Note Méthodologique

Projet 7

Anantharajah Anojan

Table des matières :

Rappel du contexte :

« Prêt à dépenser » est une société française proposant des prêts à la consommation ayant peu ou pas d'historique de prêt.

Pour faciliter l'expérience de la société, ils souhaitent développer un modèle de scoring prédisant la probabilité de défaut de paiement du client. De plus, la société souhaiterait mettre en place une transparence vis-à-vis des clients à propos de la décision d'accorder ou non un prêt. Dans cette optique, il nous a été demandé de mettre en place un Dashboard interactif afin que les décisions d'octroi de prêt soient justifiées.

Pour cela, nous allons d'abord parcourir les jeux de données fournis par l'équipe (présents ci-dessous) en réalisant un prétraitement & une préanalyse exploratoire. Puis nous allons proposer une modélisation optimale que nous implémenterons dans le Dashboard interactif accompagnée d'interprétations pertinentes.

Base de donnée	Shape	Détails
Application_train.csv	(307511, 122)	Base principale - Informations Clients
Bureau.csv	(1716428, 17)	Informations Crédits antérieurs via des organismes externes
Credit_balance.csv	(3840312 , 23)	Soldes mensuels des cartes de crédit antérieures
Home_credit_des.csv	(219, 5)	Descriptions des différentes colonnes
Install_payments.csv	(13605401 , 8)	Historique des crédits déboursés dans Home Credit
Pos_cash.csv	(10001358 , 8)	Soldes mensuels des prêts POS (points de vente) et des soldes antérieurs que le demandeur a eu avec Home Credit
Prev_app.csv	(1670214 , 37)	Toutes les demandes antérieures de prêts pour le crédit immobilier des clients présents dans notre échantillon

Méthodologie d'entraînement du modèle :

Réalisation d'un split Train / Test (80% Train & 20% Test)

- Shape Train = (246 008, 93)
- Shape Test = (61 503, 93)

Contrôle sur la variable Target

Target à 0 représente 92% du jeu de données alors que les Target à 1 seulement 8%.
On constate donc un déséquilibre au sein de nos classes (Imbalanced data).
Mise en place d'un rééquilibrage des classes via 2 méthodes : Undersampling & OverSampling

- X_train undersamp : (39 916, 93)
- y_train undersamp : (39 916,)

- X_train oversamp : (452 100, 93)
- Y_train oversamp : (452 100,)

Standardisation des valeurs

Standardisation des valeurs réalisée via la fonction StandarScaler() sur l'ensemble des données d'entrainements et de test.

Modèles avant optimisation

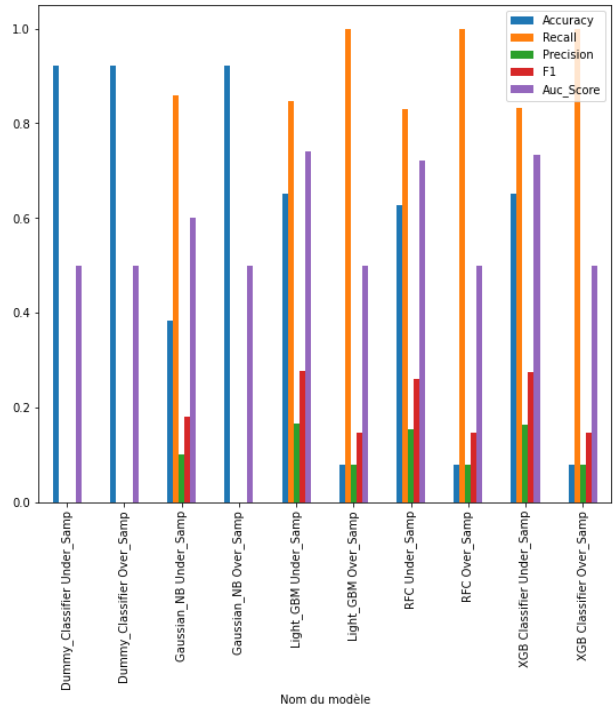
Utilisation des modèles suivants sur les Datasets UnderSampling & OverSampling:

- Dummy Classifier
- Gaussian Naive Bayes
- LGBM Classifier
- Random Forest Classifier
- XGB Classifier

	Nom du modèle	Accuracy	Recall	Precision	F1	Auc_Score	Scoring_weights	Temps de train	Temps de test
0	Dummy_Classifier Under_Samp	0.920866	0.000000	0.000000	0.000000	0.500000	7020.400000	0.004410	0.001414
0	Dummy_Classifier Over_Samp	0.920866	0.000000	0.000000	0.000000	0.500000	7020.400000	0.028374	0.000755
0	Gaussian_NB Under_Samp	0.382290	0.859051	0.100781	0.180398	0.600185	9485.800000	0.062966	0.080553
0	Gaussian_NB Over_Samp	0.920801	0.000000	0.000000	0.000000	0.499965	7020.933333	0.612391	0.078192
0	Light_GBM Under_Samp	0.650716	0.846517	0.165761	0.277236	0.740204	7313.066667	4.332100	0.395789
0	Light_GBM Over_Samp	0.079134	1.000000	0.079134	0.146663	0.500000	11651.666667	33.782687	0.298627
0	RFC Under_Samp	0.627839	0.830286	0.154804	0.260954	0.720364	7537.533333	26.802170	2.105816
0	RFC Over_Samp	0.079134	1.000000	0.079134	0.146663	0.500000	11651.666667	466.088266	0.778613
0	XGB Classifier Under_Samp	0.651464	0.831724	0.164119	0.274143	0.733849	7340.533333	15.368258	0.299305
0	XGB Classifier Over_Samp	0.079134	1.000000	0.079134	0.146663	0.500000	11651.666667	261.137866	0.304883

On constate que les modèles encadrés en rouge présentent un Scoring_weights (Score personnalisé détaillé par la suite) très élevé. Cela signifie qu'il y a une prédiction de faux négatifs (clients non conformes prédits à conforme) trop importante. Il ne faut donc pas les prendre en compte par la suite.

Pour le choix du modèle optimal nous pouvons nous baser en plus de ce score personnalisé, des autres scores :



Les modèles retenus sont donc :

- Light GBM Under_Samp
- RFC Under_Samp
- XGB Under_Samp

Modèles après optimisation

Nous allons optimiser les modèles retenus en appliquant un GridSearchCv en cherchant à minimiser la métrique personnalisée.

LGBM_Under_Samp Opti	XGB Under_Samp Opti	RFC Under_Samp Opti
Params_grid • 'num_leaves': np.arange(10, 100, 10), • 'max_depth' : np.arange(5,15,5), • 'learning_rate': np.arange(0.2, 1.0, 0.2) **Scoring** = custom_scorer	**Params_grid** • 'n_estimators' : np.arange(50,500,50), • 'learning_rate': [0.01, 0.1] **Scoring** = custom_scorer	**Params_grid** • 'n_estimators': np.arange(200, 1000, 200), • 'max_depth' : np.arange(5,15,5), **Scoring** = custom_scorer

	Nom du modèle	Accuracy	Recall	Precision	F1	Auc_Score	Scoring_weights	Temps de train	Temps de test	Best_params
0	Dummy_Classifier Under_Samp	0.920866	0.000000	0.000000	0.000000	0.500000	7020.400000	0.004410	0.001414	NaN
0	Dummy_Classifier Over_Samp	0.920866	0.000000	0.000000	0.000000	0.500000	7020.400000	0.028374	0.000755	NaN
0	Gaussian_NB Under_Samp	0.382290	0.859051	0.100781	0.180398	0.600185	9485.800000	0.062966	0.080553	NaN
0	Gaussian_NB Over_Samp	0.920801	0.000000	0.000000	0.000000	0.499965	7020.933333	0.612391	0.078192	NaN
0	Light_GBM Under_Samp	0.650716	0.846517	0.165761	0.277236	0.740204	7313.066667	4.332100	0.395789	NaN
0	Light_GBM Over_Samp	0.079134	1.000000	0.079134	0.146663	0.500000	11651.666667	33.782687	0.298627	NaN
0	RFC Under_Samp	0.627839	0.830286	0.154804	0.260954	0.720364	7537.533333	26.802170	2.105816	NaN
0	RFC Over_Samp	0.079134	1.000000	0.079134	0.146663	0.500000	11651.666667	466.088266	0.778613	NaN
0	XGB Classifier Under_Samp	0.651464	0.831724	0.164119	0.274143	0.733849	7340.533333	15.368258	0.299305	NaN
0	XGB Classifier Over_Samp	0.079134	1.000000	0.079134	0.146663	0.500000	11651.666667	261.137866	0.304883	NaN
0	Light_GBM Under_Samp Opti	0.659041	0.834806	0.167692	0.279282	0.739371	7271.400000	1516.198704	0.717676	{'learning_rate': 0.2, 'max_depth': 10, 'num_l...
0	XGB Under_Samp Opti	0.720453	0.802548	0.193962	0.312418	0.757973	6841.066667	2845.911411	1.100619	{'learning_rate': 0.1, 'n_estimators': 450}
0	RFC Under_Samp Opti	0.611206	0.833162	0.149328	0.253263	0.712647	7667.400000	2119.851735	4.996874	{'max_depth': 10, 'n_estimators': 400}

Le meilleur modèle est donc le XGB Under_Sampling

Métrique d'évaluation personnalisée :

L'objectif de cette métrique est de mettre un poids aux prédictions des différents modèles afin de mesurer leur véracité. En domaine bancaire (et dans d'autres) une mauvaise prediction peut entrainer des conséquences importantes (perte business, défaut médical…). C'est donc pour cela que nous allons mettre un poids fort aux prédictions positives alors qu'elles ont été vérifiées comme négatives. L'objectif étant d'avoir un modèle avec le score le plus bas possible.

Voici les étapes pour chaque modèle :

- Récupération des labels prédits (y_pred) et des labels test (y_test)
- Récupération de la matrice de confusion
- Scoring suivant le schéma suivant :

Label	Poids associé
Vrai négatif	**1**
Faux positif	**3**
Faux négatif	**10**
Vrai positif	**1**

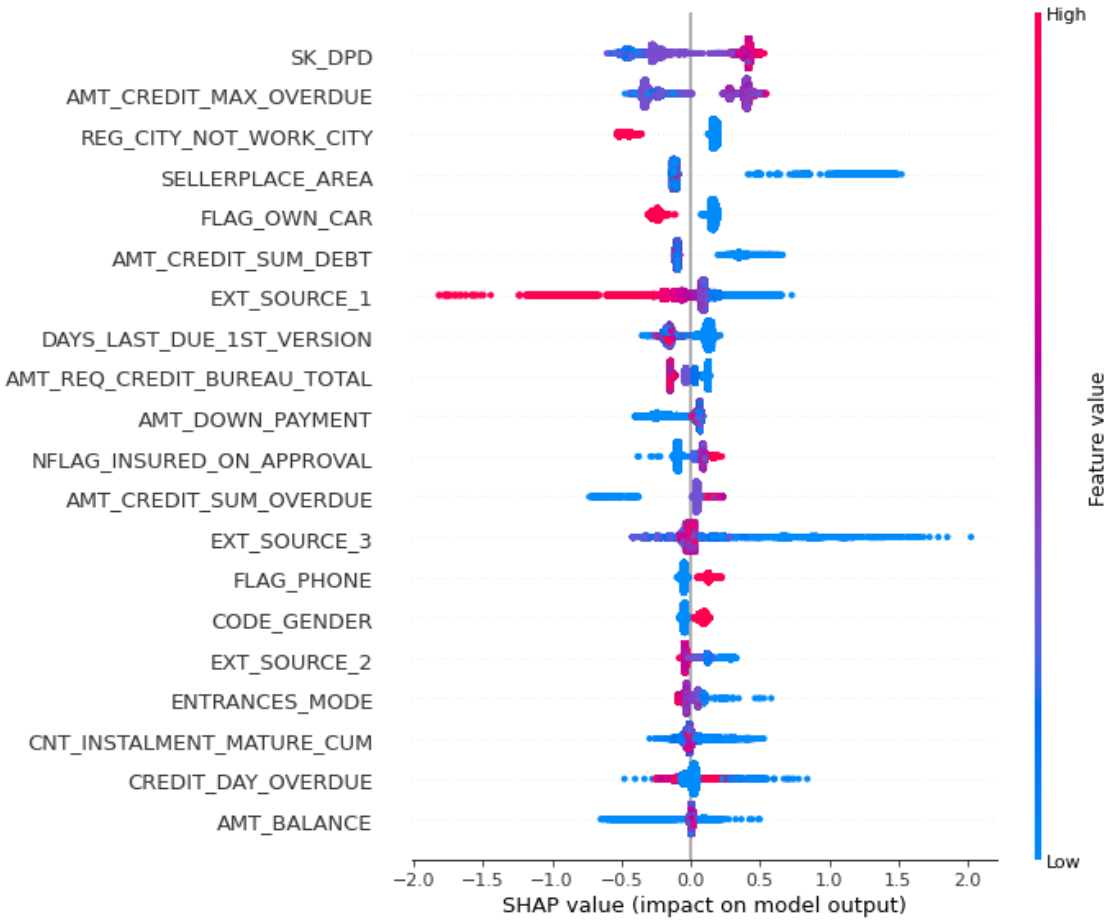
Faux positif : Perte de clients => Clients prédits négatifs alors qu'ils sont positifs
Faux négatif : Prêt accordé à des clients pas conformes
Scoring_weights = (VN * 1 + FP*3 + FN*10 + VP*1) / 15

Les valeurs poids ont été stockées dans des variables (a, b, c, d) de sorte à modifier les poids rapidement.

Interprétabilité du modèle :

Pour interpreter le modèle, nous allons utiliser la fonction Shap. Pour cela nous allons passer en entrée le modèle final ainsi qu'un jeu d'entrainement (ici le test) afin de récupérer les Shap values. A partir de cela nous allons pouvoir faire 2 interpretations importantes :

Shap Summary



Ce graphique nous indique les features qui impactent le plus le modèle en fonction de leur valeurs. En bleu nous avons les valeurs faibles et en rouge les valeurs fortes. Lorsque le modèle est impacté positivement par un feature les valeurs sont vers la droite et inversement lorsque le modèle est impacté négativement.

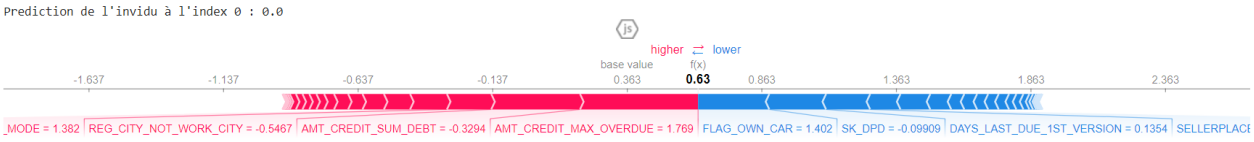
Ce graphique est très utile car il nous permet de faire un Feature selection supplémentaire. Nous allons réduire l'ensemble du jeu de données à ces colonnes et appliquer le modèle final.

Nous pouvons voir que le modèle s'améliore encore :

Nom du modèle	Accuracy	Precision	Recall	F1	Auc_Score	Scoring_weights
Modèle final	0.75	0.72	0.2	0.31	0.74	6786

C'est ce modèle que nous allons utiliser dans notre Dashboard car les valeurs d'entrées ont été réduites tout en améliorant le modèle.

Shap Individual



Ce graphique nous permet d'avoir les mêmes informations que le Shap Summary mais cette fois pour un individu en particulier.

Les caractéristiques rouges conduisent notre prédiction à être 0 : client pas en défaut
Les 3 caractéristiques qui contribuent le plus sont les suivantes : AMT_CREDIT_MAX_OVERDUE / AMT_CREDIT_SUM_DEBT / REG_CITY

Les caractéristiques bleues indiquent les caractéristiques réduisant la probabilité que le client soit à 0.

Determination du seuil à partir duquel un client est considéré comme défaut

Pour déterminer si un client est en défaut de paiement ou non, il est préférable d'utiliser la probabilité de prédiction plutôt que la prédiction. Cela nous permet de connaître le % de précision et ainsi permettre à la Banque de prendre une décision. Pour cela, il faut déterminer un seuil à partir du quel il est considéré comme défaut.

Les étapes :

- Récupération des listes de prédictions à 1 (défaut) pour un seuil allant de 0 à 1
- Custom_metric entre chacune des listes et le y_test initial pour récupérer le scoring_weights
- Scoring_weight le plus faible lorsque le seuil est à 0.7

	Thresholder	Custom_metrics
0	0.00	11651.666667
1	0.05	11466.933333
2	0.10	10840.666667
3	0.15	10052.200000
4	0.20	9325.466667
5	0.25	8704.066667
6	0.30	8160.266667
7	0.35	7706.200000
8	0.40	7329.200000
9	0.45	7039.133333
10	0.50	6786.666667
11	0.55	6589.800000
12	0.60	6430.666667
13	0.65	6324.400000
14	0.70	6272.800000
15	0.75	6275.933333
16	0.80	6326.866667
17	0.85	6420.000000
18	0.90	6626.133333
19	0.95	6814.733333

Si le client est prédit à 1 avec une probabilité supérieure ou égale à 70%, il sera considéré comme un client à défaut.

Limites et améliorations possibles :

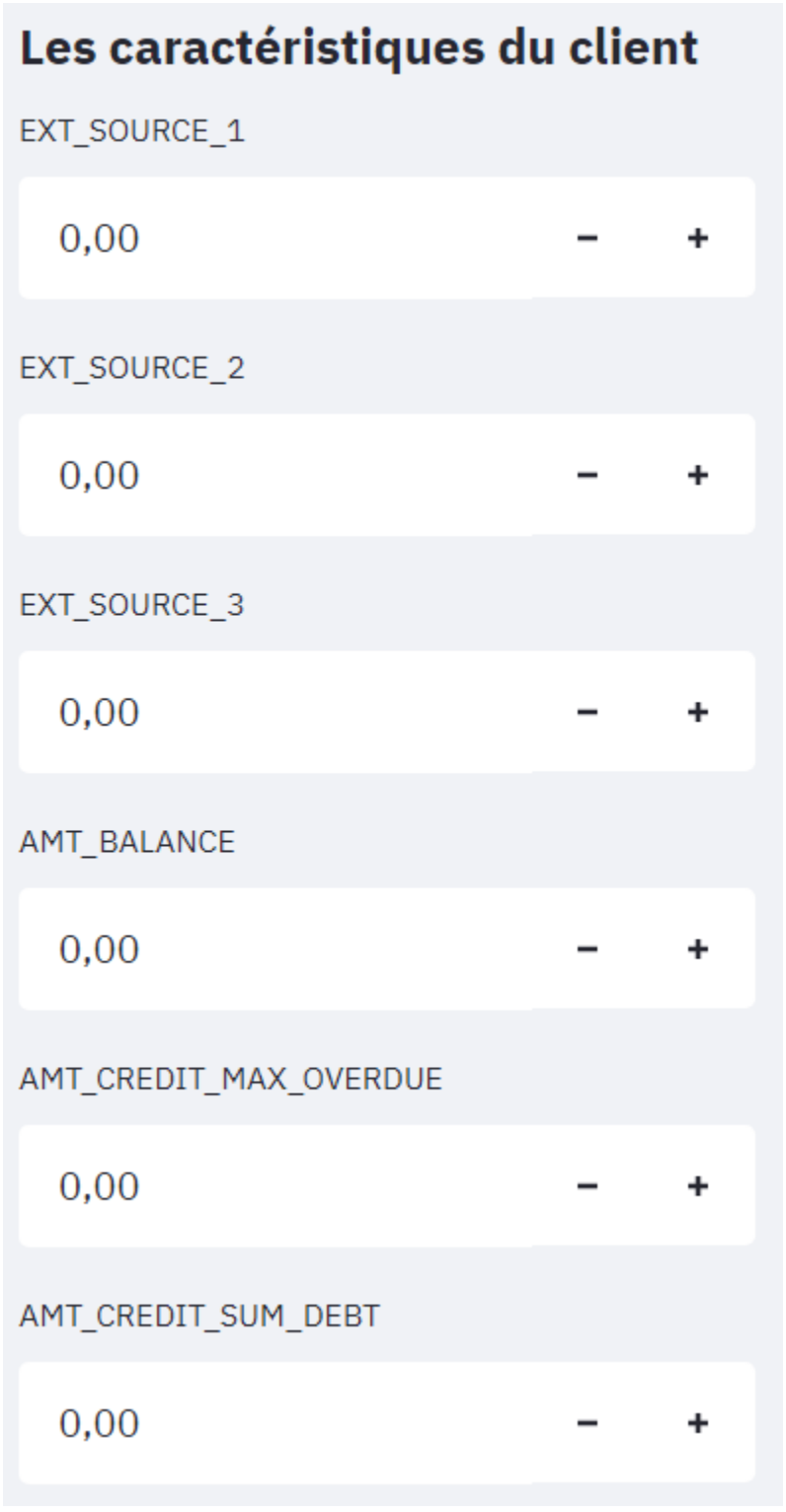
Comme le montre les interprétations Shap, de nombreux features impactent positivement ou négativement le modèle. La réduction du Dataset à ces features seulement a permis d'améliorer les résultats de sortie. Cela signifie que certains features perturbaient le modèle mais il faudrait comprendre pourquoi.

Malgré le scoring weights bas par rapport aux autres modèles, il reste tout de même présent. Cela veut dire que le taux de faux négatifs est toujours présent et peut impacter négativement la banque.

Les paramètres utilisés dans le GridSearchCv peuvent être grandement améliorés.
Les technologies utilisées ne permettent pas d'avoir le modèle le plus optimal possible

Pour améliorer le modèle nous pourrions donc utiliser des plages de valeurs plus grandes pour le paramétrage du GridSearchCv. Cela prendrait plus de temps et demanderait plus de ressources mais nous pourrions avoir un modèle beaucoup plus performant.

Présentation du Dashboard :



Application permettant de prédire l'accord à un prêt bancaire

Synthèse Client

	EXT_SOURCE_1	EXT_SOURCE_2	EXT_SOURCE_3	AMT_BALANCE	AMT_CREDIT_MAX_OV
39916	0.0000	0.0000	0.0000	0.0000	0

Resultat de la prévision

	Probabilité d'être en défaut (Défault si >= 70%)
0	97.32504272460938

Client en défault

Synthèse globale des clients

	EXT_SOURCE_1	EXT_SOURCE_2	EXT_SOURCE_3	AMT_BALANCE	AMT_CREDIT_MAX_OVER
0	0.0830	0.2629	0.1394	131,079.2497	1,681.
1	0.4099	0.6200	0.3606	118,763.6839	558.
2	0.4859	0.5758	0.3540	0.0000	0.
3	0.5494	0.3542	0.6212	44,387.4990	0.
4	0.5945	0.4466	0.3313	4,680.0331	-1,990.
5	0.2244	0.1852	0.3125	80,274.6660	7,128.
6	0.7245	0.7496	0.7993	67,817.9452	17,147.

Filtrage

Nombre de lignes à afficher

2 − +

Nombre de colonnes à afficher

2 − +

Colonne numéro 1

EXT_SOURCE_1

Colonne numéro 2

EXT_SOURCE_2

	EXT_SOURCE_1	EXT_SOURCE_2
0	0.0830	0.2629
1	0.4099	0.6200

Téléchargement du fichier filtré

Download CSV File

Analyse Shap

Nombre d'individus à comparer

2 − +

Index Individu numéro 1

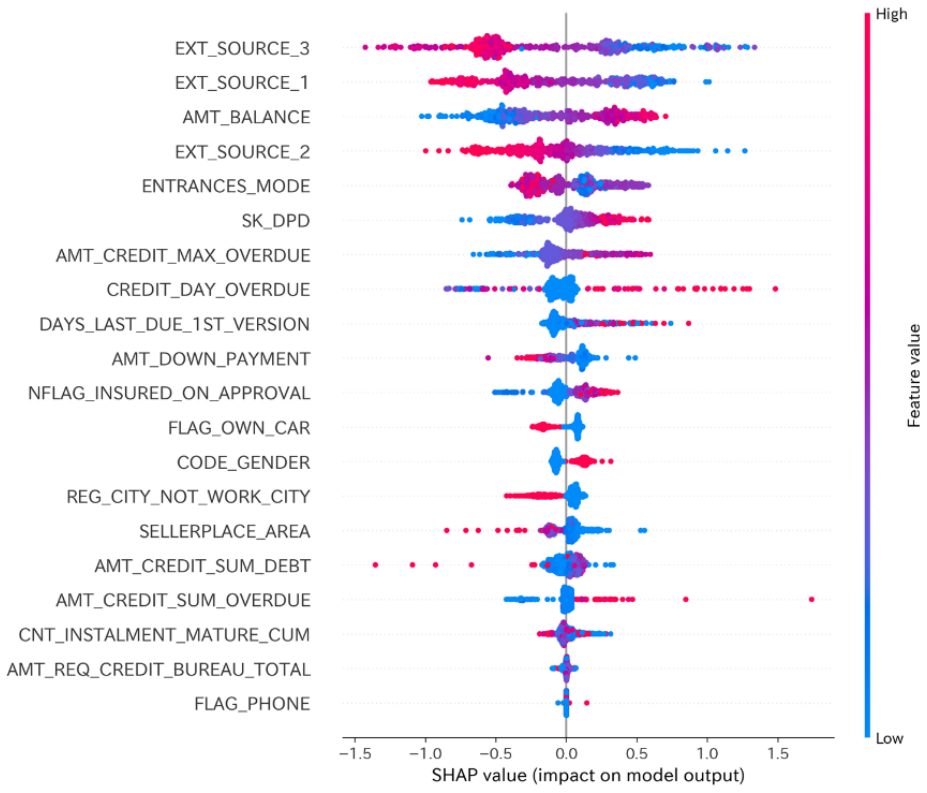
0

Index Individu numéro 2

1

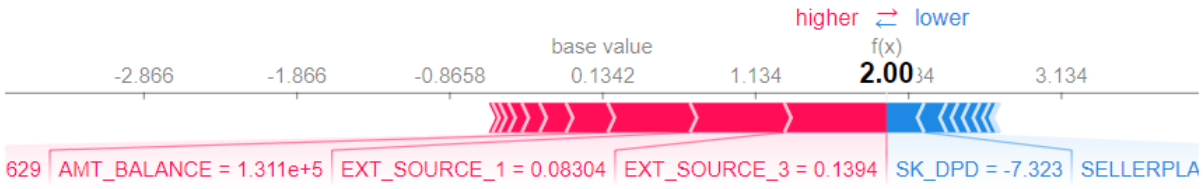
Lancer l'analyse Shap

Shap Summary



Shap Individual

Interpretation Shap à l'index: `0` - Prédiction : `1.0`



Interpretation Shap à l'index: `1` - Prédiction : `1.0`