/bin/dnf check-release-update for full release and version update info", and "Amazon Linux 2023".

# Interface d'accès à l'API



Entrez l'ID du client recherché : 456254  
Client ID: 456254, Solvable: True  
Informations sur le client :

	feature	value
0	SK_ID_CURR	456254
1	NAME_CONTRACT_TYPE	0
2	FLAG_OWN_CAR	0
3	FLAG_OWN_REALTY	1
4	CNT_CHILDREN	0
5	AMT_INCOME_TOTAL	171000.0
6	AMT_CREDIT	370107.0

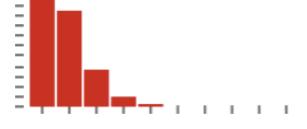
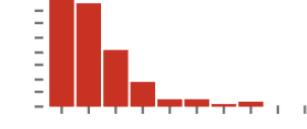
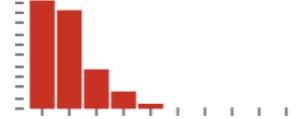
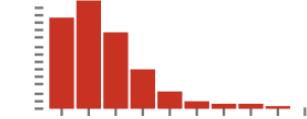
[http://ec2-35-181-57-191.eu-west-3.compute.amazonaws.com:8000/predict/{client\\_id}](http://ec2-35-181-57-191.eu-west-3.compute.amazonaws.com:8000/predict/{client_id})

## 4. POST-DÉPLOIEMENT

# Surveillance du data drift

---

Drift is detected for 7.438% of columns (9 out of 121).

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
» AMT_REQ_CREDIT_BUREAU_QRT	num			Detected	Wasserstein distance (normed)	0.359052
» AMT_REQ_CREDIT_BUREAU_MON	num			Detected	Wasserstein distance (normed)	0.281765
» AMT_GOODS_PRICE	num			Detected	Wasserstein distance (normed)	0.210785
» AMT_CREDIT	num			Detected	Wasserstein distance (normed)	0.207334

## 5. CONCLUSIONS ET PERSPECTIVES

# Conclusions

---

- La mise en place d'un modèle de scoring est possible, et va permettre de diminuer les risques de l'entreprise
- Il s'agit d'un outil d'**aide à la décision** et ne peut en aucun cas remplacer une expertise humaine
- L'interprétabilité du modèle permettra également d'ajouter de la **transparence** à la prise de décision
- Le déploiement sur le cloud ouvre la voie vers une déploiement généralisé de l'outil

# Perspectives

---

- **Feature importance globale** : faire par groupe mauvais clients/bons clients pour montrer les raisons d'acceptation/de refus
- Creuser davantage le **feature engineering** avec des gens du métier pour identifier des variables pertinentes
- Explorer les **autres jeux de données** pour voir impact sur la capacité de classification

# MERCI !

Avez-vous des questions ?