Projet 7

Implémentez un modèle de scoring

par Rouba Yaacoub

Quelle société nous travaillons avec ??

Nous travaillons avec Prêt à dépenser, une société financière qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt.

Notre mission

Suite au données bancaires de 307511 clients, nous allons:

1. Construire **un modèle de scoring** qui donnera une prédiction d'une façon automatique.
2. Construire **un dashboard interactif** permettant l'interprétation des résultats.

TABLE OF CONTENTS

Plan

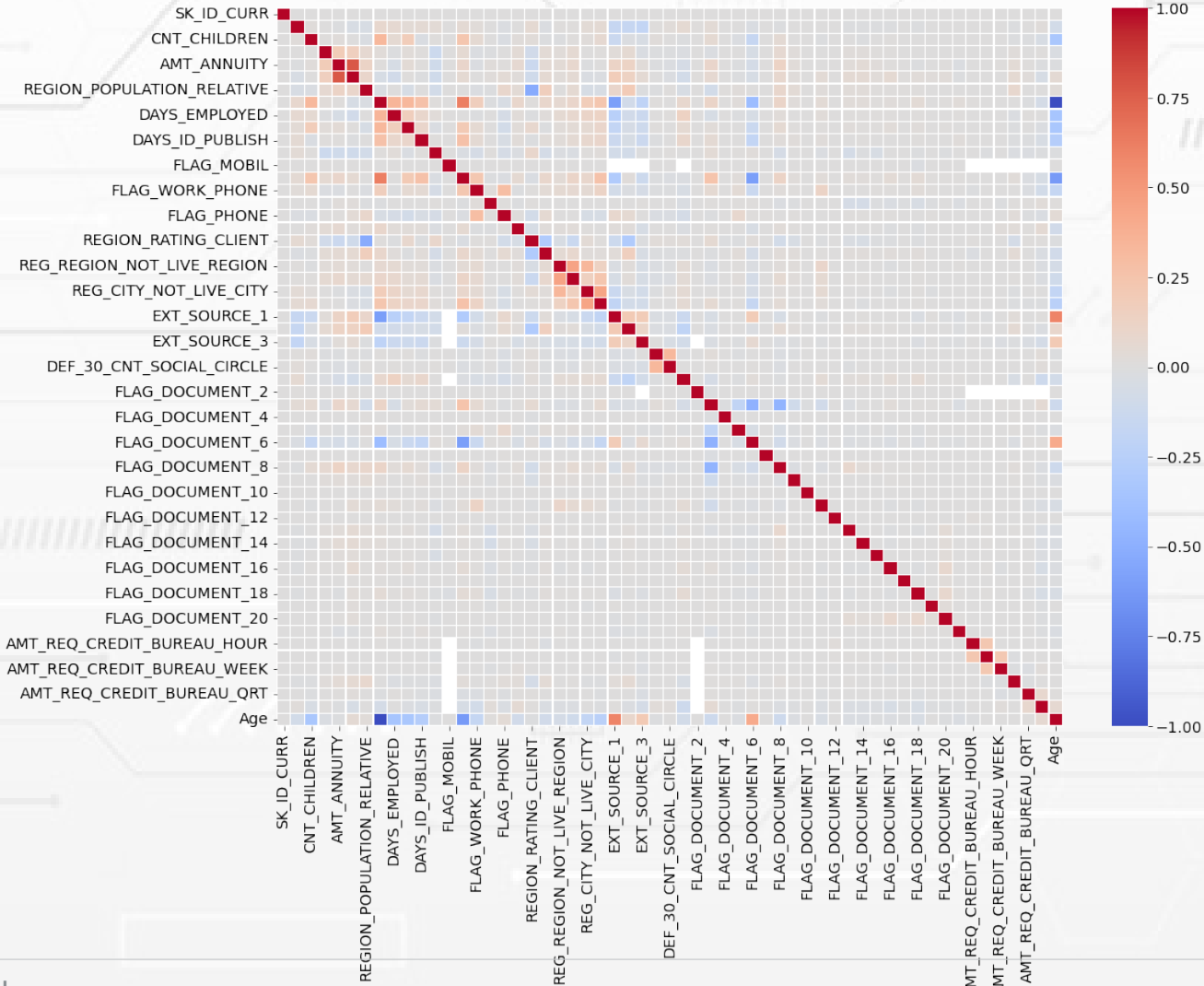
1. Data train/test & classification baseline
2. Ajout de données avec les prêts précédents
3. Amélioration du modèle
4. Dashboard - LIME
5. Conclusion & perspectives

Analyse exploratoire & Nettoyage des données:

Des analyses approfondies ont été effectuées pour nettoyer les deux dataframes, les données ont été nettoyées et sont prêtes pour le traitement.

La correlation

Nous avons implémenté une fonction qui supprime les colonnes corrélées automatiquement.



Variables catégorielle

Traitement des variables catégorielles:

Afin de non pas surchargé nos dataframes, nous utiliserons le **LabelEncoder()** pour les variables catégorielles qui ont plus de deux valeurs différentes.

Et nous utiliserons **get_dummies()** pour celle qui ont deux valeurs differentes ou moins.

Classification baseline:

Nous utilisons la méthode logistique regression pour la classification de notre baseline.

Les scores métriques sont comme suivant:

Score	ROC-AUC	Accuracy	F1	Precision	Recall
Valeur	0.5	0.92	0.02	0.08	0.001

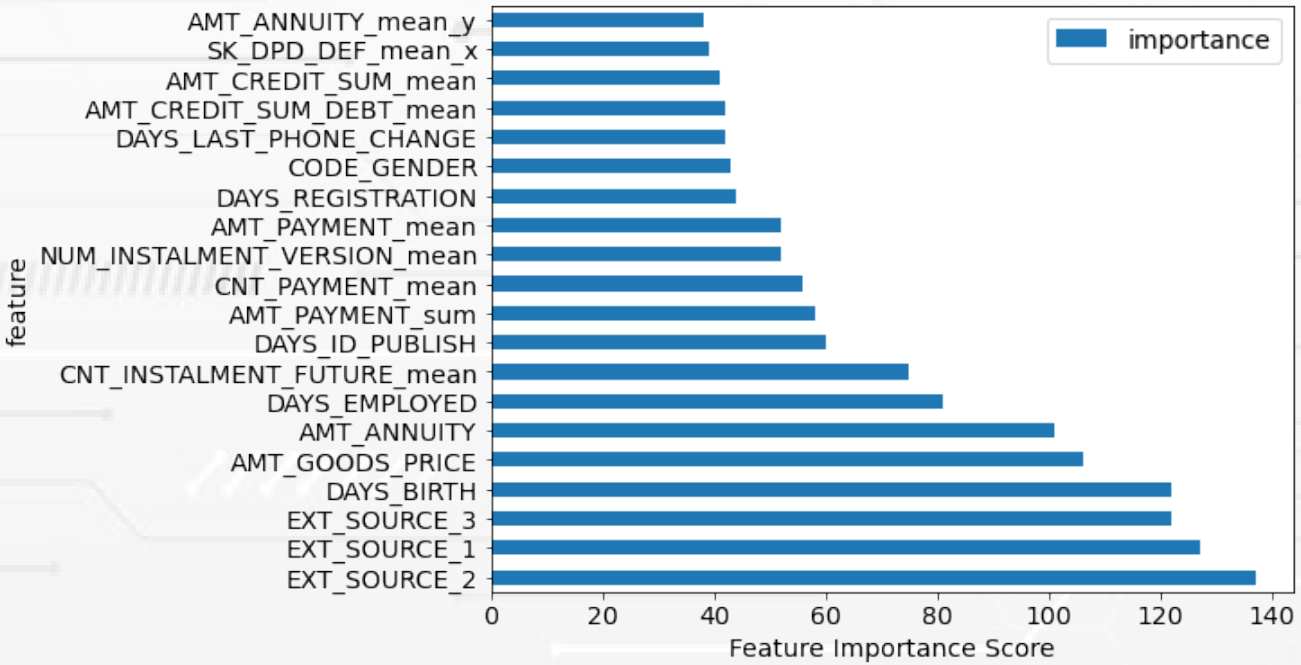
Plan

1. Data train/test & classification baseline
2. **Ajout de données avec les prêts précédents**
3. Amélioration du modèle
4. Dashboard - LIME
5. Conclusion & perspectives

Ajout des data

Afin d'améliorer le score de notre modèle, nous avons décidé d'inclure toutes les données des 6 dataframes restants dans notre data train et test.

Ceci a conduit à un dataframe très chargé avec beaucoup de données, d'où l'obligation d'étudier et d'évaluer l'importance de chaque variable en utilisant **Light Gradient Boosting Machine (Light-GBM)**.



Plan

1. Data train/test & classification baseline
2. Ajout de données avec les prêts précédents
3. **Amélioration du modèle**
4. Dashboard - LIME
5. Conclusion & perspectives

Cross validation de fit_params

Nous utilisons la méthode Light-GBM pour la classification

In [6]:
```
fit_params={"early_stopping_rounds":30,
            "eval_metric" : 'auc',
            "eval_set" : [(X_test,y_test)],
            'eval_names': ['valid'],
            #'callbacks': [lgb.reset_parameter(learning_rate=learning_rate_010_decay_power_099)],
            'verbose': 100,
            'categorical_feature': 'auto'}
```

In [7]:
```
param_test ={'num_leaves': sp_randint(6, 50),
             'min_child_samples': sp_randint(100, 500),
             'min_child_weight': [1e-5, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3, 1e4],
             'subsample': sp_uniform(loc=0.2, scale=0.8),
             'colsample_bytree': sp_uniform(loc=0.4, scale=0.6),
             'reg_alpha': [0, 1e-1, 1, 2, 5, 7, 10, 50, 100],
             'reg_lambda': [0, 1e-1, 1, 5, 10, 20, 50, 100]}
```

In [8]:
```
n_HP_points_to_test = 100
clf = lgb.LGBMClassifier(max_depth=-1, random_state=314, silent=True, metric='None', n_jobs=4, n_estimators=5000)
gs = RandomizedSearchCV(
    estimator=clf, param_distributions=param_test,
    n_iter=n_HP_points_to_test,
    scoring='roc_auc',
    cv=3,
    refit=True,
    random_state=314,
    verbose=True)
```

Cross validation de fit_params

Nous utilisons la méthode Light-GBM pour la classification

In [12]:
```
gs_sample_weight = GridSearchCV(estimator=clf_sw,
                                param_grid={'scale_pos_weight':[1,2,6,12]},
                                scoring='roc_auc',
                                cv=5,
                                refit=True,
                                verbose=True)
```

In [14]:
```
print("AUC score  :   Parameters")
for i in np.argsort(gs_sample_weight.cv_results_['mean_test_score'])[-5:]:
    print('{1:.4f}     :  {0}'.format(gs_sample_weight.cv_results_['params'][i],
                                      gs_sample_weight.cv_results_['mean_test_score'][i]))
```

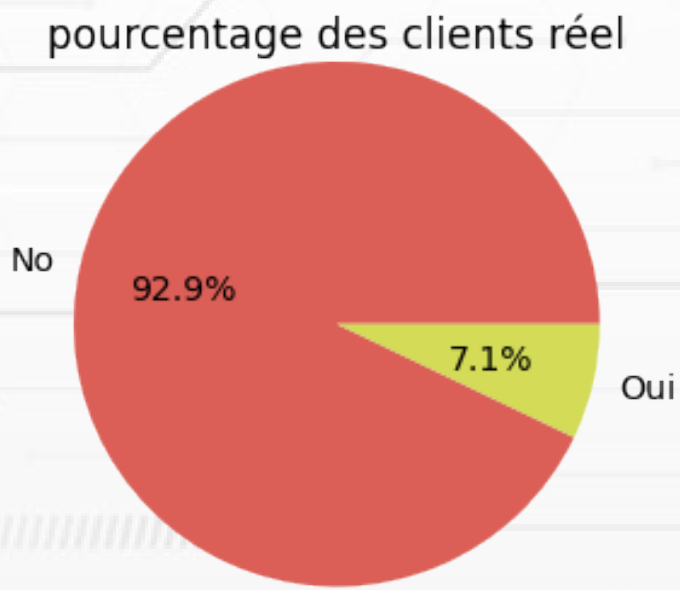
```
AUC score  :   Parameters
0.7795     :  {'scale_pos_weight': 6}
0.7797     :  {'scale_pos_weight': 1}
0.7800     :  {'scale_pos_weight': 12}
0.7801     :  {'scale_pos_weight': 2}
```

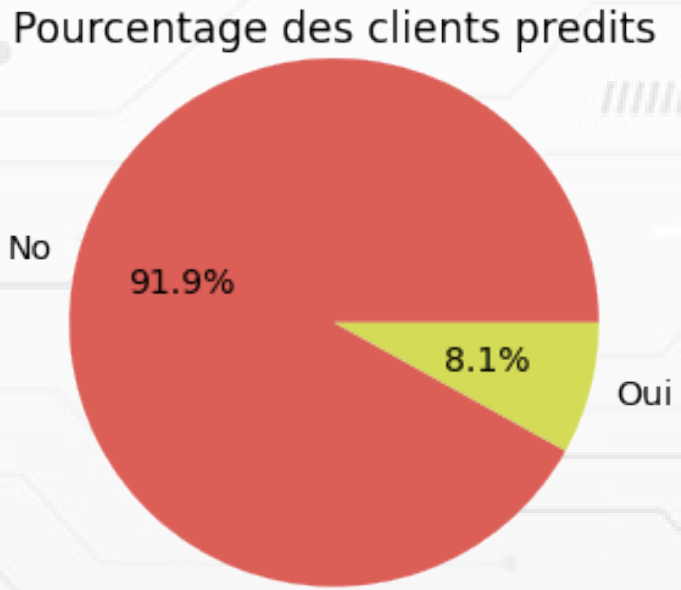
Résultats du modèle

Les scores métriques de notre nouvelle méthode avec **Light-GBM** sont comme suivant:

Score	ROC-AUC	Accuracy	F1	Precision	Recall
Valeur	0.5	0.93	0.95	0.92	0.99

Résultats du modèle





Le scoring

- FN ➔ Perte d'argent pour la banque : **- 100**
- TP ➔ Refus de prêt, la banque ne perd pas d'argent : **+10**
- TN ➔ Prêt accordé, gain d'argent pour la banque : **+ 10**
- FP ➔ Client potentiel perdu, perte d'argent pour la banque : **- 1**

$Tot = FN*-100+TP*10+TN*10+FP*-1$

$gain_{max} = (TN+FP)*10 + (TP+FN)*0$

$gain_{min} = (TN+FP)*-1 + (TP+FN)*-100$

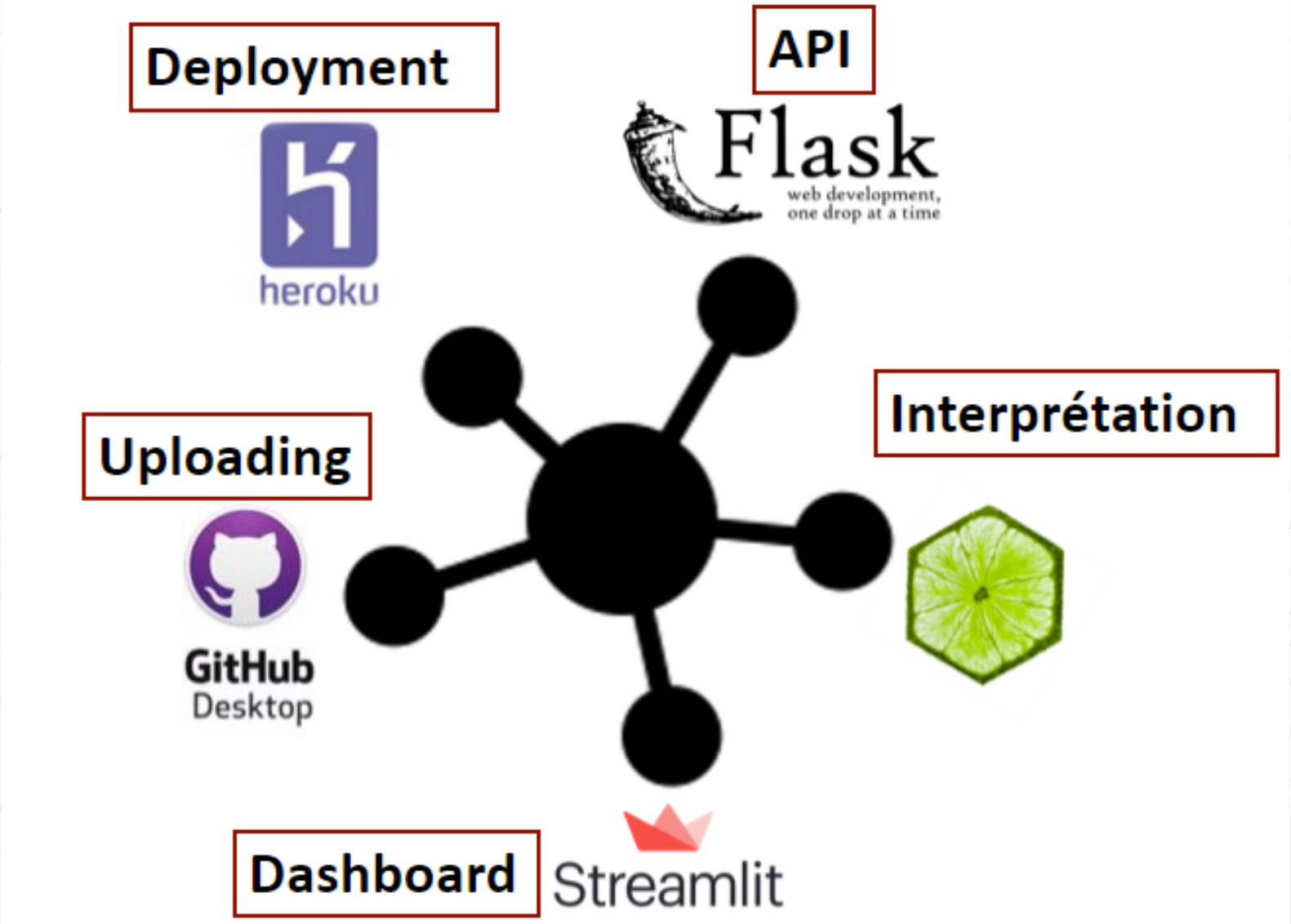
$$gain = \frac{(Tot - gain_{min})}{(gain_{Max} - gain_{min})}$$

gain = **91%**

		Actual Value: Present	Actual Value: Absent
Predicted Value: Present		TP	FP
Predicted Value: Absent		FN	TN

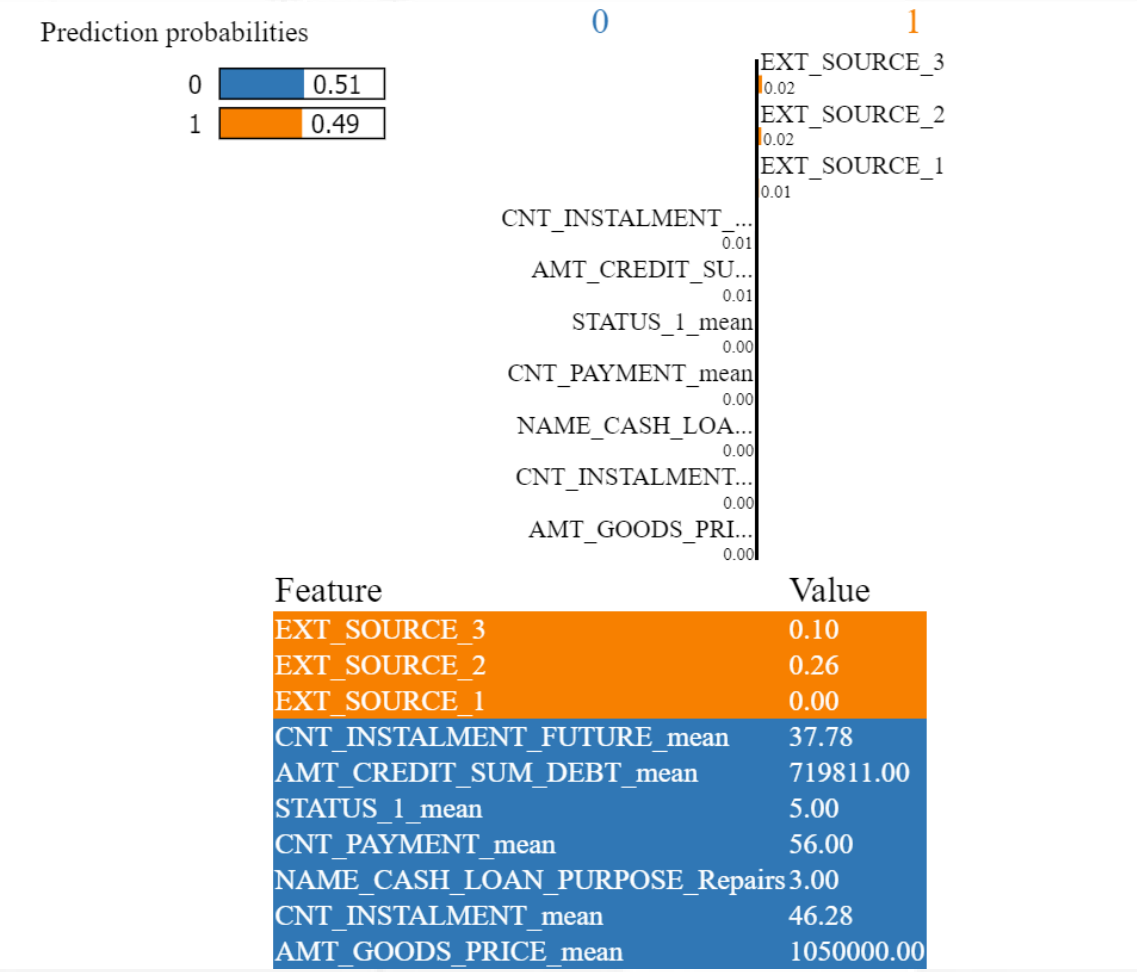
Plan

Dashboard - LIME



Introduction du LIME:

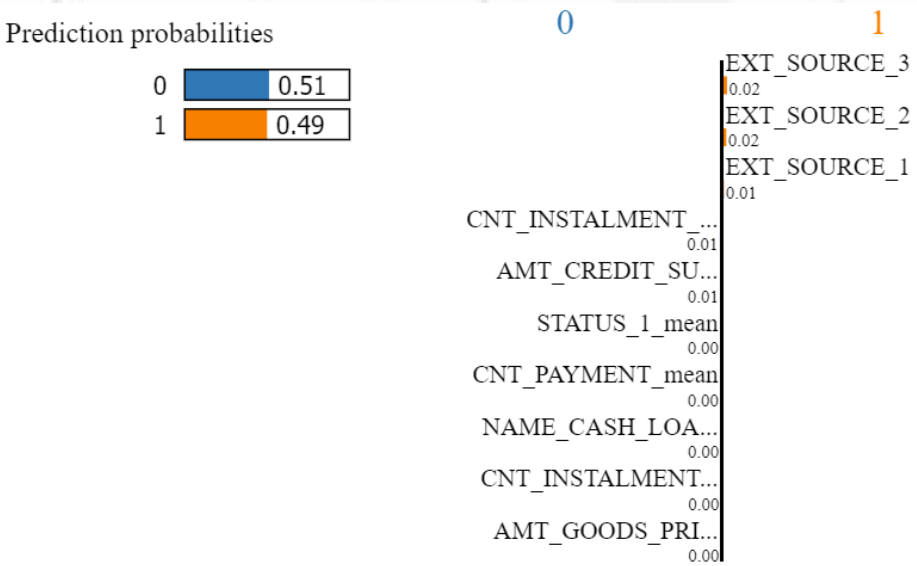
Lime permet d'observer l'évolution des prédictions en fonction de l'importance de certains critères. Grâce à Lime nous pouvons expliquer facilement les raisons pour lesquels chaque individu a été scoré ou pas.



Profile individuelle

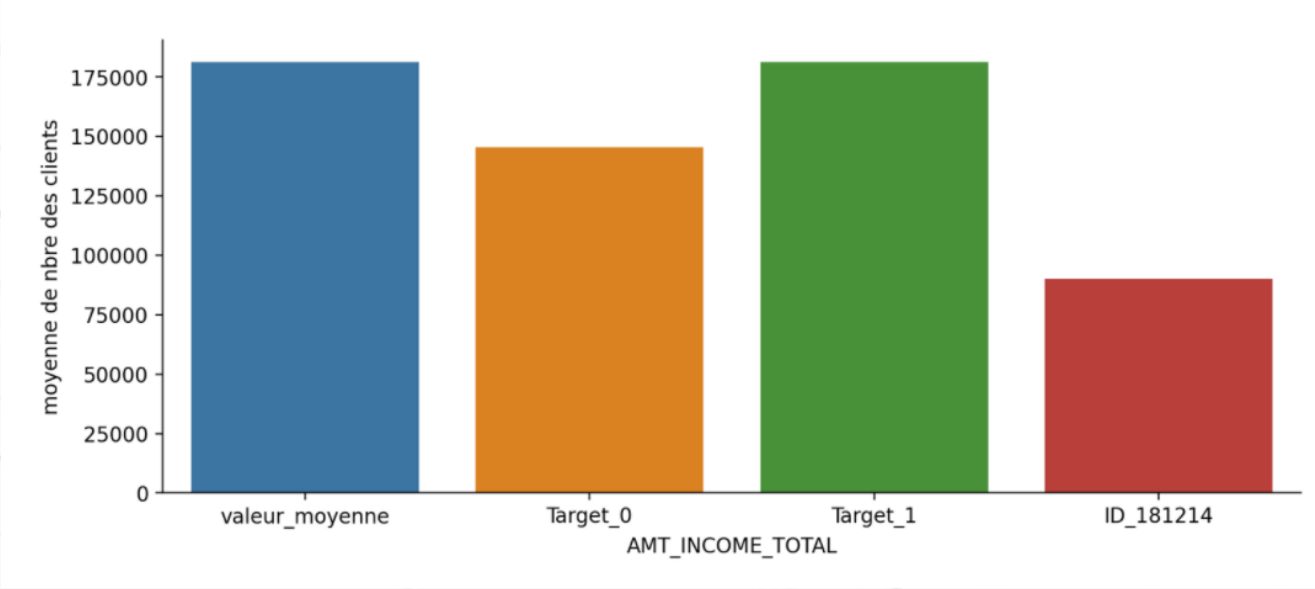
Profil client 181214

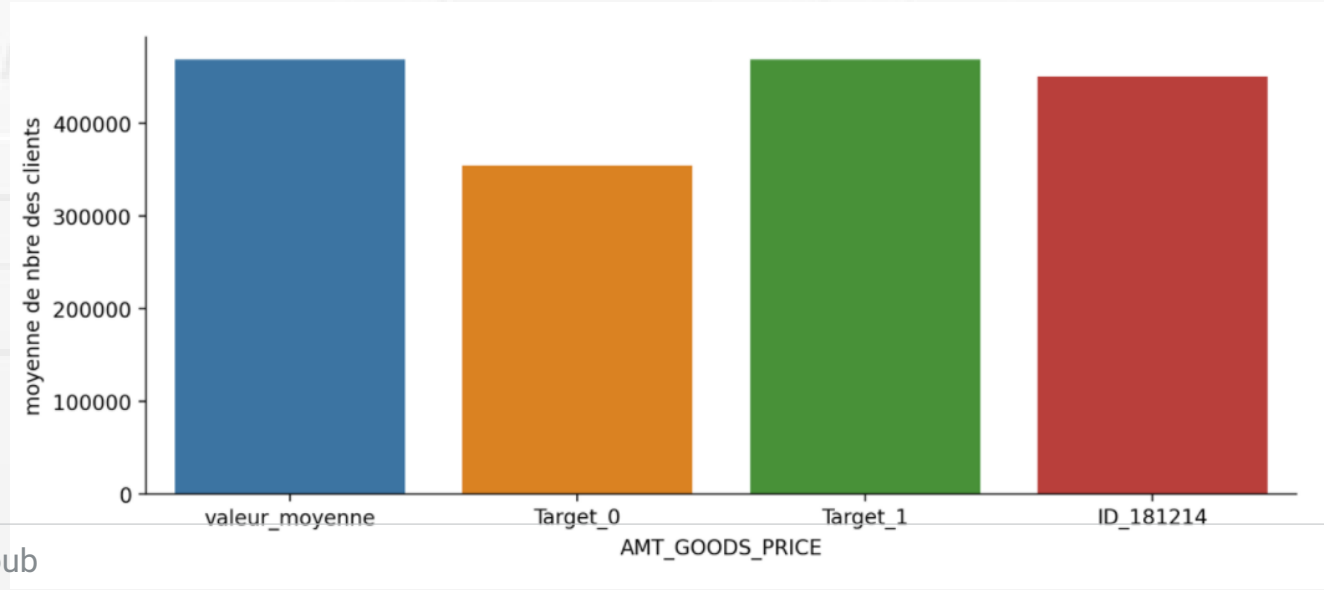
	AGE	GENRE	ENFANT	CODE_REGION
181214	53	F	0	2



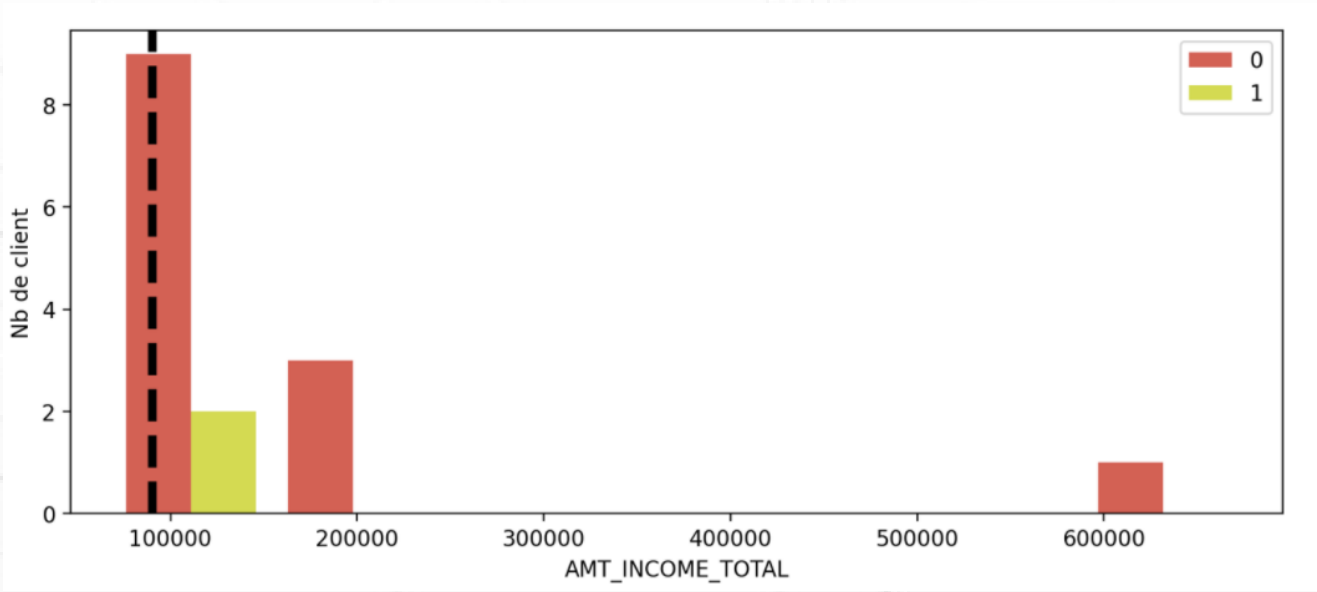
Feature	Value
EXT_SOURCE_3	0.10
EXT_SOURCE_2	0.26
EXT_SOURCE_1	0.00
CNT_INSTALMENT_FUTURE_mean	37.78
AMT_CREDIT_SUM_DEBT_mean	719811.00
STATUS_1_mean	5.00
CNT_PAYMENT_mean	56.00
NAME_CASH_LOAN_PURPOSE_Repairs	3.00
CNT_INSTALMENT_mean	46.28
AMT_GOODS_PRICE_mean	1050000.00

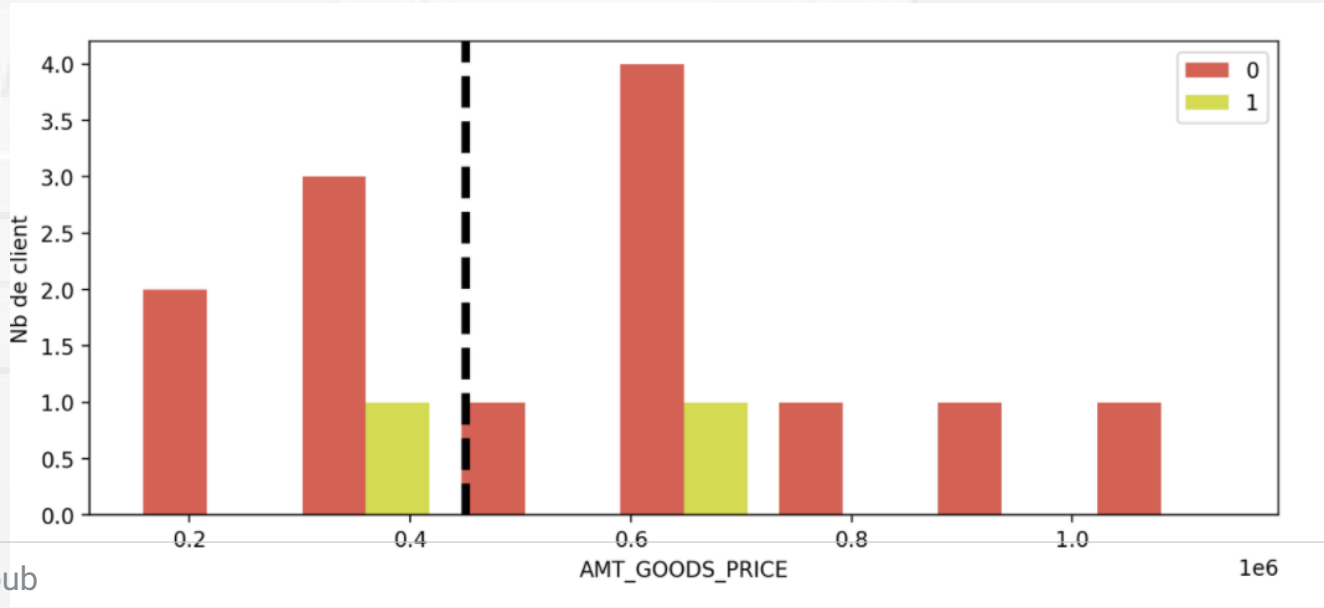
Profile globale





Profile similaires





Plan

Conclusion & perspectives

Dans un premier temps, nous avons:

- introduit un modèle qui met en place une métrique bancaire.
- modélisé la probabilité de solvabilité d'un client à un prêt de consommation.
- crée un dashboard interactif.

Dans un deuxième temps, nous pouvons:

- introduire un autre modèle d'échantillonnage sur les données déséquilibrés (Over-Sampling ou/et Under-sampling).
- améliorer les fonctions de gain présenté.
- améliorer les Dashboard selon la demande de la société.