Projet 7 - Implémentez un modèle de Scoring

Création d'une application de détection de faillite bancaire pour la société Prêt à dépenser qui propose des crédits à la consommation pour des clients n'ayant pas ou peu d'historique de prêts

Sommaire


Objectifs et contexte de la mission de Data Science


Exploration des données et Feature Engineering


Méthodologie de Modélisation


Modèle choisi et interprétabilité

Création et déploiement de l'API (mise en place de tests unitaires)


Création et déploiement du Dashboard

puis remarques & axes d'amélioration.

Objectifs & Contexte de la mission

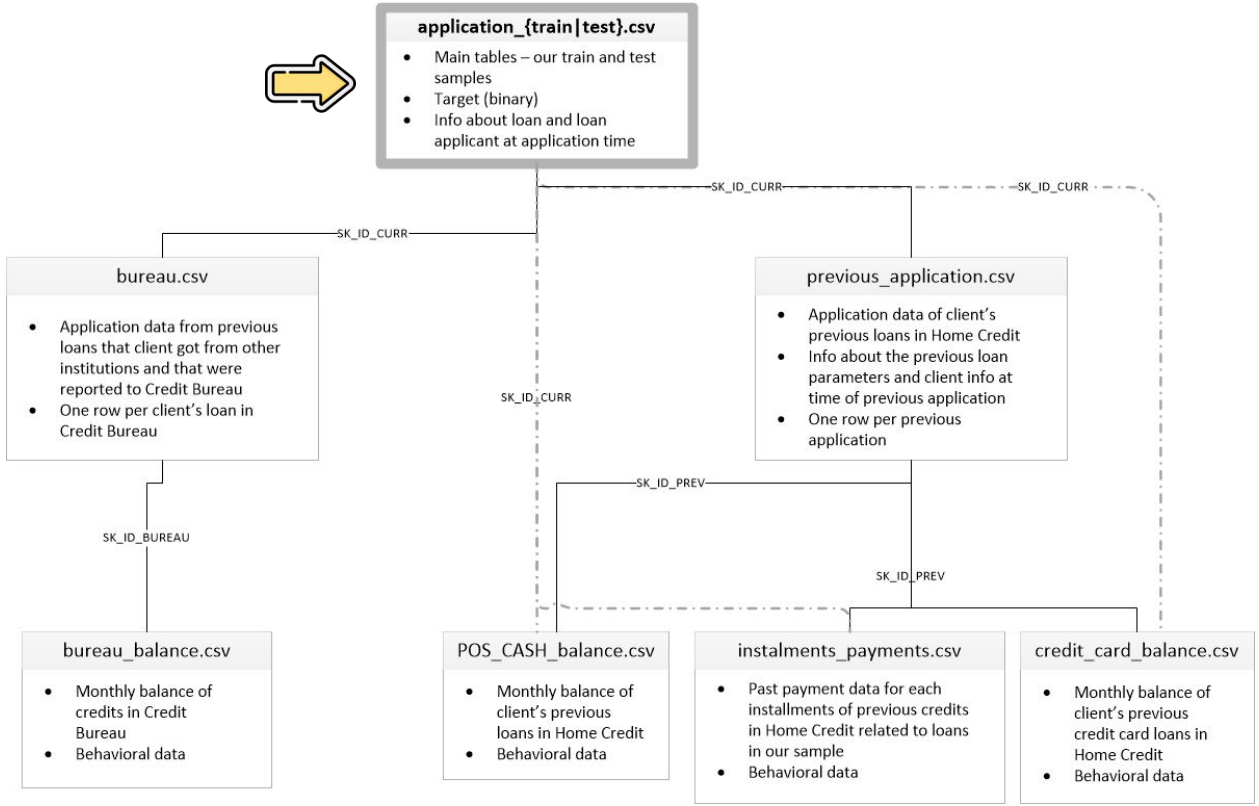
🎯 Objectifs & Contexte de la mission

Contexte : Prêt à dépenser propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt. L'entreprise souhaite mettre en œuvre un outil de "scoring crédit" pour calculer la probabilité qu'un client rembourse son crédit, puis classifier la demande en accord ou refus. Les clients ont fait remonter un besoin de transparence sur les raisons desdits accord ou refus.

Objectif : développer un dashboard interactif pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d'octroi de crédit.

Moyens : Une BDD clients anonymisées avec des données comportementales, financières, si la personne était accompagné lors de la demande de crédit et par qui, nombre d'enfants etc. Les données se trouvent ICI.

Les données

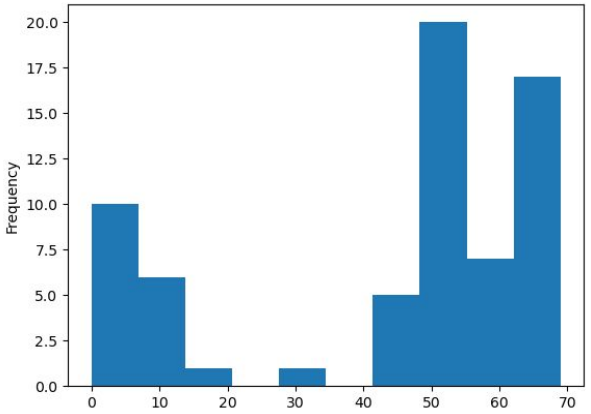

application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
previous_application.csv
Application data of client's previous loans in Home Credit
Info about the previous loan parameters and client info at time of previous application
One row per previous application
SK_ID_BUREAU
SK_ID_PREV
SK_ID_PREV
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data
instalments_payments.csv
Past payment data for each installments of previous credits in Home Credit related to loans in our sample
Behavioral data
credit_card_balance.csv
Monthly balance of client's previous credit card loans in Home Credit
Behavioral data

Exploration des données & Feature Engineering

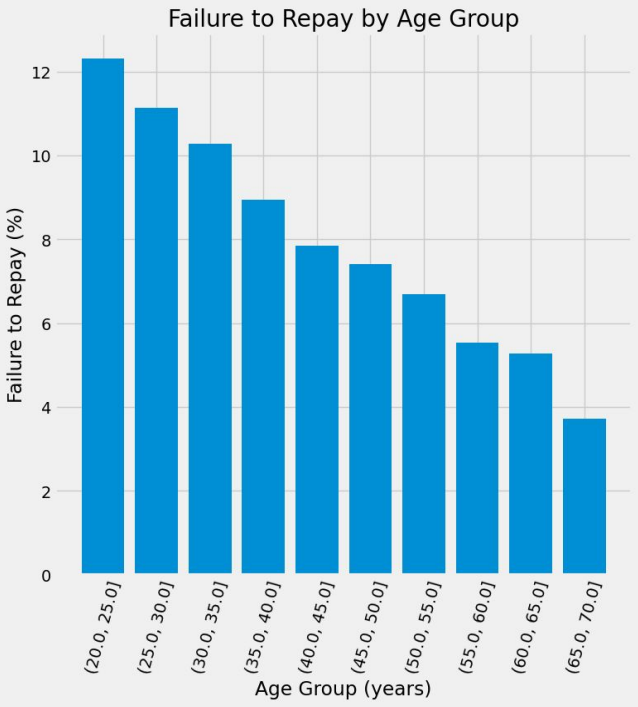
🔎 Exploration des données et Feature Engineering

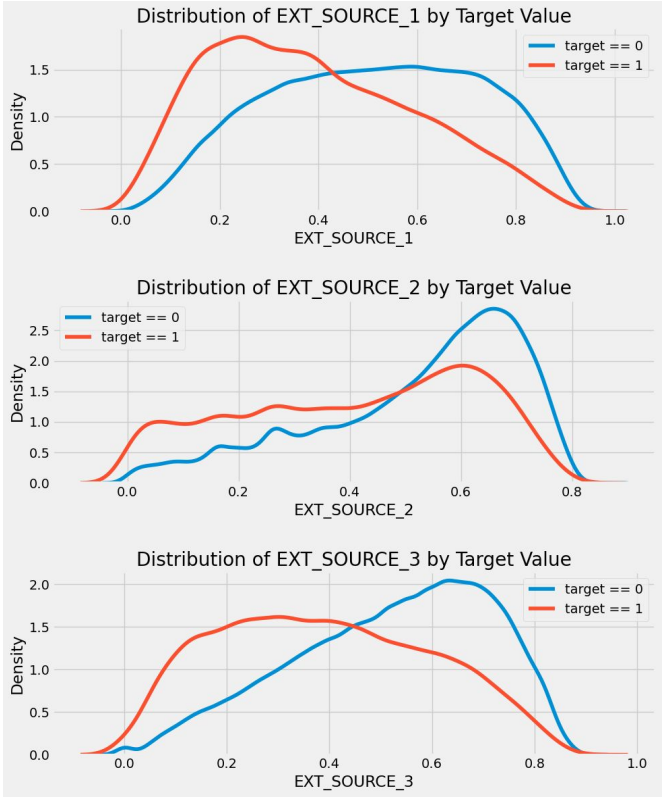
- **Répartition des classes pour la faillite bancaire** : 91.93% sans problème de remboursement 🙂 et 8.07% avec des soucis 😡 ⇒ **Imbalanced Problem**
- **Données manquantes** :67/122 colonnes ⇒ Simple Imputer avec median Strategy
- **Les différents types de variables** : 65 float64, 41 int64 et 16 object. ⇒ encodage
- **Encodage des variables catégorielles** :
  - **Label Encoder** pour les variables catégorielles à 2 catégories
  - **One Hot Encoding** pour les variables catégorielles à plus de 2 catégories pour que le modèle ne mésinterprète pas les poids associés par un label encoder arbitraire.
- **Traitement des anomalies** : 18% des individus ont près de 1000 ans de 'DAYS_EMPLOYED' ⇒ remplacement par NaN et créer une colonne pour flag les individus avec l'anomalie 'DAYS_EMPLOYED_ANOM'
- **MinMax Scaler 0 à 1**

Tableau de répartition des fréquences du pourcentage de données manquantes par colonne



Exploration des données et Feature Engineering





Exploration des données et Feature Engineering

Data de Base

données telles que modifiées jusqu'à présent

- Data de Base

Data Poly

Ajout de polynomial features

['EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3', 'DAYS_BIRTH']

- Data de Base
- + Polynomial Features

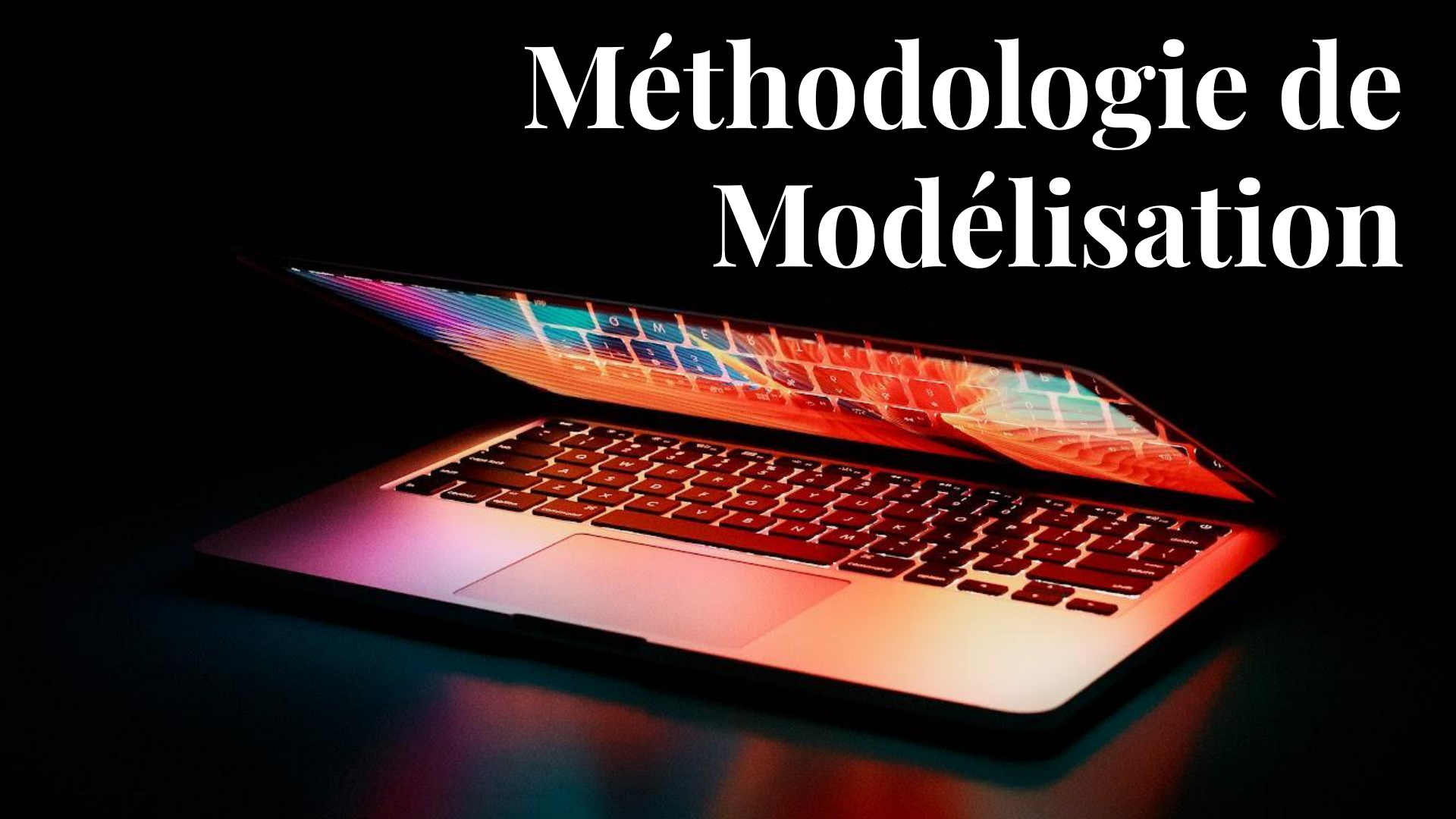
Données Domain Knowledge

Ajout d'indicateurs financiers

`CREDIT_INCOME_PERCENT`: `ANNUITY_INCOME_PERCENT`: `CREDIT_TERM` `DAYS_EMPLOYED_PERCENT`

- Data de Base
- + Domain Knowledge Features

Méthodologie de Modélisation



Méthodologie de modélisation

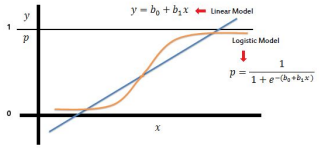
- Séparation des données en train et test (test size = 20%)
- Tester différents modèles
- Création d'une fonction pour rechercher les hyperparamètres pour chaque modèle et afficher les métriques qui servent de comparaison :
    - **Recall** : proportion de vrais positifs correctement identifiés parmis tous les cas réels positifs
    - **F1-Score** : moyenne harmonique de la précision et du recall, offrant un équilibre entre les deux
    - **AUC-ROC** : Aire sous la courbe ROC, qui mesure la capacité d'un modèle à distinguer entre les classes à tous les seuils de classification.
- Traitement du déséquilibre des classes avec le paramètre 'class_weight' = 'balanced'
- Création d'une fonction de coût métier pour trouver le meilleur modèle qui réponde à la problématique métier
- Choix du modèle

Les méthodes testées pour modéliser

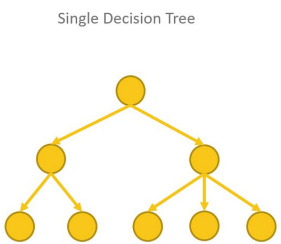
MÉTHODES PLUS TRADITIONNELLES

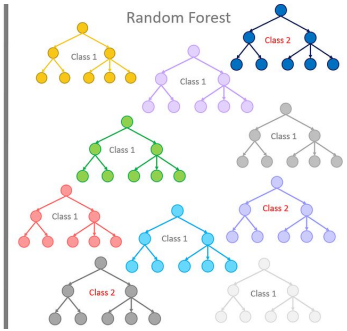
Dummy Classifier (most frequent strategy)

Régression Logistique



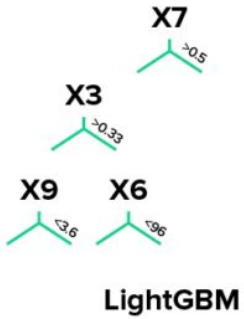
Arbre de décision & Forêts Aléatoires

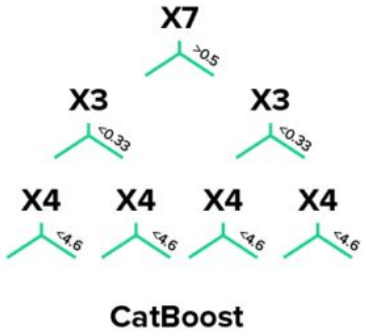




Méthodes de Boosting

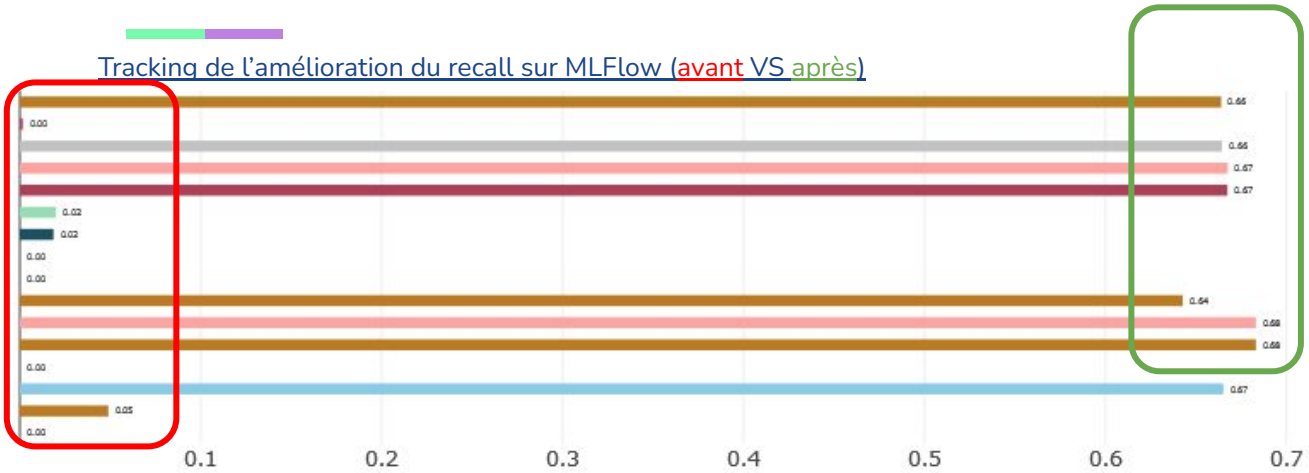
LightGBM & Catboost

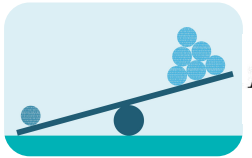




Traitement du déséquilibre des classes

Tracking de l'amélioration du recall sur MLFlow (avant VS après)





$$Recall = \frac{TP}{TP + FN}$$

Répartition des classes pour la faillite bancaire :

91.93% sans problème de remboursement 🙂 0

8.07% avec des soucis 😡 1

A cause du problème de répartition des classes soit la **sous représentation des positifs**(personnes ne remboursant pas leur crédit) , beaucoup sont détectés comme négatif (sans problème) : on a donc **énormément de faux négatifs** !

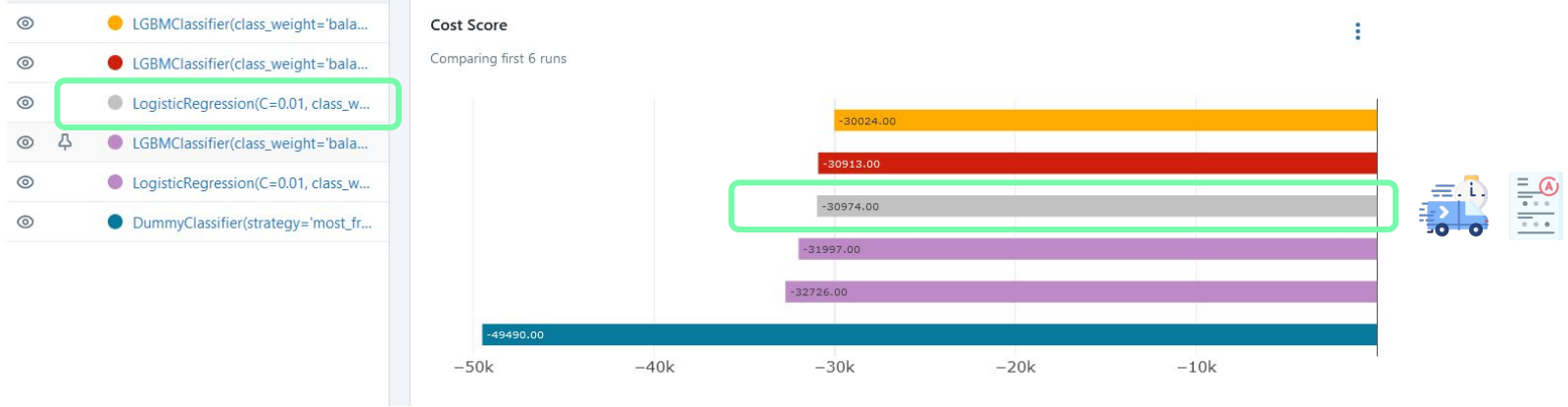
⇒ **mauvais Recall** qui est nettement amélioré par l'utilisation du paramètre **class_weight**.

Il permet lors de l'apprentissage du modèle de **pénaliser** plus durement les **erreurs de la classe sous représentée**.

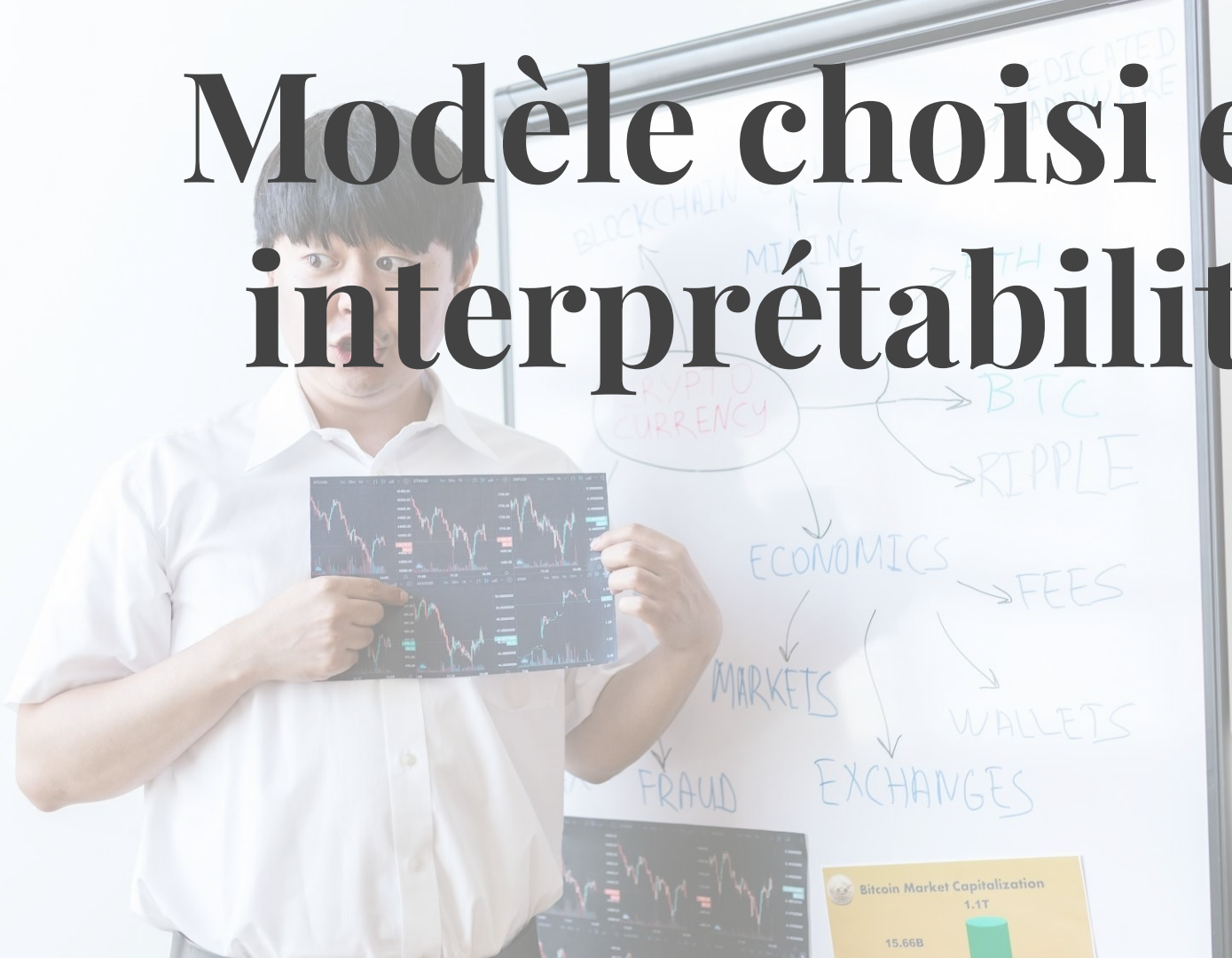
Fonction de coût métier & choix du modèle

```python
def cost_function(y_true, y_pred):
    tp = np.sum((y_true == 0) & (y_pred == 0))
    tn = np.sum((y_true == 1) & (y_pred == 1))
    fp = np.sum((y_true == 0) & (y_pred == 1))
    fn = np.sum((y_true == 1) & (y_pred == 0))

    total_cost = tn * 1 - fp * 1 - fn * 10
    return total_cost


from sklearn.metrics import make_scorer

cost_scorer = make_scorer(cost_function, greater_is_better=True)
```

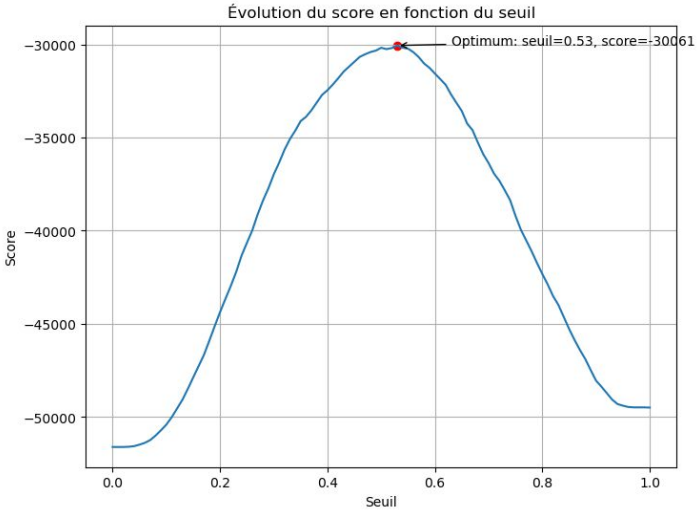
Vrai Positif = Neutre
Vrai Négatif = Gain de 1 (perte évitée)
Faux Positif = Perte de 1 (manque à gagner)
Faux Négatif= Perte de 10 (perte en capital non remboursé)



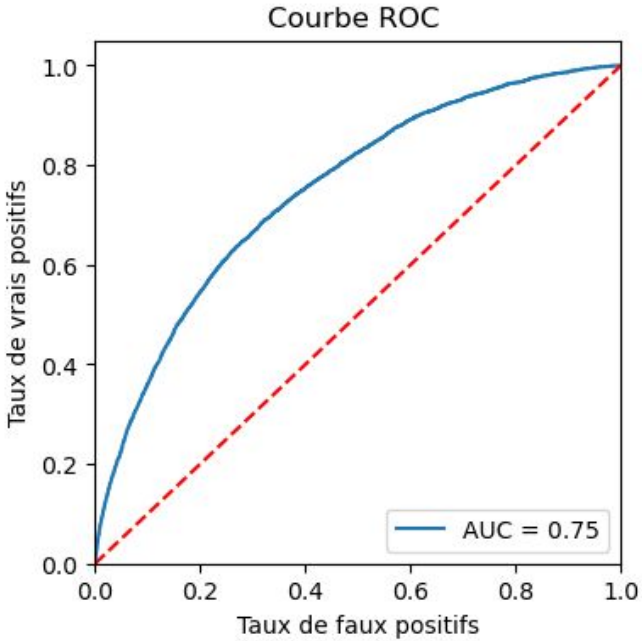
Modèle choisi et interprétabilité



Modèle choisi & Interprétabilité



Optimisation du seuil de détection de faillite bancaire en faisant varier les seuils de détection



Régression Logistique

Recall : 064

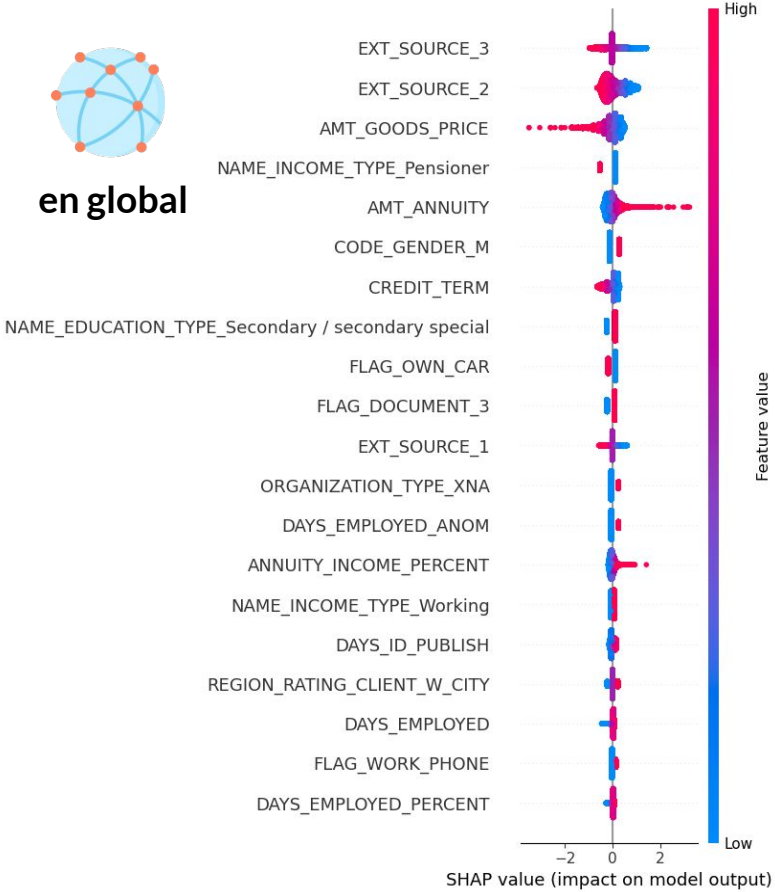
F1-Score : 0.27

AUC-ROC : 0.749

Cost Function : -30061

Modèle choisi & Interprétabilité

en global



en local

Meilleur client

Pire client

Création & déploiement de l'API

Création & Déploiement de l'API

Outils

Git pour le versionnage et Github pour l'hébergement >Lien du Repo API<

FastAPI pour la création de l'API

Heroku pour le déploiement continu de l'API à partir du repo GitHub >Lien API< (off après soutenance)

Github Actions & Pytest pour la mise en place des tests unitaires

In & Out

Input : données clients brutes

Processing : featuring engineering puis prédiction du score avec le modèle choisi

Output : Score, Prédiction (binary), infos nécessaires pour l'interprétabilité locale shap

```
test_api.py::test_make_prediction1
test_api.py::test_make_prediction2
test_api.py::test_make_prediction3
========================================== 3 passed, 4 warnings in 5.13s ==
```

- **final test pr deploy cont**
  Push Test Workflow #6: Commit cdbf734 pushed by naomigua — `main` — 4 minutes ago — 1m 13s
- **Merge pull request #1 from naomigua/git...**
  Push Test Workflow #5: Commit 4e8bf44 pushed by naomigua — `main` — 11 minutes ago — 1m 2s
- **pour test**
  Push Test Workflow #4: Pull request #1 opened by naomigua — `githubactions_test` — 11 minutes ago — 1m 16s
- **pour test**
  Push Test Workflow #3: Commit d4f4abe pushed by naomigua — `githubactions_test` — 14 minutes ago — 1m 7s
- **requirements add httpx**
  Push Test Workflow #2: Commit 4680ae2 pushed by naomigua — `main` — 17 minutes ago — 1m 14s
- **Github Action Workflow, pytest et maj req...**
  Push Test Workflow #1: Commit 01eb755 pushed by naomigua — `main` — 21 minutes ago — 1m 20s

Automatic deploys from `main` are enabled

naomi.guarneri@gmail.com: Deployed `cdbf734b`
Today at 11:05 AM · v4 · Compare diff

naomi.guarneri@gmail.com: Build succeeded
Today at 11:03 AM · View build log

naomi.guarneri@gmail.com: Deployed `b4b2b26c`
Jul 14 at 3:40 PM · v3

naomi.guarneri@gmail.com: Build succeeded
Jul 14 at 3:38 PM · View build log

pour test #1

Merged — naomigua merged 1 commit into `main` from `githubactions_test` 12 minutes ago

Conversation 0 · Commits 1 · Checks 2 · Files changed 1

naomigua commented 12 minutes ago — Owner

Test passé sur GitHub Actions donc Merge autorisé

pour test — d4f4abe

naomigua merged commit `4e8bf44` into `main` 12 minutes ago — View details · Revert
2 checks passed

Pull request successfully merged and closed
You're all set—the `githubactions_te...` branch can be safely deleted. — Delete branch


Création &
déploiement du
Dashboard

Création & Déploiement du dashboard

Outils

Git pour le versionnage et Github pour l'hébergement >Lien du Repo API<

Streamlit pour la création du dashboard

Heroku pour le déploiement du dashboard à partir du repo GitHub >Lien Dashboard< (off après soutenance)

3 volets

Tableau clientèle : parcourir les données clients avec des filtres

Comparaison : comparer les clients entre eux

Visualisation score : prédiction de faillite bancaire et viz de l'interprétabilité locale

Application de détection de faillite bancaire

Parcourir les données clients

Filtrer par NAME_CONTRACT_TYPE: all

Filtrer par CODE_GENDER: all

Filtrer par NAME_INCOME_TYPE: all

Filtrer par NAME_EDUCATION_TYPE: all

Filtrer par NAME_FAMILY_STATUS: all

Minimum pour CNT_CHILDREN: 0

Minimum pour AMT_INCOME_TOTAL: 26100.00

Maximum pour CNT_CHILDREN: 12

Maximum pour AMT_INCOME_TOTAL: 117000000.00

Nombre de correspondances trouvées: 61503

	SK_ID_CURR	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	NAME_TYPE_SUITE
0	384,575	Cash loans	M	Y	N	2	207,000	465,457.5	52,641	418,500	Unaccompanied
1	214,010	Cash loans	F	Y	Y	0	247,500	1,281,712.5	48,946.5	1,179,000	Unaccompanied
2	142,232	Cash loans	F	Y	N	0	202,500	495,000	39,109.5	495,000	Unaccompanied
3	389,171	Cash loans	F	N	Y	0	247,500	254,700	24,939	225,000	Unaccompanied
4	283,617	Cash loans	M	N	Y	0	112,500	308,133	15,862.5	234,000	Unaccompanied
5	362,171	Cash loans	M	N	Y	0	85,500	152,820	16,456.5	135,000	Unaccompanied
6	180,689	Cash loans	F	N	N	1	112,500	900,000	24,750	900,000	Family
7	310,328	Cash loans	M	Y	Y	0	141,606	810,000	33,120	810,000	Unaccompanied

Comparaison clientèle

Comparaison du client avec les autres clients

Sélectionnez le numéro du client

384575

	SK_ID_CURR	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_P
0	384,575	Cash loans	M	Y	N	2	207,000	465,457.5	52,641	418

Informations sur le client sélectionné :

ID client : 384575

Genre : M

Type de contrat : Cash loans

Nombre d'enfant(s) : 2

Revenu total : 207000.0

Type de revenu : Commercial associate

Type d'éducation : Secondary / secondary special

Statut familial : Married

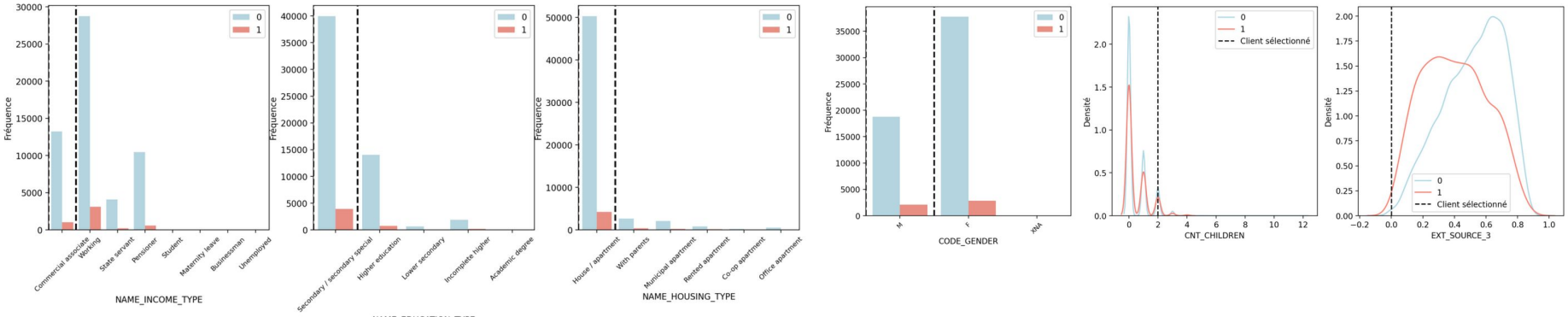
Type de logement : House / apartment

Graphiques de comparaison avec les autres clients :

Sélection du nombre de clients les plus proches pour la comparaison

15000

100 61503



Application de détection de faillite bancaire

Visualisation des scores de prédiction

Veuillez sélectionner un numéro de demande de prêt

Sélectionnez le numéro du client

384575

	SK_ID_CURR	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE
0	384,575	Cash loans	M	Y	N	2	207,000	465,457.5	52,641	418,500

Informations sur le client sélectionné :

ID client : 384575

Genre : M

Type de contrat : Cash loans

Nombre d'enfant(s) : 2

Revenu total : 207000.0

Type de revenu : Commercial associate

Type d'éducation : Secondary / secondary special

Statut familial : Married

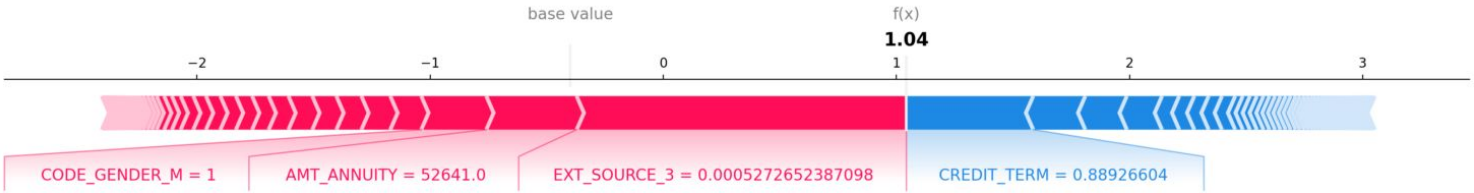
Type de logement : House / apartment

Prédiction de Faillite Bancaire :

Attention client à risque !

Prédiction de faillite bancaire : Oui

Score de faillite bancaire (seuil 0,53) : 0.7314



Analyse du Data Drift via evidently

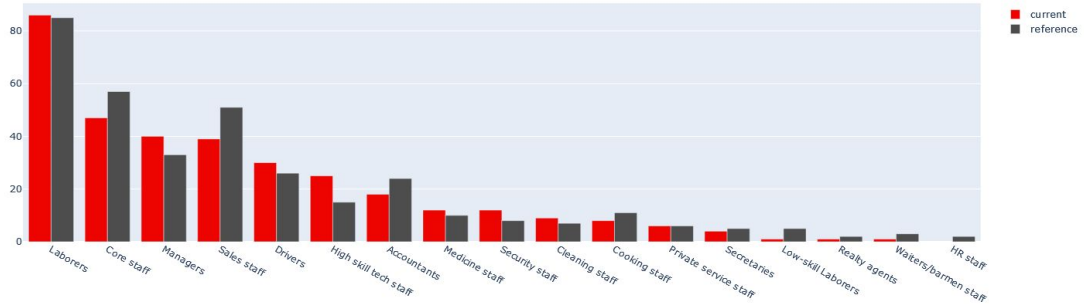
Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

120	7	0.0583
Columns	Drifted Columns	Share of Drifted Columns

Pas de data drift notoire sur ce jeu de données.

Exemple de colonne qui a très faiblement drift ‘OCCUPATION_TYPE’



Cela dit, il sera forcément **utile** de le **remettre en question** de temps à autre.

Conclusions & Remarques

Ce que nous soumettons :

- Un modèle de classification basée sur une régression logistique avec un seuil optimisé via une fonction coût métier pour la détection du risque de faillite bancaire.

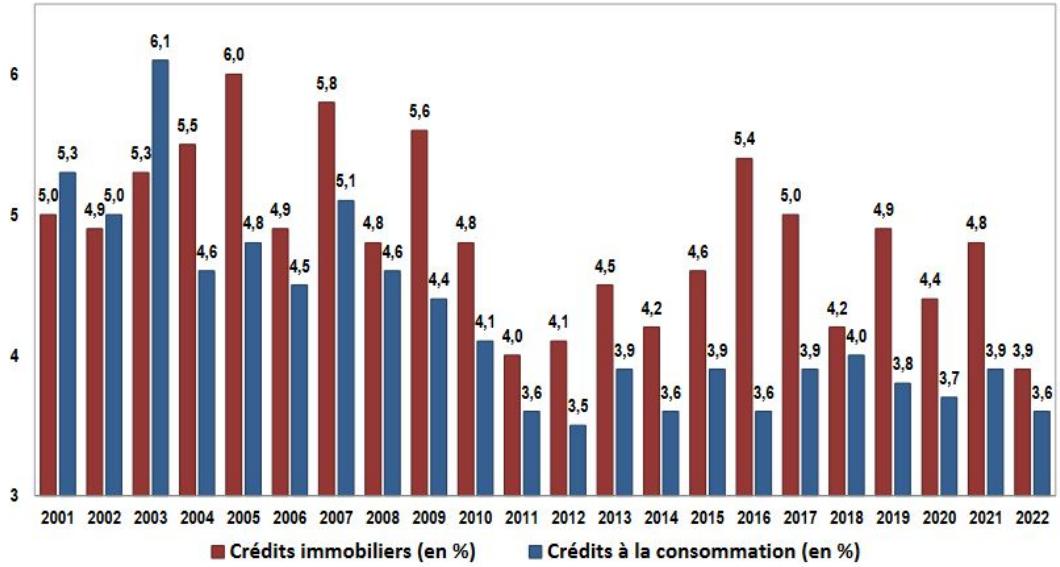
Améliorations possibles :

- Meilleur recherche des hyperparamètres pour les modèles de boosting, plus efficaces mais plus long.
- Point de vigilance : la conjecture économique cause une baisse des demandes de crédits à la consommation, à prendre en considération pour ajuster le seuil en fonction cf. slides suivantes

Les intentions de souscription de nouveaux crédits pour les 6 prochains mois en recul.

Les intentions concernant les crédits immobiliers reculent nettement, après le rebond de 2021 : elles descendent à leur niveau le plus bas depuis 1997, très en deçà de leur moyenne de longue période (4,8 %). Le repli constaté pour les crédits à la consommation est moindre : ces intentions sont moins fréquentes qu'en longue période (3,8 % depuis 2011).

La part des ménages ayant l'intention de souscrire des crédits
(Source : Observatoire des Crédits aux Ménages)

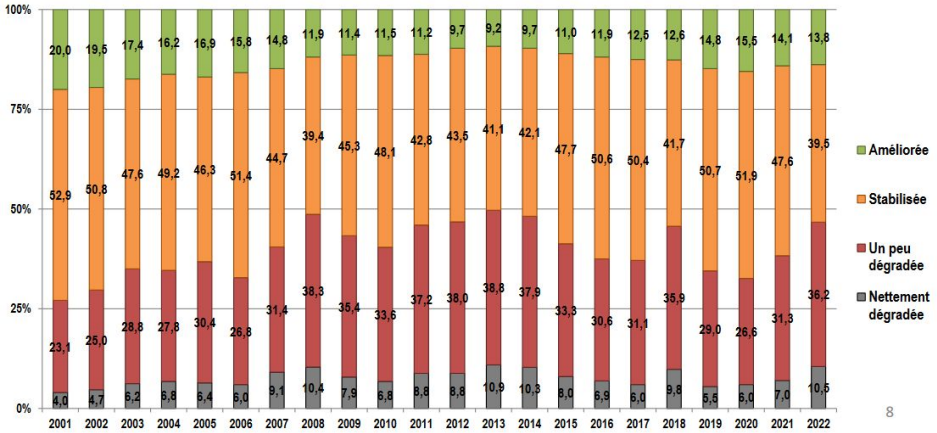


Alors que l'environnement économique et financier s'est fortement dégradé en 2022, les intentions de souscription de crédits immobiliers reculent et ne concernent plus que 3,9 % des ménages. De même les intentions concernant les crédits à la consommation s'affaiblissent, 3,6 % des ménages en faisant état : ceux-ci recourent le plus souvent à ces crédits pour réaliser des projets de consommation durable.

L'appréciation portée par les ménages sur leur situation financière s'est nettement dégradée en 2022, après avoir commencé à se détériorer en 2021.

13,8 % considèrent qu'elle s'est améliorée (14,1 % en 2021 et 15,5 % en 2020)
39,5 % qu'elle s'est stabilisée (47,6 % en 2021 et 51,9 % en 2020)
36,2 % qu'elle s'est un peu dégradée (31,3 % en 2021 et 26,6 % en 2020)
10,5 % qu'elle s'est nettement dégradée (7,0 en 2021 et 6,0 % en 2020)

Cette situation est comparable à celles constatées durant la 1ère moitié des années 2010.
Pour la plupart des ménages, le pouvoir d'achat se serait dégradé dans un contexte de reprise de l'inflation, comme cela s'observe aussi chez les ménages ne détenant pas de crédit.



En dépit de la dégradation de leur situation budgétaire et financière, les ménages estiment que le poids de leurs charges de remboursement reste supportable.

52,2 % le considèrent comme supportable ou très supportable (51,8 % en 2021 et 56,5 % en 2020)
35,1 % élevé mais supportable (35,7 % en 2021 et 32,2 % en 2020)
12,7 % trop élevé (12,5 % en 2021 et 11,3 % en 2020)

En 2022, le sentiment partagé par 87,3 % des ménages est celui de charges de remboursement supportables : cette proportion se situe au-dessus de sa moyenne de longue période (85,6 %). Et la part des ménages estimant leurs charges trop ou beaucoup trop élevées (12,7 % en 2022) a rarement été aussi faible depuis la fin des années 80.