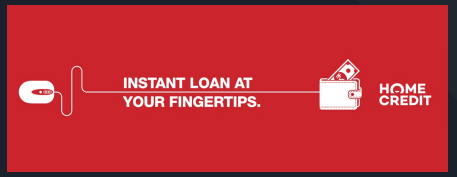
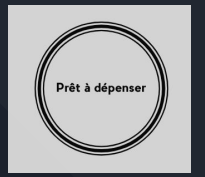




Prêt à dépenser

Objectif : construction d'un modèle de scoring

Katrin-Misel Ponomarjova
OpenClassrooms parcours Ingénieur IA

1. Problématique métier

Algorithme de classification des clients

Prêt à dépenser :

- Propose des crédits à la consommation pour des personnes ayant peu ou pas d'historique de prêt
- L'entreprise souhaite mettre en oeuvre un outil de scoring qui calcule la probabilité qu'un client rembourse le crédit ou pas

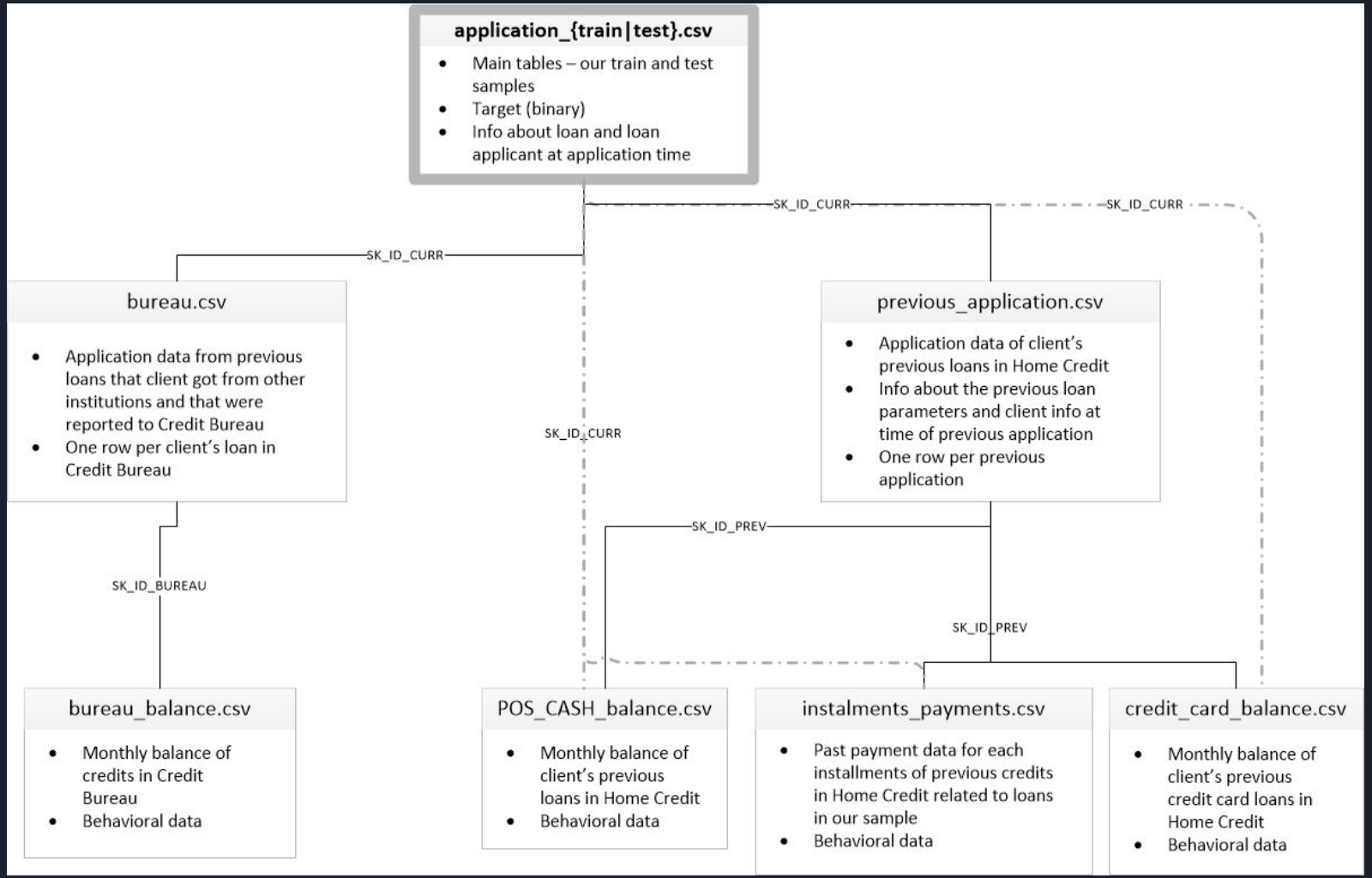
Description de l'outil :

- Un algorithme de classification capable de décider si un prêt peut être accordé à un client
- Le modèle doit être facilement interprétable avec une mesure de l'importance des variables

Problématique

- Coût d’un crédit non remboursé est 10x plus que le coût d’un crédit non accordé à un client qui avait la possibilité de rembourser
- Éviter d’accorder des crédits à des clients qui ne sont pas capables de le rembourser
- Un client refusé peut faire une nouvelle demande de crédit

2. Description du jeu de 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_CURR
SK_ID_PREV
SK_ID_BUREAU
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

TRAIN ET TEST

- Dans le fichier principal, application_train, nous avons :
  - 307 511 prêts
  - 122 features
    - dont TARGET, indiquant si le client a remboursé (TARGET=0) ou pas (TARGET=1)
- Dans le fichier application_test, nous avons :
  - 48 744 prêts et pas de colonne TARGET

```
train.dtypes.value_counts()
```
✓ 0.1s

```
float64    65
int64      41
object     16
dtype: int64
```

```
df_cat.columns
```
✓ 0.1s

```
Index(['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY',
       'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
       'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE',
       'WEEKDAY_APPR_PROCESS_START', 'ORGANIZATION_TYPE', 'FONDKAPREMONT_MODE',
       'HOUSETYPE_MODE', 'WALLSMATERIAL_MODE', 'EMERGENCYSTATE_MODE'],
      dtype='object')
```

Portrait de l’emprunteur moyen

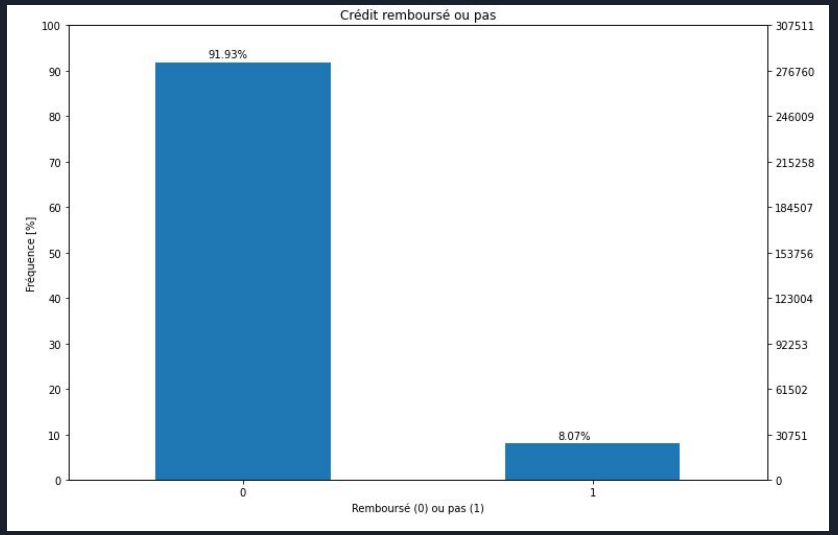
- La plupart des prêts sont des cash loans et sont souscrit par des femmes.
- Ne possèdent pas de voiture mais possèdent de l’immobilier
- Des ouvriers et ne sont pas accompagnés à la signature du prêt
- En moyenne, les clients ont travaillé pendant 6 ans et ont en moyenne 44 ans

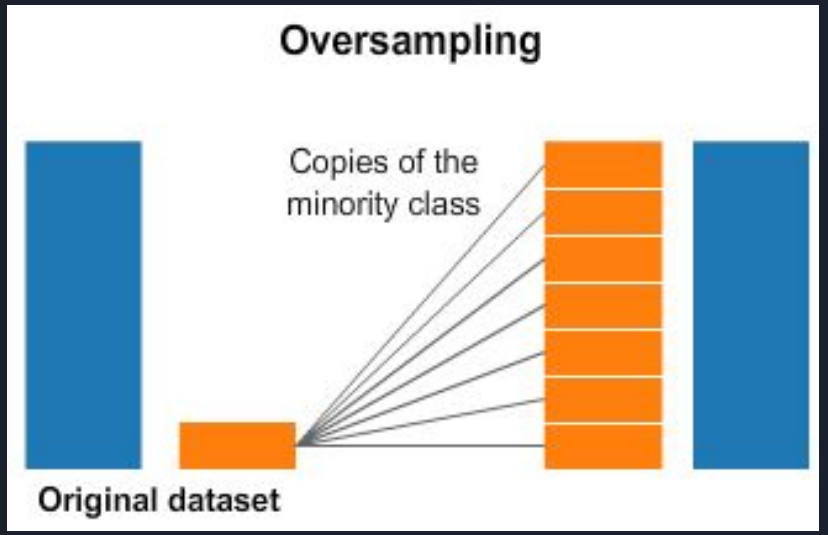
Domain knowledge d’un ancien analyste Home Credit* :

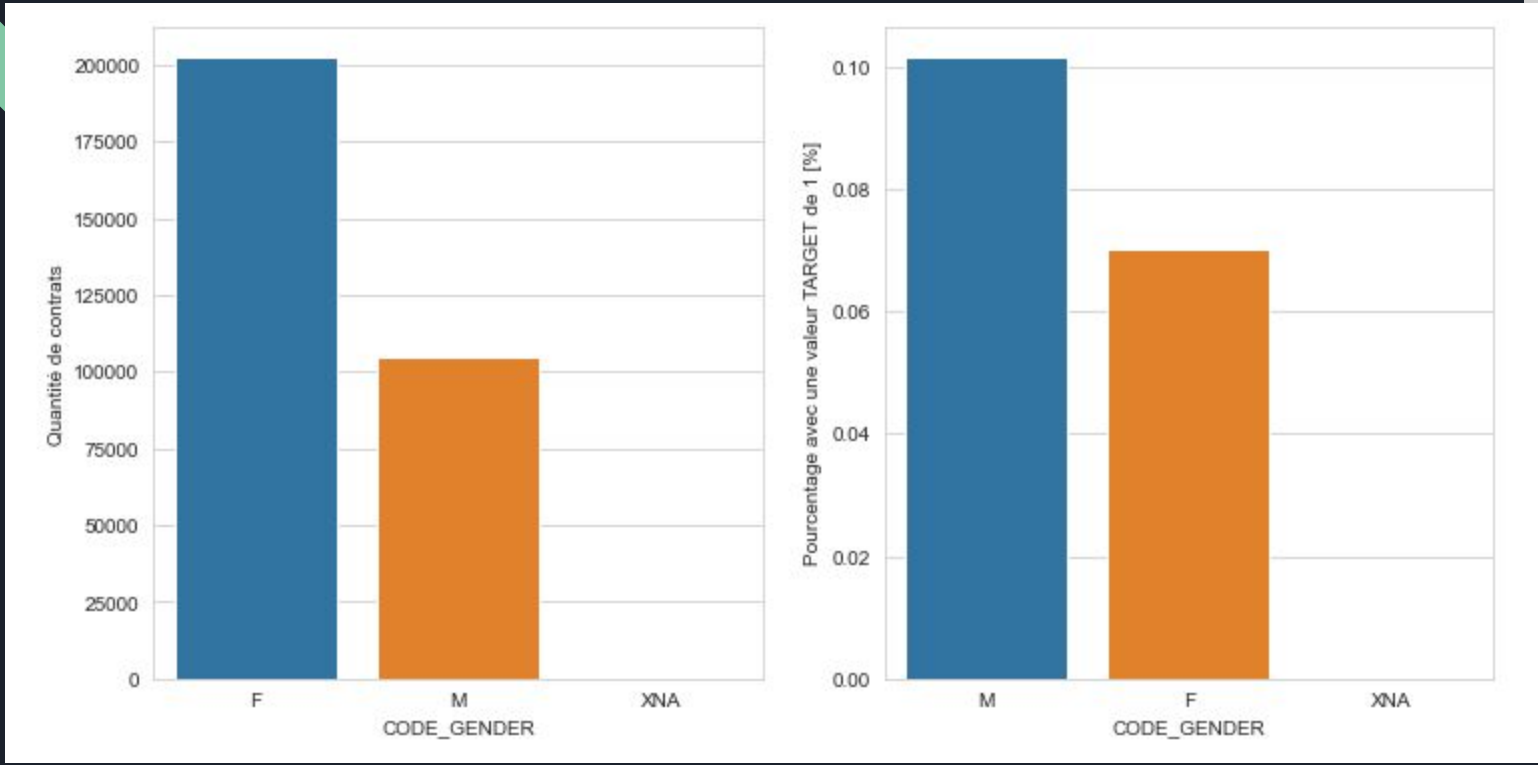
- Les clients HC sont domiciliés en Russie, Vietnam, Chine, Indonésie, etc.
- Très peu d’informations bureau / bancaires
- Current loan et previous loan plus fiables
- Surtout des loans CASH et non pas revolving loan (credit card) populaires aux Etats-Unis

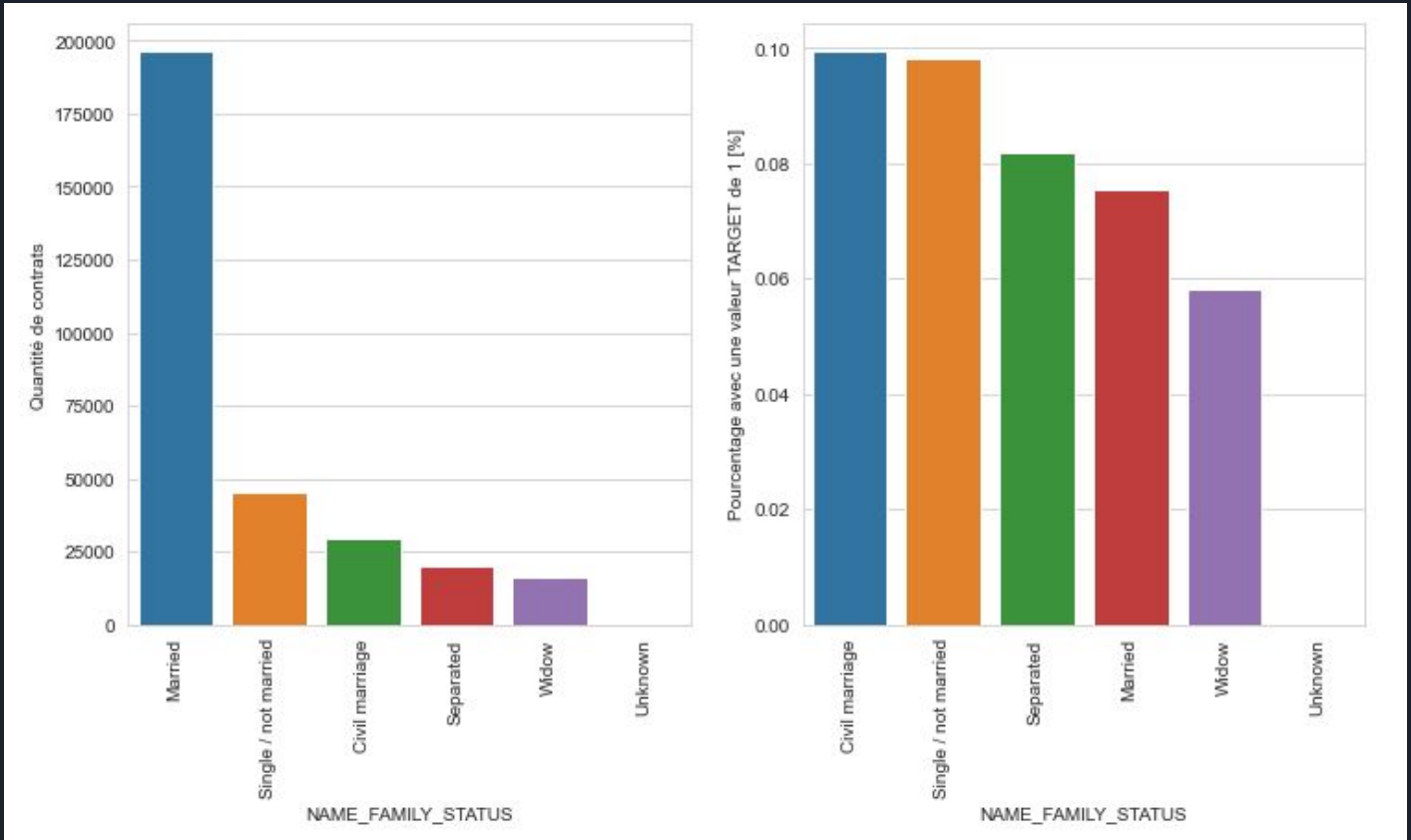
*https://www.kaggle.com/c/home-credit-default-risk/discussion/63032

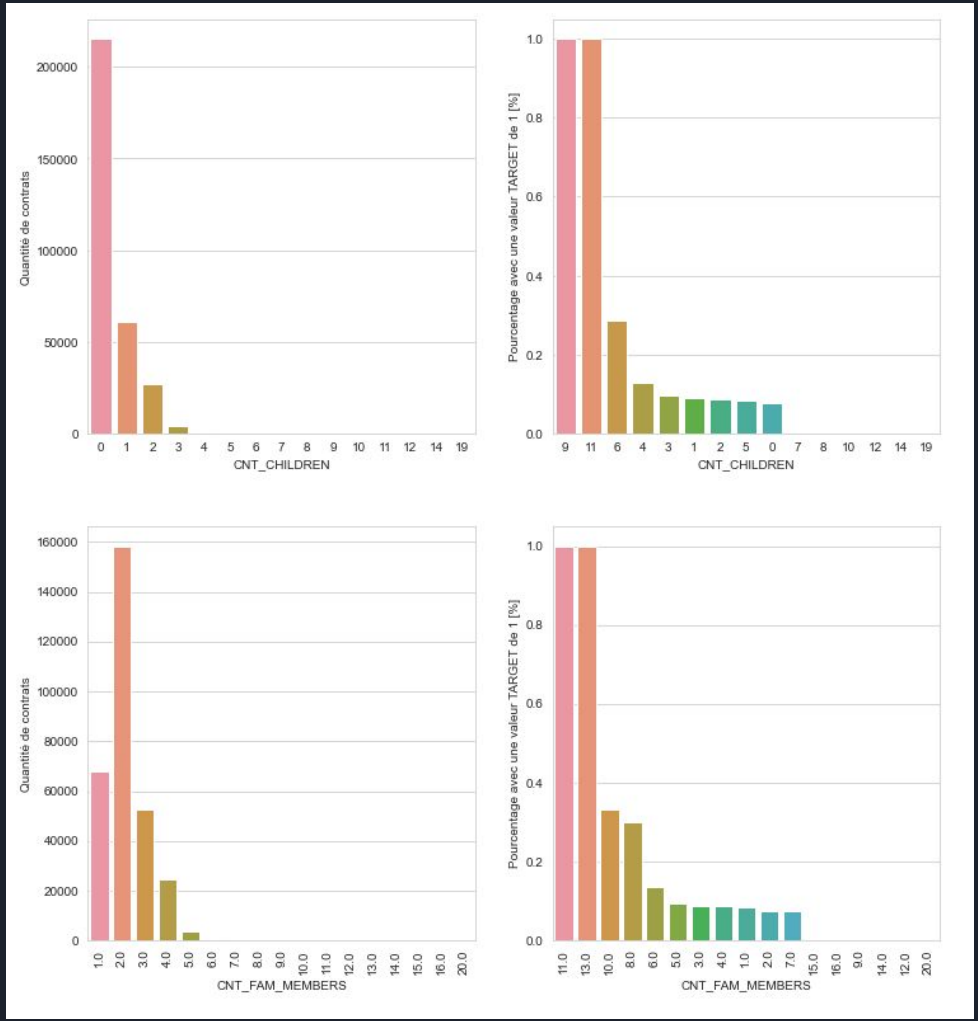
TARGET : classes déséquilibrés et biais

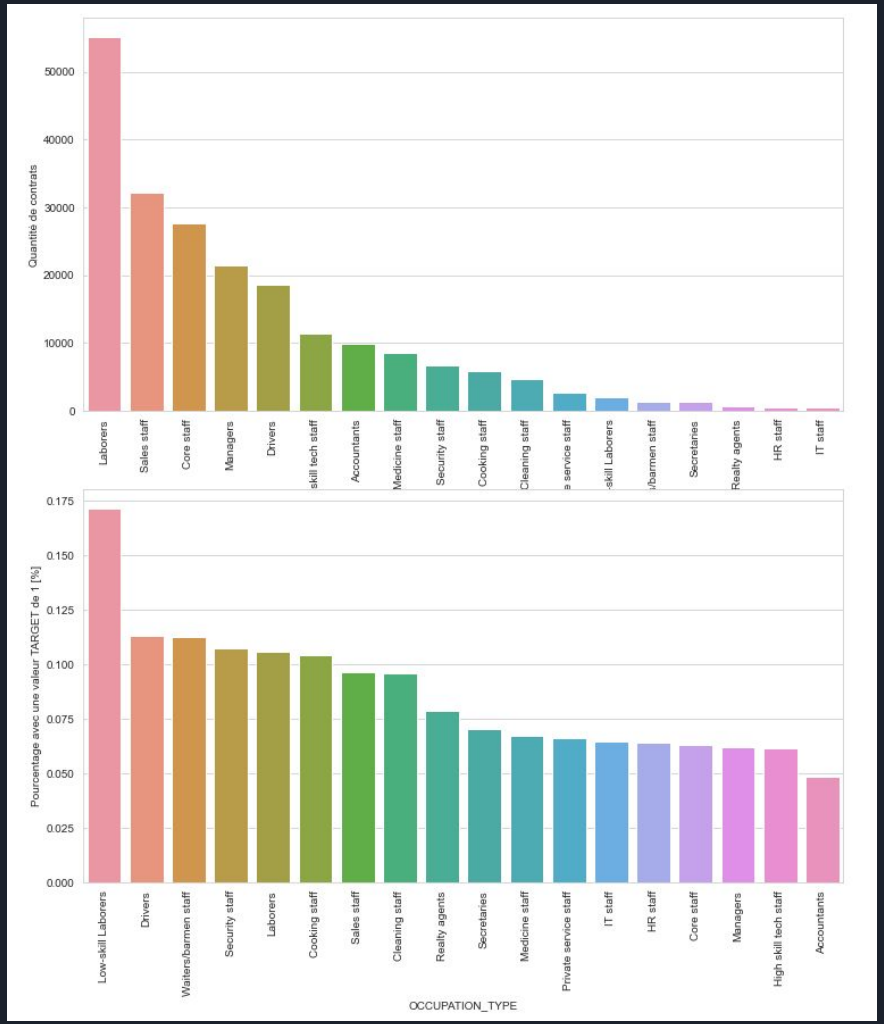





Quantité de contrats
200000
175000
150000
125000
100000
75000
50000
25000
0
F
M
XNA
CODE_GENDER
Pourcentage avec une valeur TARGET de 1 [%]
0.10
0.08
0.06
0.04
0.02
0.00
M
F
XNA
CODE_GENDER

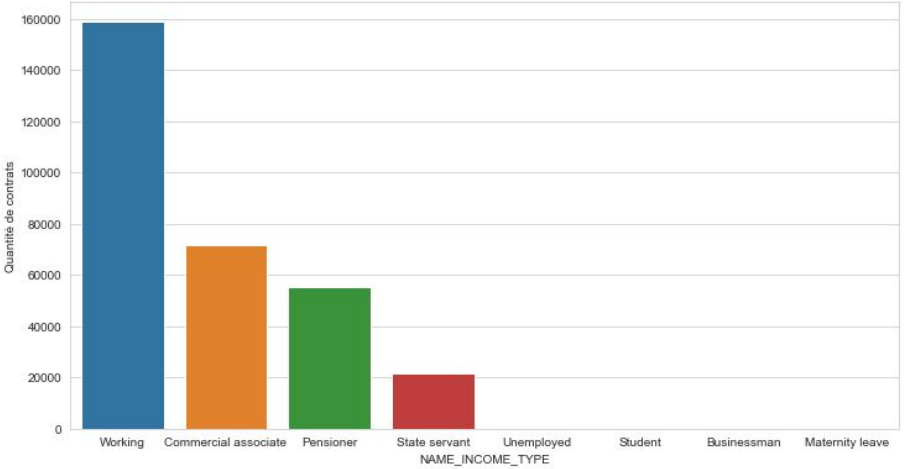

Quantité de contrats
200000
175000
150000
125000
100000
75000
50000
25000
0
Married
Single / not married
Civil marriage
Separated
Widow
Unknown
NAME_FAMILY_STATUS
Pourcentage avec une valeur TARGET de 1 [%]
0.10
0.08
0.06
0.04
0.02
0.00
Civil marriage
Single / not married
Separated
Married
Widow
Unknown
NAME_FAMILY_STATUS

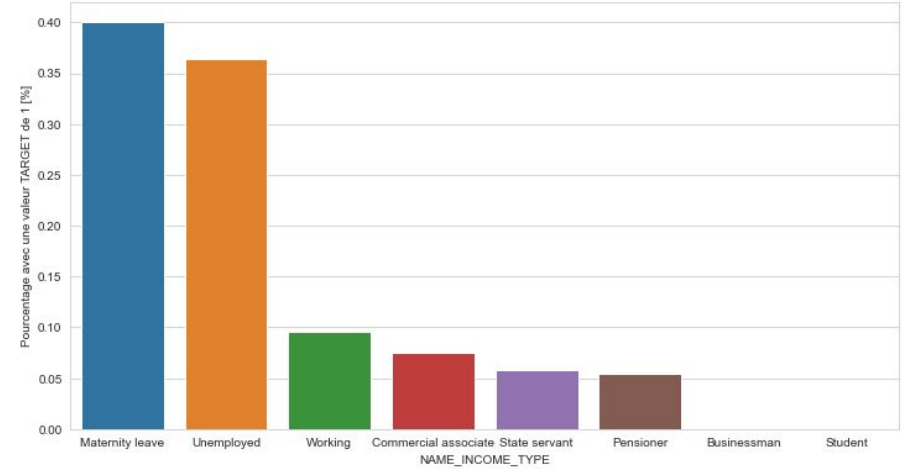

Quantité de contrats
200000
150000
100000
50000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 14 19
CNT_CHILDREN
Pourcentage avec une valeur TARGET de 1 [%]
1.0
0.8
0.6
0.4
0.2
0.0
9 11 6 4 3 1 2 5 0 7 8 10 12 14 19
CNT_CHILDREN
Quantité de contrats
160000
140000
120000
100000
80000
60000
40000
20000
0
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 20.0
CNT_FAM_MEMBERS
Pourcentage avec une valeur TARGET de 1 [%]
1.0
0.8
0.6
0.4
0.2
0.0
11.0 13.0 10.0 8.0 6.0 5.0 3.0 4.0 1.0 2.0 7.0 15.0 16.0 9.0 14.0 12.0 20.0
CNT_FAM_MEMBERS

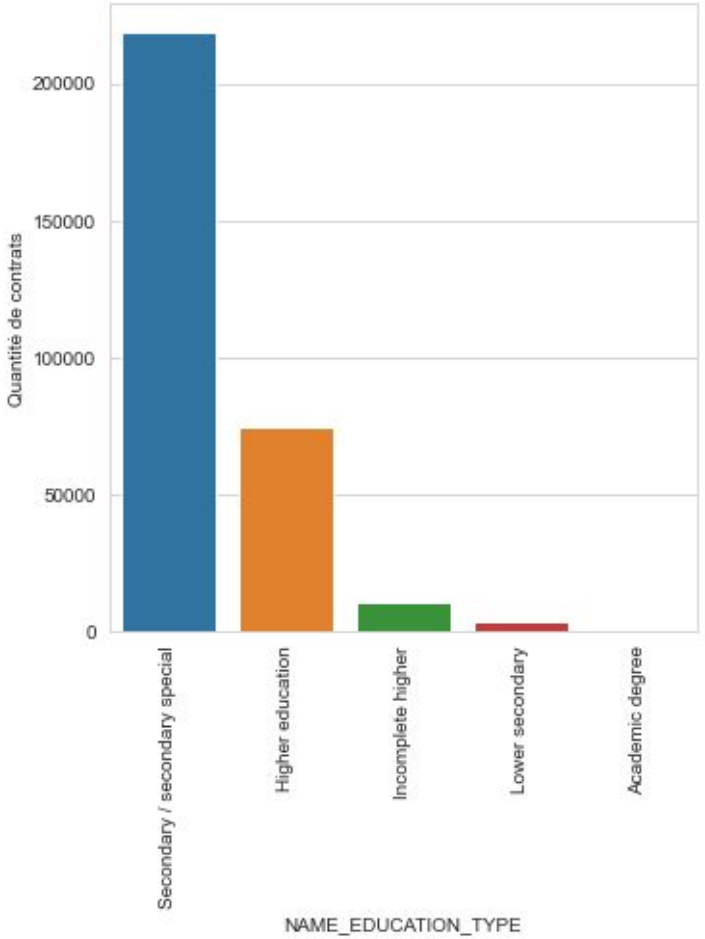


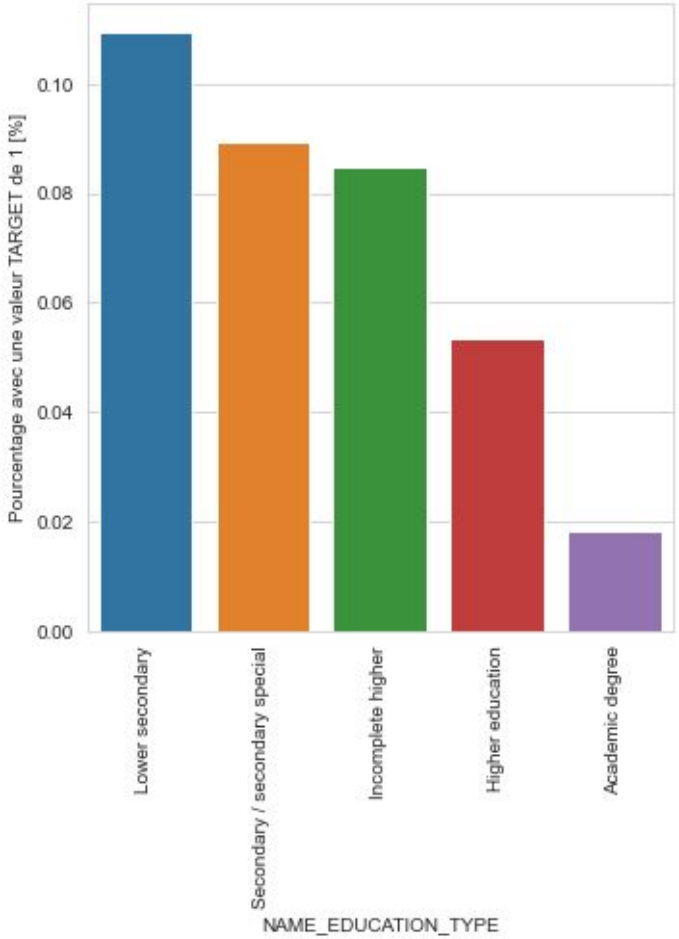
Prêt à dépenser

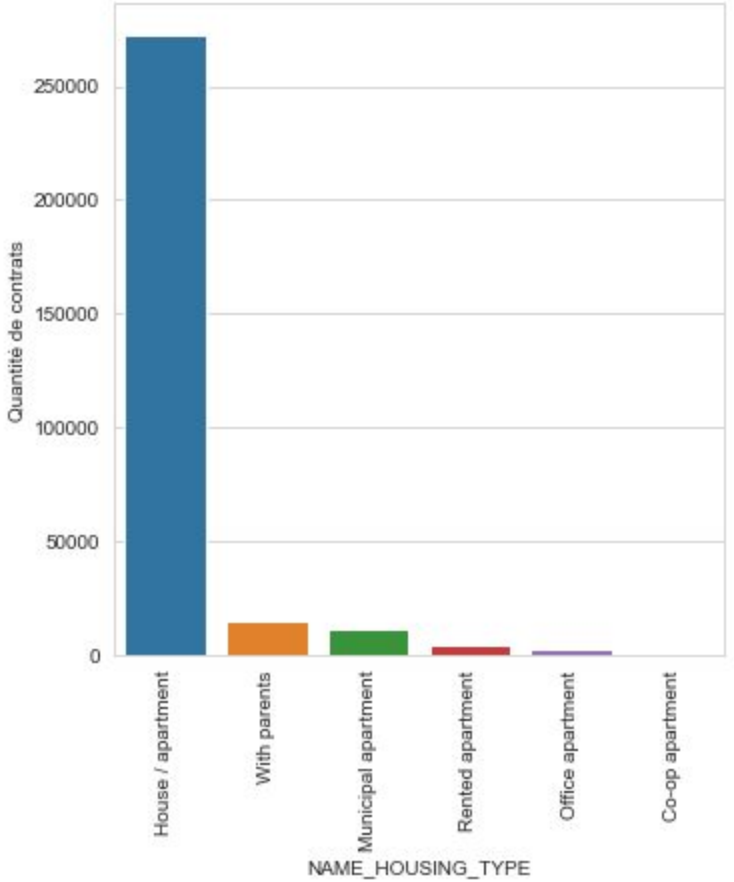
Prêt à dépenser

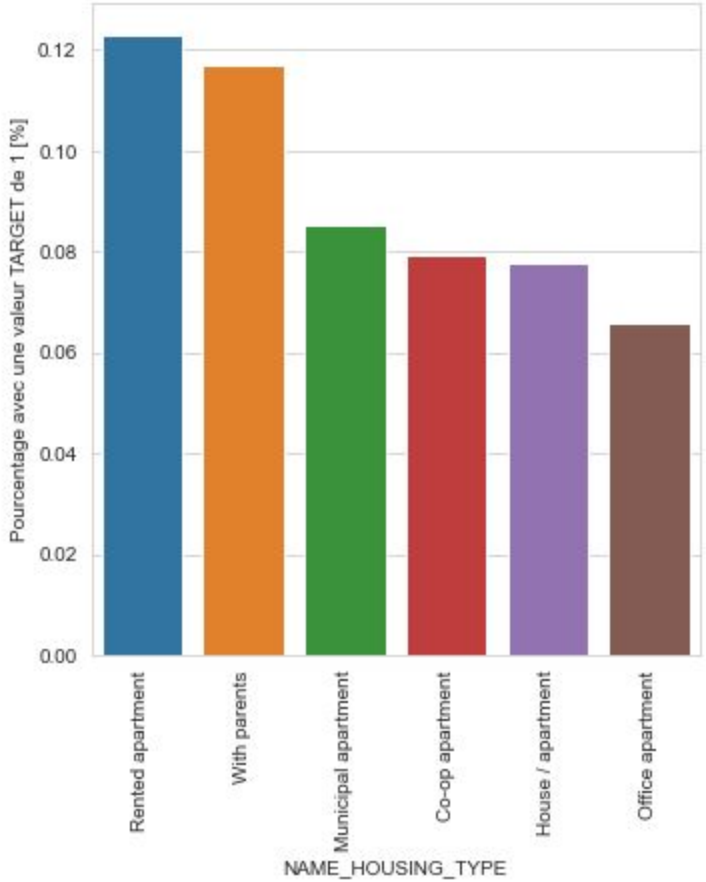




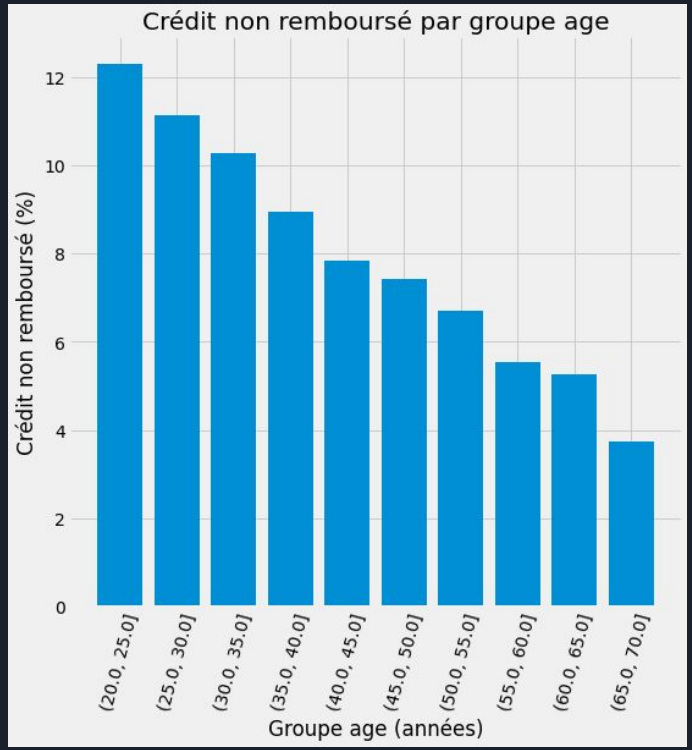

Quantité de contrats
200000
150000
100000
50000
0
Secondary / secondary special
Higher education
Incomplete higher
Lower secondary
Academic degree
NAME_EDUCATION_TYPE

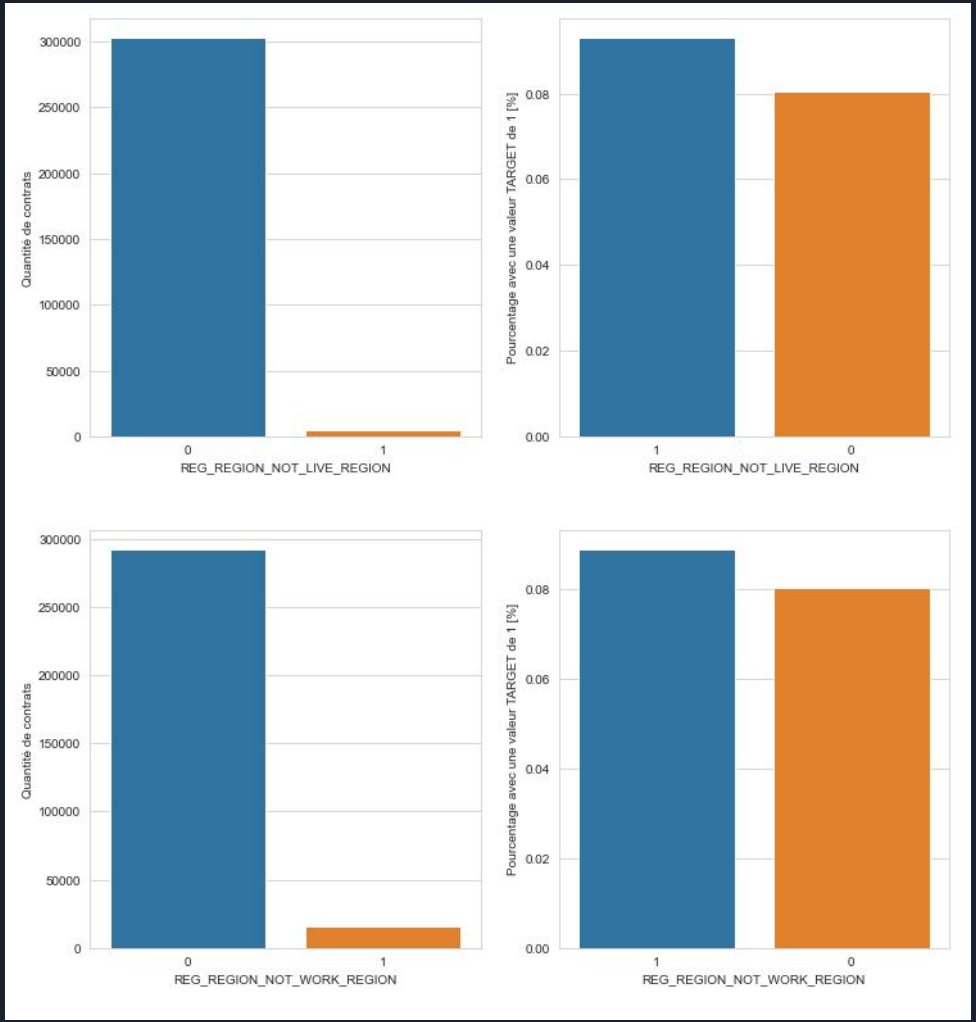

Pourcentage avec une valeur TARGET de 1 [%]
0.10
0.08
0.06
0.04
0.02
0.00
Lower secondary
Secondary / secondary special
Incomplete higher
Higher education
Academic degree
NAME_EDUCATION_TYPE


Quantité de contrats
250000
200000
150000
100000
50000
0
House / apartment
With parents
Municipal apartment
Rented apartment
Office apartment
Co-op apartment
NAME_HOUSING_TYPE


Pourcentage avec une valeur TARGET de 1 [%]
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Rented apartment
With parents
Municipal apartment
Co-op apartment
House / apartment
Office apartment
NAME_HOUSING_TYPE

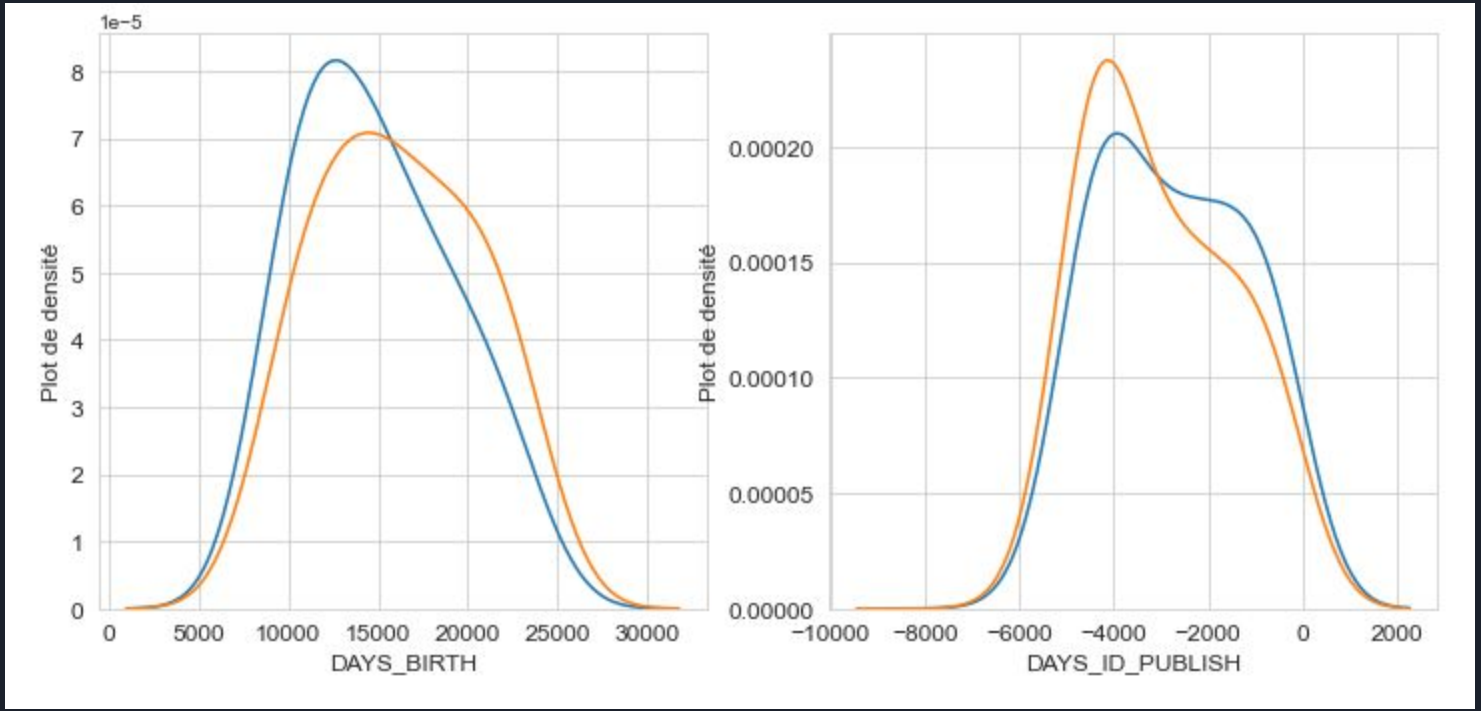

Age du client
Nombre de crédits
17500
15000
12500
10000
7500
5000
2500
0
20
30
40
50
60
70
Age (années)


Crédit non remboursé par groupe age
Crédit non remboursé (%)
12
10
8
6
4
2
0
(20.0, 25.0]
(25.0, 30.0]
(30.0, 35.0]
(35.0, 40.0]
(40.0, 45.0]
(45.0, 50.0]
(50.0, 55.0]
(55.0, 60.0]
(60.0, 65.0]
(65.0, 70.0]
Groupe age (années)

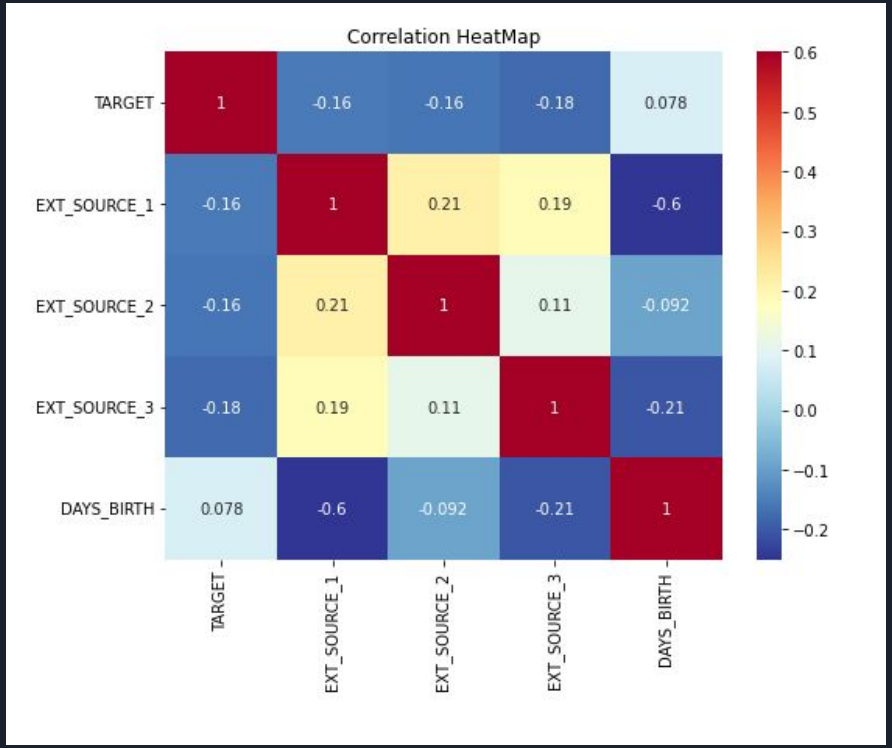


Prêt à dépenser

Prêt à dépenser



Prêt à dépenser



3. Transformation du jeu de données (nettoyage et feature engineering)

Nettoyage

Prêt à dépenser

Gestion des anomalies :

- DAYS_EMPLOYED = 365243
- Créer des flags, remplacer par 0

Supprimer les features avec plus de 60 pourcent valeurs manquantes

Remplacer les valeurs manquantes restantes avec :

- la médiane pour les variables numériques continues
- “Unknown” pour les variables catégorielles

Nettoyage

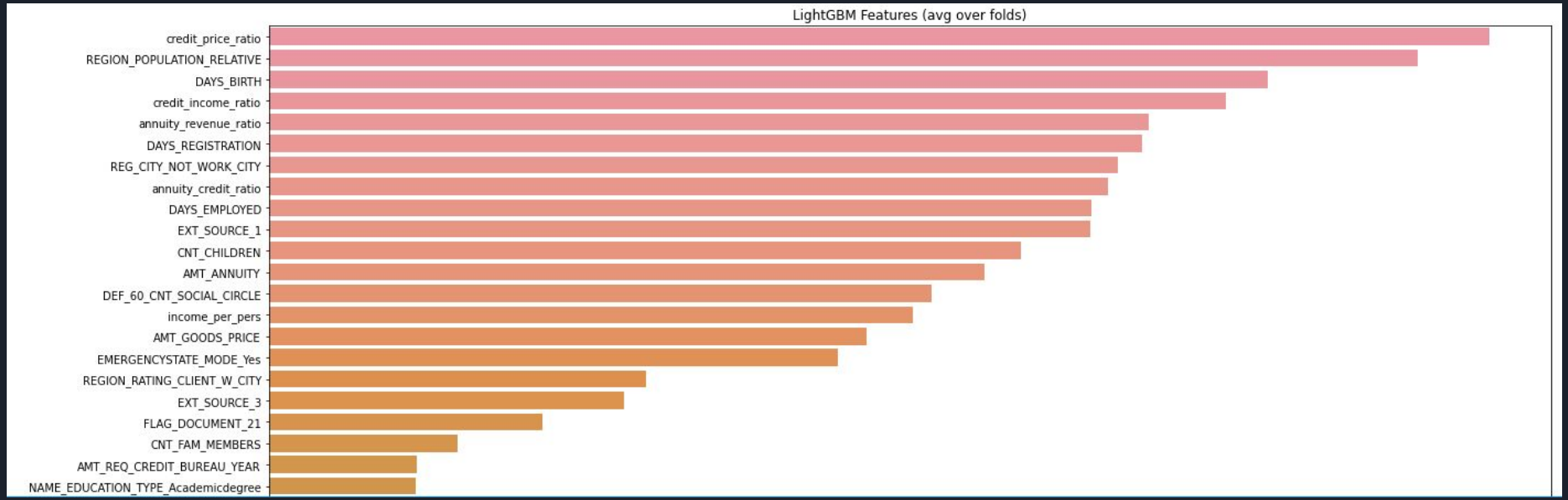
Prêt à dépenser

Encodage des variables catégorielles :

- Label Encoder : pour les variables avec 2 valeurs possibles
- One Hot Encoder : pour les variables avec plusieurs valeurs possibles

Aligner test et train

Sélection des variables



Nouvelles features

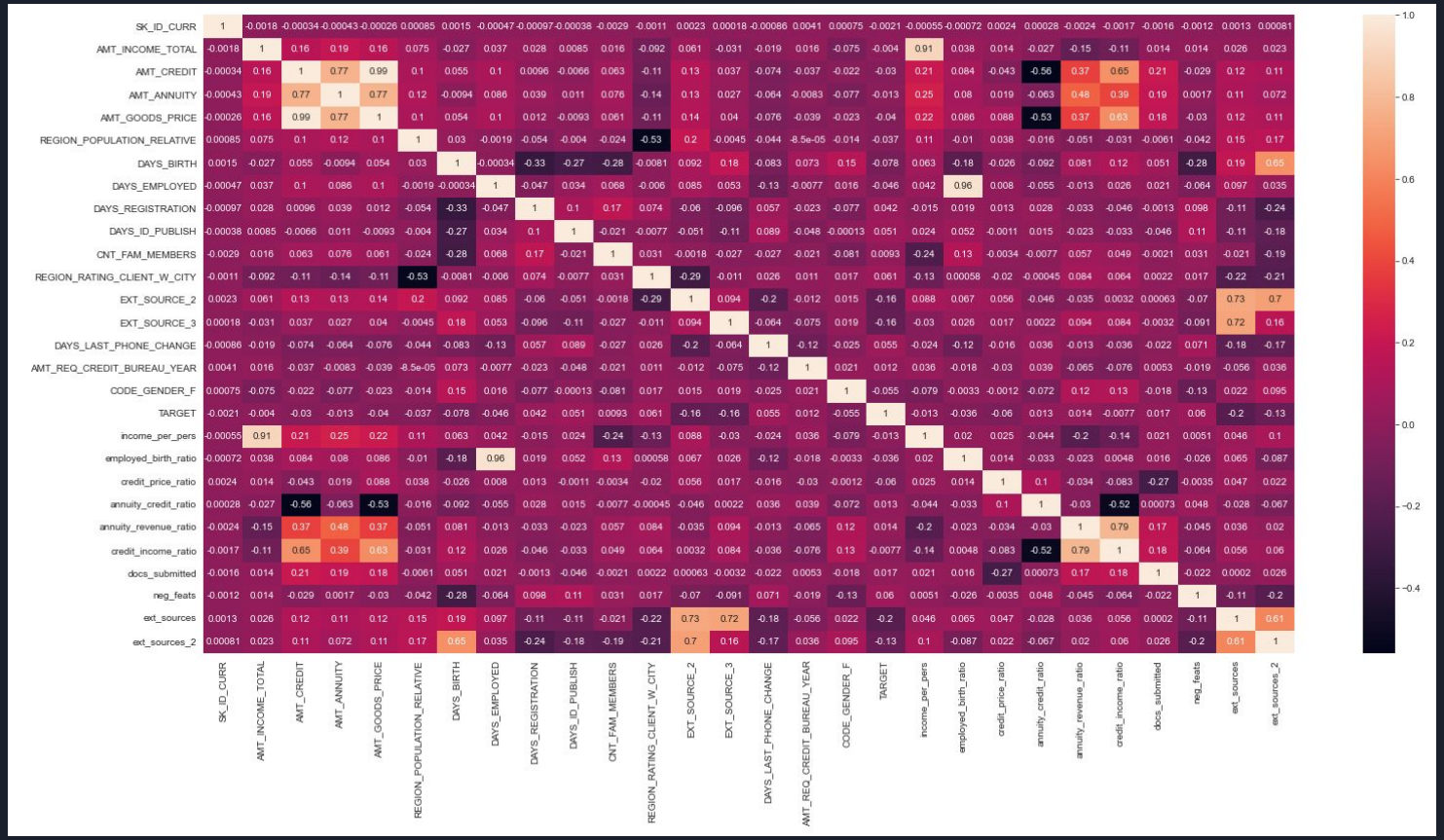
Exemples :

- income_per_person = amt_income_total / cnt_children
- employed_birth_ratio = days_employed / days_birth
- credit_price_ratio = amt_credit / amt_goods_price
- annuity_revenue_ratio = amt_annuity / amt_income_total
- credit_income_ratio = amt_credit / amt_income_total
- docs_submitted
- neg_feats

Features polynôme :

- ext_sources = ext_source_2^2 + ext_source_3^2
- ext_sources_2 = ext_source_2^2 * days_birth^2 + ext_source_3^2

Gérer la colinéarité : 23 features qui nous restent



Métrique personnalisée

Métrique personnalisée :

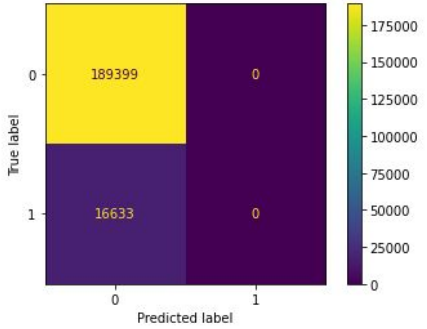
- F2 score à maximiser
- Recall à maximiser au prix de la précision
- Le taux des faux négatifs à minimiser au prix de maximiser les faux positifs

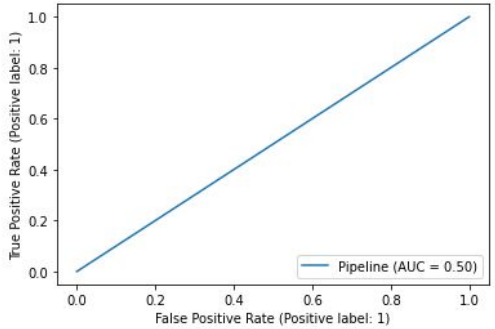
4. Comparaison et synthèse des résultats pour les modèles utilisés

Modèle de référence

- Stratégie : most_frequent
- fbeta_score = `0.0`

	precision	recall	f1-score	support
0	0.92	1.00	0.96	189399
1	0.00	0.00	0.00	16633
accuracy			0.92	206032
macro avg	0.46	0.50	0.48	206032
weighted avg	0.85	0.92	0.88	206032





Comparaison de modèles de classification

Prêt à dépenser

	Model Name	Mean fit time	Mean score time	Mean precision	Mean recall	Mean ROC AUC	Mean F2
0	(SMOTE(), LinearSVC(dual=False, random_state=4...	1.871222	0.031562	0.122744	0.688222	0.668269	0.358181
1	(SMOTE(), DecisionTreeClassifier(random_state=...	3.638697	0.033104	0.128016	0.209135	0.542245	0.185546
2	(SMOTE(), LogisticRegression(random_state=42))	1.128781	0.025532	0.118966	0.677515	0.659022	0.349371
3	(RidgeClassifier(random_state=42))	0.070350	0.020332	0.000000	0.000000	0.726955	0.000000
4	(SMOTE(), RidgeClassifier(random_state=42))	0.389963	0.023731	0.147962	0.528592	0.684429	0.349000
5	(SMOTE(), RandomForestClassifier(n_estimators=...	37.168294	0.643051	0.218891	0.102649	0.692700	0.114829
6	(RandomForestClassifier(class_weight='balanced...	13.857474	0.558079	0.542377	0.008284	0.719331	0.010315
7	(SMOTE(), GradientBoostingClassifier(random_st...	62.150082	0.068528	0.231164	0.086887	0.693583	0.099185
8	(SMOTE(), XGBClassifier(base_score=None, boost...	20.323182	0.083777	0.335223	0.061223	0.703421	0.073163
9	(SMOTE(), LGBMClassifier(objective='binary', r...	1.766468	0.153192	0.347372	0.047687	0.720489	0.057625
10	(LGBMClassifier(class_weight='balanced', objec...	0.775529	0.124667	0.173895	0.602754	0.737634	0.403652

LightGBM Classifier avec Class Weights

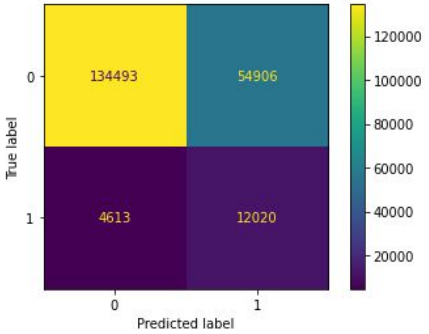
- Sans nos nouvelles features, nous obtenons:
  - LightGBM ROC AUC = 0.7040081671738827
  - Recall = 0.7067877111765767
  - F2 = 0.4358109360518999
- Avec nos nouvelles features :
  - LightGBM ROC AUC = 0.7166815849424928
  - Recall = 0.7239824445379667
  - F2 = 0.45061631379240663

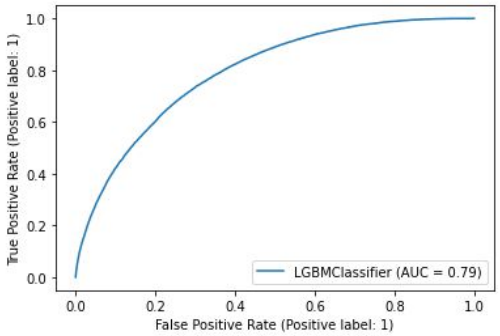
LightGBM avant optimisation sur le jeu de **training**

F2 = 0.45061631379240663

- Faux négatifs : 2.2%
- Faux positifs : 26.6%
- Vrai négatifs : 5.8%
- Vrai positifs : 65.3%

	precision	recall	f1-score	support
0	0.97	0.71	0.82	189399
1	0.18	0.72	0.29	16633
accuracy			0.71	206032
macro avg	0.57	0.72	0.55	206032
weighted avg	0.90	0.71	0.78	206032





Optimisation des hyperparamètres

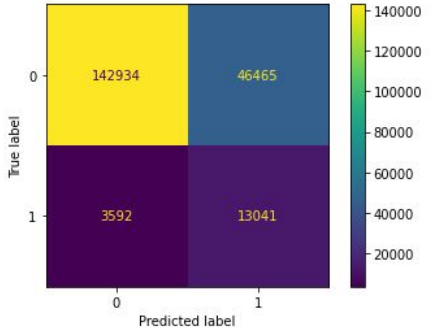
- StratifiedKFold avec 5 folds
- RandomizedSearchCV avec 8 hyperparamètres à tester

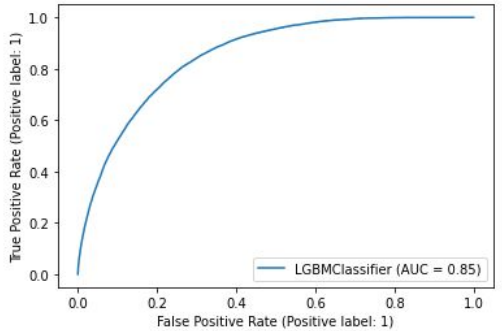
LightGBM après optimisation sur le jeu de **training**

F2 = 0.5173439756263984

- Faux négatifs : 1.7% (avant : 2.2%)
- Faux positifs : 22.5% (avant : 26.6%)
- Vrai négatifs : 6.3% (avant : 5.8%)
- Vrai positifs : 69.4% (avant : 65.3%)

	precision	recall	f1-score	support
0	0.98	0.75	0.85	189399
1	0.22	0.78	0.34	16633
accuracy			0.76	206032
macro avg	0.60	0.77	0.60	206032
weighted avg	0.91	0.76	0.81	206032



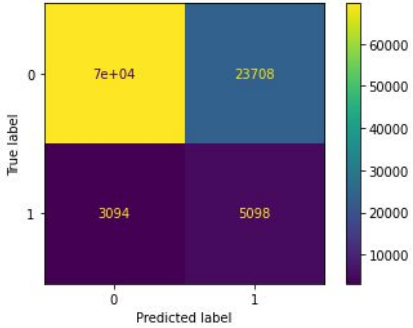


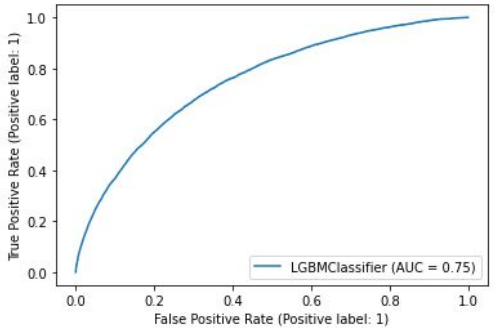
LightGBM après optimisation pour F2 sur le jeu de **testing**

F2 = 0.4139734303439763

- Faux négatifs : 3%
- Faux positifs : 23.4%
- Vrai négatifs : 5%
- Vrai positifs : 69%

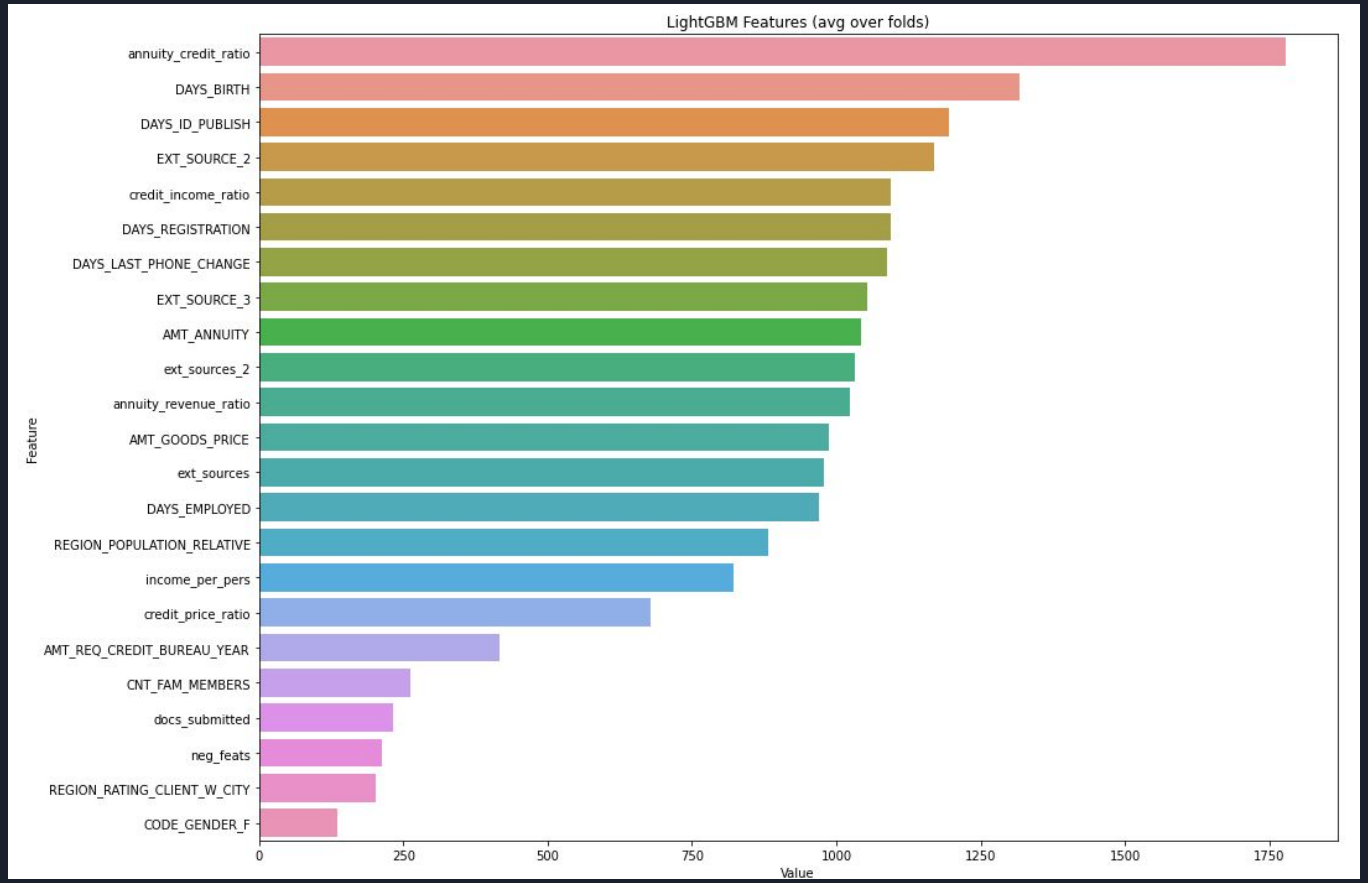
	precision	recall	f1-score	support
0	0.96	0.75	0.84	93287
1	0.18	0.62	0.28	8192
accuracy			0.74	101479
macro avg	0.57	0.68	0.56	101479
weighted avg	0.89	0.74	0.79	101479



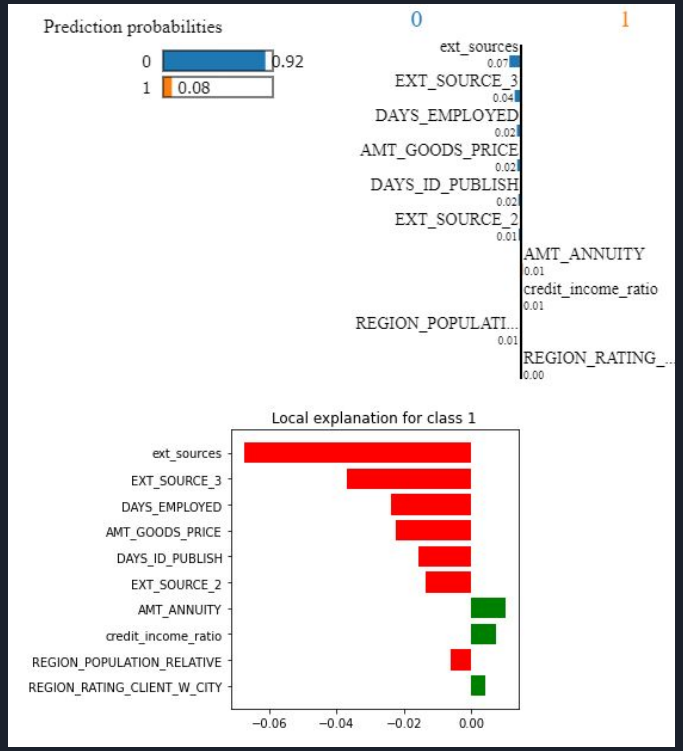


5. Interprétabilité du modèle

Importance des features pour LightGBM

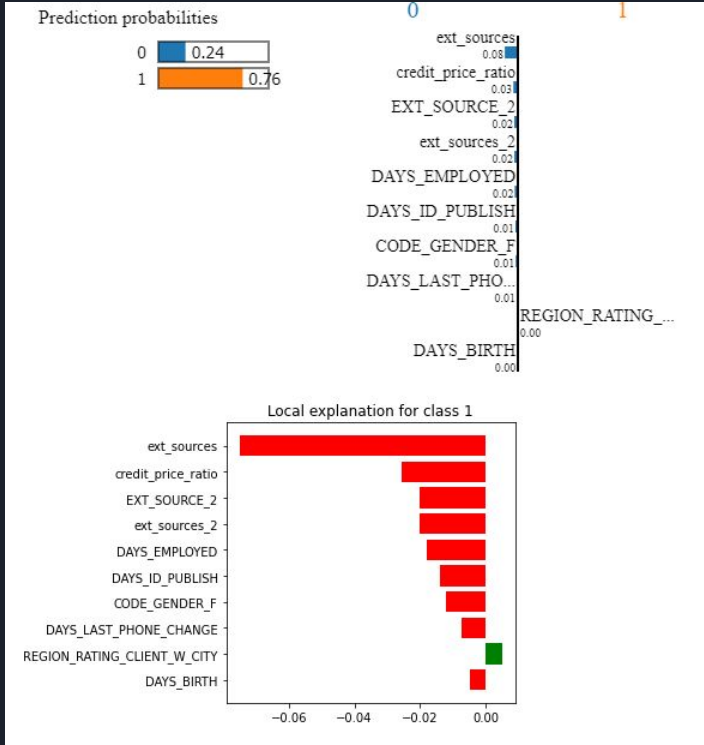


Interprétation avec LIME



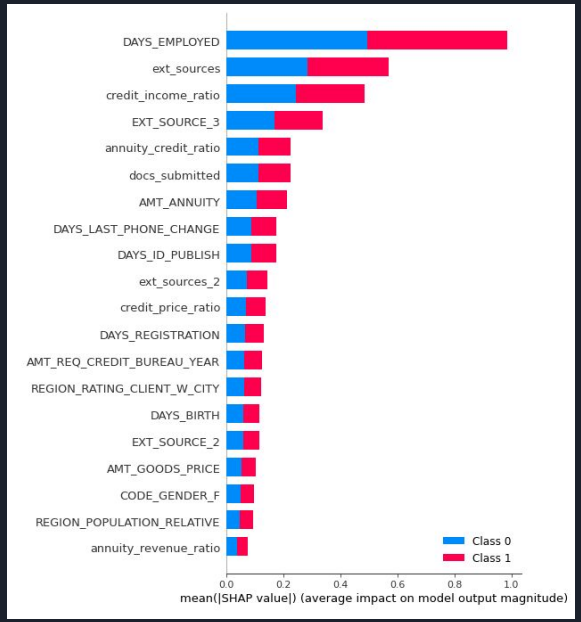
AMT_REQ_CREDIT_BUREAU_YEAR	0.20
CODE_GENDER_F	1.00
income_per_pers	0.01
credit_price_ratio	0.11
annuity_credit_ratio	0.61
annuity_revenue_ratio	0.09
credit_income_ratio	0.03
docs_submitted	0.25
neg_feats	0.17
ext_sources	0.63
ext_sources_2	0.22

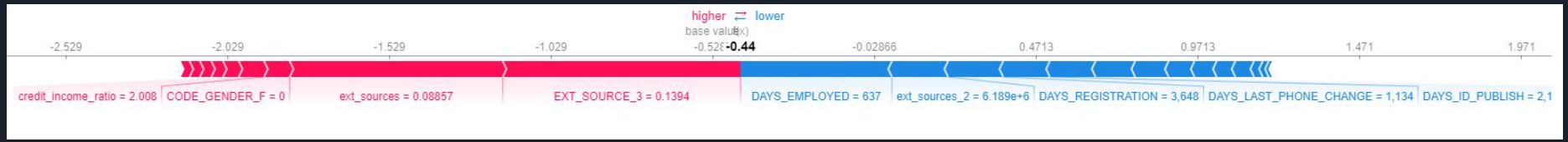
Interprétation avec LIME



Feature	Value
AMT_ANNUITY	0.08
AMT_GOODS_PRICE	0.11
REGION_POPULATION_RELATIVE	0.09
DAYS_BIRTH	0.20
DAYS_EMPLOYED	0.05
DAYS_REGISTRATION	0.13
DAYS_ID_PUBLISH	0.51
CNT_FAM_MEMBERS	0.16
REGION_RATING_CLIENT_W_CITY	1.00
EXT_SOURCE_2	0.19

Interprétation avec SHAP





Recommandations

Possibilités d’amélioration

- Utiliser un autre moyen d’optimisation des hyperparamètres
- Domain knowledge
- Algorithmes d’imputation plus performantes sur les données manquantes