NOTE MÉTHODOLOGIQUE

Implémentez un modèle de scoring

Table des matières

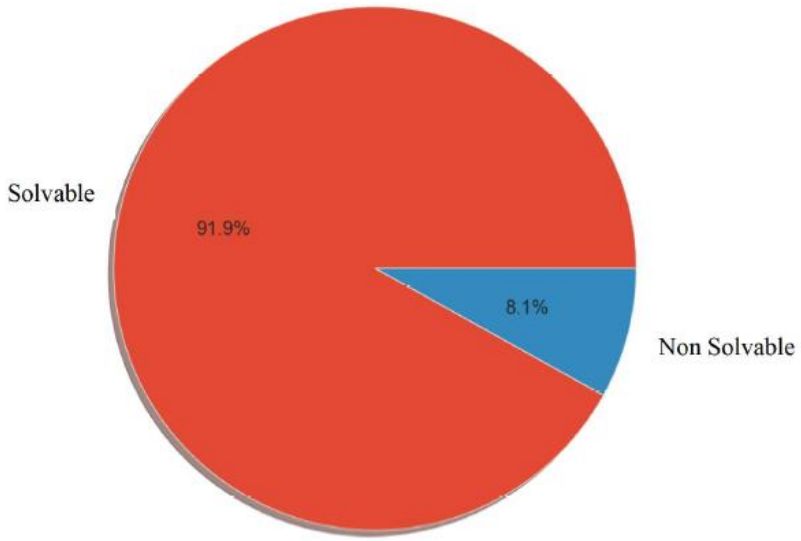
1) Le contexte

La société finance nommée “Prêt à dépenser “ propose des crédits à la consommation pour des personnes ayant peu ou pas d’historique de prêt. Cette entreprise souhaite développer un outil utilisant un modèle de scroring permettant d’obtenir la probabilité de défaut de paiement pour appuyer la décision d’accorder ou non un prêt à un client potentiel en s’appuyant sur des sources de données variés (données comportementales, données provenant d’autres institutions financières, etc.). Cet outil sera accompagné d’un Dashboard interactif permettant d’interpréter les prédictions faites par l’outil. Prédire si un client remboursera ou non un prêt ou s’il aura des difficultés à le faire est un besoin commercial essentiel.

2) Les données

Pour réaliser cet outil, c’est le service dédié à la fourniture de crédits à la population non bancarisée, Home Credit, qui fournit une base de données. Cette base de données regroupe des informations générales pour chaque client ainsi que des informations supplémentaires sur chaque prêt contracté par les clients. N’étant pas du métier de la banque, je me concentre essentiellement sur les données générales de chaque client comme l’âge, le sexe, les revenus, l’emploi, le logement, les informations de crédit en cours, des notations externes, etc.

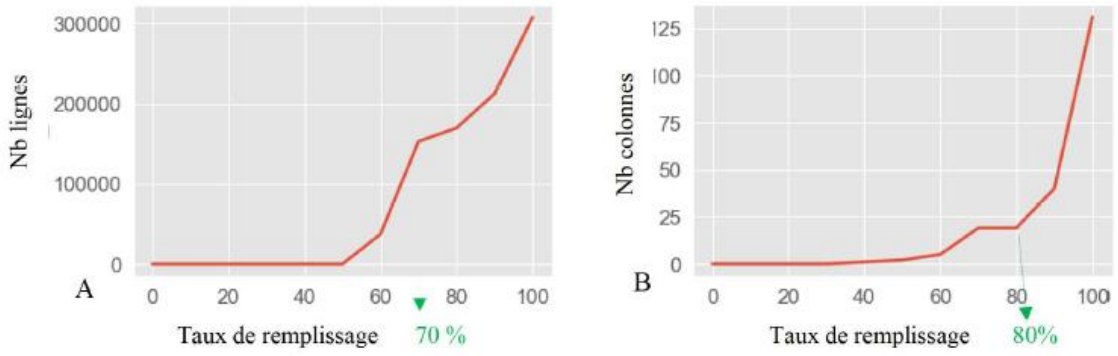
En observant de plus près la base de données, il apparait un fort déséquilibre entre les personnes solvables et les personnes non solvable. Cette observation est représentée sur la figure ci-dessous.



3) Traitement des données

Avant d’utiliser et d’exploiter au mieux les données à des fins de prédictions, le prétraitement de celles-ci est une tâche essentielle qui intervient après l’exploration des données. Il s’agit de plusieurs étapes de nettoyage de données brutes qui sont souvent peu fiables et incomplètes et leur utilisation peut générer des résultats faussés. Dans un premier temps, il est possible de les enrichir par de nouvelles variables (compréhensibles par des personnes qui ne sont pas du domaine bancaire) à partir des données déjà existantes comme le pourcentage du montant des crédits sur la totalité des revenus, la durée totale du paiement des crédit... De nombreux clients ont un important nombre de données peu fiables (moins de 70% de remplissage). Cette même démarche peut être faite pour les colonnes (moins de 80% de remplissage). Cependant il faut veiller à garder suffisamment d’informations.

Nombre de lignes (A) et de colonnes supprimées (B) en fonction d'un taux de remplissage



Après cette réduction de données, il est nécessaire d'imputer les données manquantes restantes, effectuer une standardisation des données pour finir par un encodage numérique des variables catégorielles.

4) Méthodologique : entrainement de modèles

A. Choix d'algorithmes

Un modèle de prédiction prend en entrée des données et donne en sortie une conclusion.
Pour déterminer l'algorithme optimal de classification adapté à la problématique, deux algorithmes ont été testés. Un algorithme de régression binomiale (régression logistique comme baseline) et un autre algorithme de gradient boosting Light-GBM.

B. Division des données

Dans le cas d'une prédiction supervisée, l'entrainement d'un modèle nécessite une étape de division du jeu de données entre le prétraitement des données et la modélisation. Cette division permet de créer un échantillon d'apprentissage (train) qui représente 80% des données et un échantillon test qui représente 20% des données.

C. Choix des hyperparamètres de chaque algorithme

Pour chaque algorithme à tester, une recherche d'hyper paramètres doit être effectué pour que le modèle soit le plus adapté aux données. L'échantillon d'apprentissage est l'échantillon principal qui permet à un modèle d'ajuster sa prédiction par le biais d'une validation croisée. Durant cette étape, cet échantillon est séparé en plusieurs sous-ensemble sur lesquels les algorithmes apprennent pour ajuster le modèle. L'échantillon test, quant à lui, joue un rôle important car il permet d'évaluer les performances d'un modèle en comparant les prédictions du modèle aux vraies valeurs de la variable à expliquer. La version du modèle avec le score d'erreur minimisé est celui à garder pour la suite avec les paramètres optimums.

D. Contrecarrer le déséquilibre

Comme mentionné précédemment, dans les données il y a un fort déséquilibre entre les clients solvable et non solvable. Ce déséquilibre va avoir un impact sur la performance de l'algorithme et les prédictions seront faussées car le modèle attribuera beaucoup plus fréquemment la classe la plus

représentée aux clients. Pour pallier ce problème déséquilibrage, quatre approches peuvent être appliquées sur les données d'entrainement :

- Classe weights est une méthode directement gérée par les modèles et qui permet de pénaliser les poids associés aux observations de la classe sur-représentés (ici solvable).
- Over-sampling est une méthode qui va dupliquer aléatoirement des données existantes de la classe sous représentée pour que chaque classe ait le même nombre de données que la classe sur-représentée à l'origine.
- SMOTE est une méthode qui va créer de nouvelles données pour la classe sous-représentée à partir des données existantes (et donc de la variété) pour que chaque classe ait le même nombre de données que la classe sur-représentée à l'origine.

Under-sampling est une méthode qui va sélectionner une partie des observations sur-représentées pour que chaque classe ait le même nombre de données que la classe sous-représentée à l'origine.

5) Choix d'un algorithme adapté

A. Fonction coût : métrique métier

Pour déterminer m'algorithme optimal, chacun des modèles a été entrainé selon les différentes méthodes pours pallier le déséquilibre. Les classes prédites sont comparées aux classes réelles et il est possible de créer une matrice de confusion pour représenter les erreurs commises par le modèle :

		Classe réelle	
		Negatif (0)	Positif (1)
Classe prédite	Negatif (0)	TN	FP
	Positif (1)	FN	TP

Dans cette représentation, la classe 0 correspond aux clients solvables qui sont négatifs ai refus de l'accord de prêt contrairement aux clients non solvables de la classe 1 qui sont positifs au refus de l'accord de prêt.

Ainsi :
-Accorder un crédit à un client ne pouvant pas le rembourser par la suite (**FN**) est synonyme de perte.
-Accorder un crédit à un client qui le remboursera par la suite (**TN**) est un gain.
-Ne pas accorder le prêt et que le client ne peut pas rembourser (**TP**) n'est ni une perte, ni un gain.
-Ne pas accorder le prêt alors que le client pouvait rembourser (**FP**) est une perte de client donc d'argent.

Une société de crédit cherche à gagner de l'argent et à "maximiser" ses gains. Il est donc possible de créer pour chaque modèle un score gain total en pondérant chaque cas par un coefficient qui pénalisera négativement les prédictions néfastes à la banque (**FN** et **FP**) et un qui valorisera les prédictions bénéfiques pour la banque (**TN**). Ce score sera alors normalisé par normalisation min-max feature scalling.

$$gain\_total = (TN * coeff\_tn + FP * coeff\_fp + FN * coeff\_fn + TP * coeff\_tp)$$

$$gain = \frac{(gain\_total - gain\_min)}{(gain\_max - gain\_min)}$$

Avec :

Coefficient arbitraire possible en respectant la métrique métier :
- FN ➔ perte d'argent pour la banque ➔ -100
- TP ➔ refus de prêt ➔ 0
- TN ➔ prêt accordé, gain d'argent pour la banque ➔ +10

$$gain\_\min = (TN + FP) * coff_{fp} + (TP + FN) * coeff\_fn$$
$$gain\_\max = (TN + FP) * coeff\_tn + (TP + FN) * coeff\_tp$$

- FP ➔ client perdu, perte d'argent pour la banque ➔ -1

Cette fonction coût à été implémentée afin de pénaliser l'impact des erreurs sur la décision d'octroi de crédit.

B. Métrique d'évaluation

Pour analyser les résultats produits et ainsi démontrer l'efficacité de chaque modèle, il est nécessaire de se baser sur différentes métriques :
-L'accuracy est le ratio entre le nombre total de prédiction correctes et le nombre total de prédictions.
- La Precision est le ratio entre les vrais positifs et tous les positifs, il donne des indications sur les faux positifs.
- Le Recall est la proportion dans laquelle notre modèle identifie correctement les vrais positifs.
- Le Score F1 est la moyenne harmonique de la précision et du rappel. Un bon score F1 indique une bonne précision et une bonne valeur de rappel.
- L'AUC (l'aire sous la courbe ROC). Plus l'AUC est élevée, plus la performance du modèle à distinguer les classes positives et négatives est bonne.

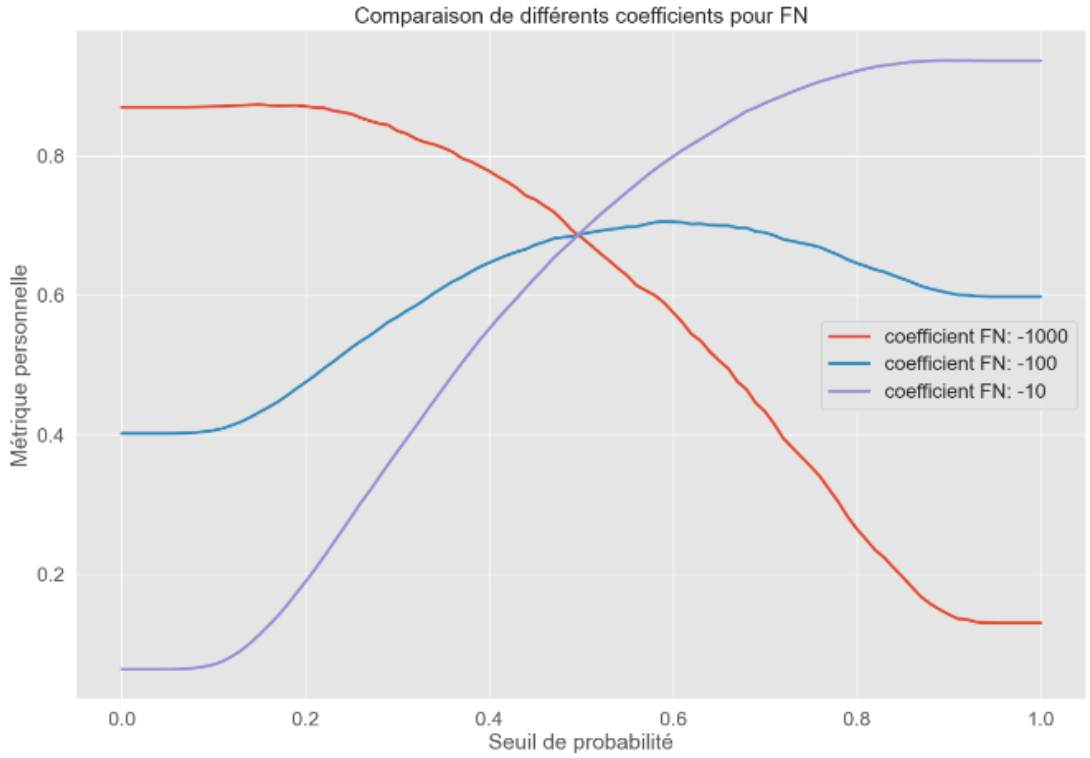
Le modèle utilisé pour l'outil doit maximiser l'AUC, le score F1, le Recall. Une banque cherche à ne pas accorder de prêt à des client qui ne peuvent pas le rembourser (TP à maximiser). La banque ne veut pas accorder un prêt à un client qui ne peut pas le rembourser (FN à minimiser).

Chaque modèle doit donc maximiser ce gain et remplir les contraintes des métriques d'évaluation. Toutes ces conditions sont validées par le modèle **Light Gradient Boosting Machine** (modèle robuste qui repose sur le principe d'arbre de décisions) avec la méthode de l'under-sampling pour pallier le déséquilibre qui donc ici le modèle optimal.

C. Coefficient optimal

Le seuil de probabilité est la valeur de prédiction à partir de laquelle le modèle va considérer qu'un client est solvable ou non compte tenu de la fonction gain employée. Des coefficients arbitraires ont précédemment été définis mais ceux-ci ne sont peut-être pas optimums. Le coefficient de FN est celui qui a la plus grande importance dans notre métrique mais le coefficient a-t-il le bon ordre de

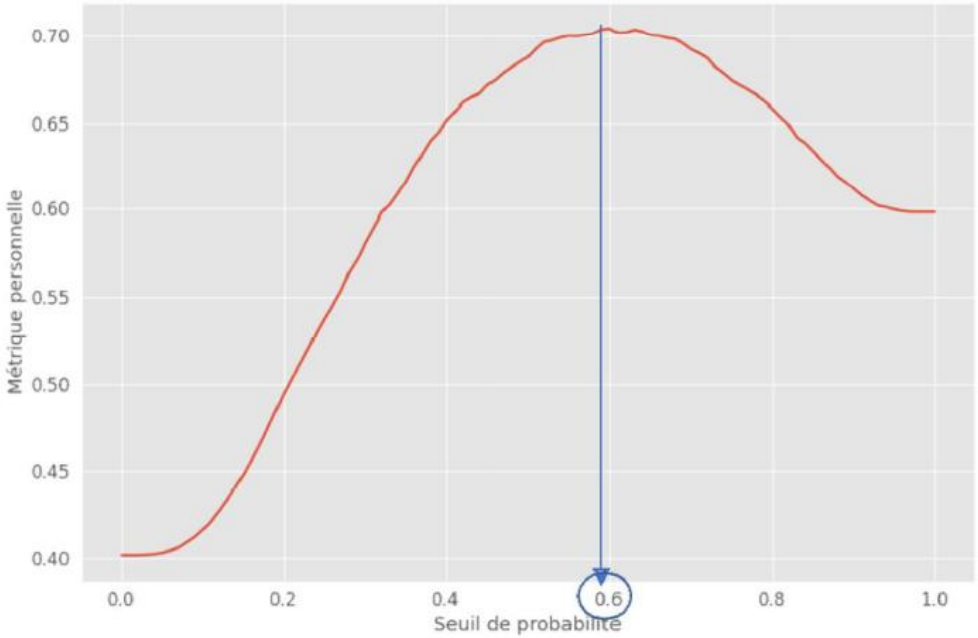
grandeur. Pour vérifier cela, le score gain a été recalculé avec le modèle optimal pour différent seuil de probabilité en variant l'ordre de grandeur du coefficient FN :



Avec un coefficient de -100, le seuil optimal pour que la banque ait le maximum de gain d'argent se situe aux alentours d'une probabilité de 0.2 à partir de laquelle on peut voir que la courbe des gains décroit rapidement. Ce coefficient est trop strict car seules les personnes avec une probabilité de défaut de paiement <0.2% se voit accepter leurs prêts ce qui correspond à peu de personnes.
Au contraire, avec un coefficient de -10, la banque est laxiste dans le choix des clients pour l'accord d'un prêt car le seuil optimal pour que la banque ait le maximum de gain d'argent se situe aux alentours d'une probabilité de 0.9 et toutes les personnes avec une probabilité de défaut de paiement <0.9% se verrait accepter leur prêt ce qui correspond à beaucoup de personnes.
Un coefficient de -100 est mieux adapté car le seuil optimal se situerait aux alentours de 0.55 ce qui est correct pour déterminer si oui ou non le prêt est accepté.

D. Seuil de probabilité

La métrique personnelle gain doit être maximiser pour que la banque gagne le plus d'argent et ainsi permettre de déterminer à partir de quel seuil de probabilité un client est considéré comme solvable ou non. Généralement le seuil de probabilité choisi par les modèles pour classer les individus est de 0.5 mais au vu de la courbe du score gain (avec un coefficient de -10), il ne s'agit pas du seuil optimal pour que la banque maximise ses gains.



Les résultats montrent un seuil de 0.6 ce qui signifie qu'un client avec une prédiction au-dessus de cette valeur sera catégorisé comme non solvable et un client avec une prédiction en dessous de cette valeur se considéré comme solvable.

6) Tableau de synthèse des résultats

Classification Models Provide Feedback — Share

Experiment ID: 268604497707487495 Artifact Location: mlflow-artifacts:/268604497707487495

Description Edit

Table view | Chart view | Artifact view | metrics.rmse < 1 and params.model = "tree" | Time created | State Active | Refresh

Sort: Created | Columns

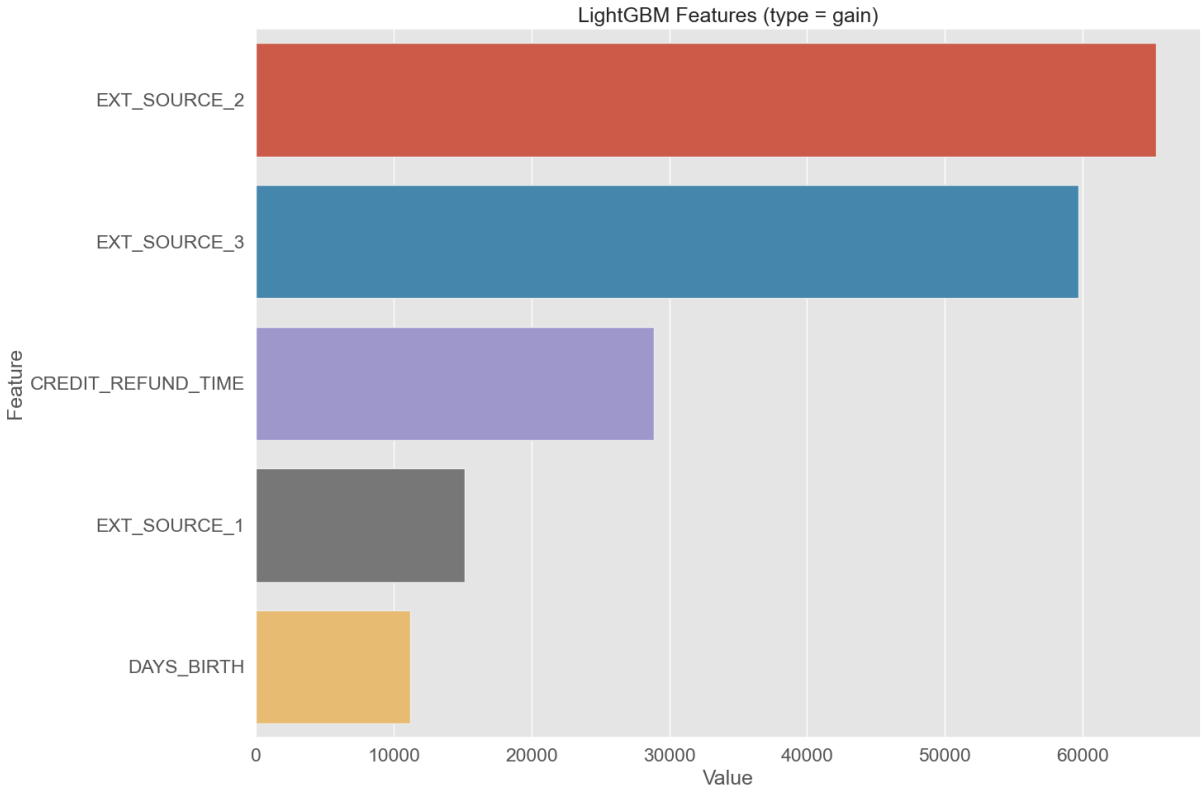
				Metrics					
Run Name	Created	Duration	AUC	accuracy	execution_time	f1	precision	prediction_time
classification/SMOTE - LGBMClassifier	22 hours ago	7.3s	0.739	0.93	49.5	0.059	0.422	1.251
classification/SMOTE - Regression logistique	22 hours ago	7.0s	0.728	0.931	9.66	0.026	0.519	0.056
classification/RandomOverSampler - LGBMClassi...	22 hours ago	7.6s	0.759	0.744	74.83	0.253	0.158	2.26
classification/RandomOverSampler Regression I...	22 hours ago	7.1s	0.743	0.689	10.55	0.228	0.137	0.056
classification/RandomUnderSampler - LGBMClas...	22 hours ago	10.7s	0.755	0.691	13.55	0.233	0.14	2.058
classification/RandomUnderSampler Regression ...	22 hours ago	9.5s	0.74	0.678	0.907	0.222	0.133	0.064
classification/Class Weight - LGBMClassifier	22 hours ago	9.0s	0.761	0.742	44.75	0.254	0.159	2.342

10 matching runs

7) Interprétabilité

A. Importance des features

Maintenant que le modèle, les coefficients et le seuil de solvabilité sont été définie, il est désormais intéressant de savoir quelles sont les informations qui ont un poids important dans le calcul de la probabilité de solvabilité d'un client. Le modèle LGBM est un modèle basé sur des arbre et l'importance des features est données par la fonction feature_importance avec pour option importance_type='gain'. L'importance des feature se base sur la réduction moyenne de la perte obtenue lors de l'entrainement du modèle.



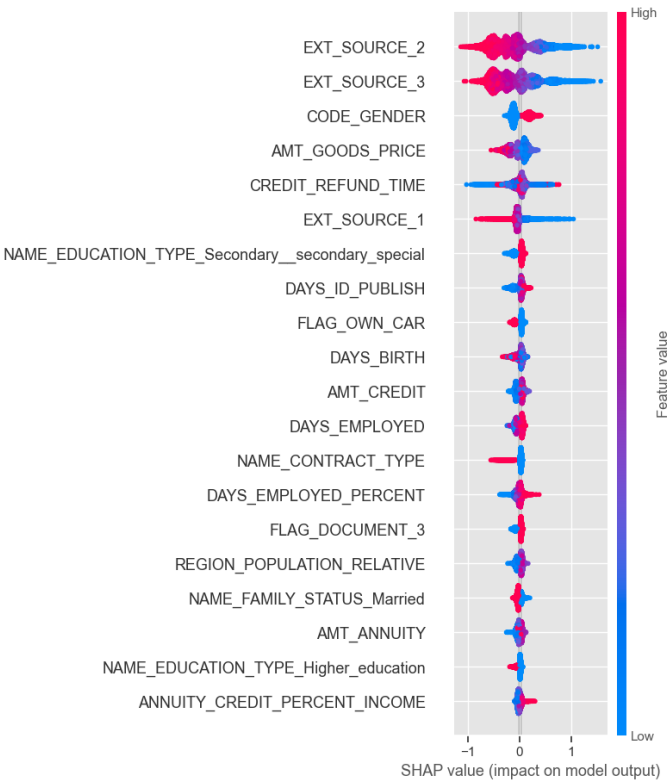
Avec cette représentation, on peut dire que les features les plus importantes pour la prédiction d'accord d'un prêt sont les source extérieurs 2 et 3 qui sont des scores normalisés crées à partir de sources des données externes. Puis on trouve la durée que va mettre un client à rembourser un prêt en année ainsi que la source externe 1. Enfin le nombre de jours depuis la naissance des clients donc leur âge joue un rôle important dans l'acceptation d'un crédit.

Pour plus de précision et pour connaitre le sens d'influence de chacune des variables, il est possible d'utiliser la librairie SHAP (Shapley Additive exPlanation) qui explique la sortie du tout modèle d'apprentissage automatique en utilisant la théorie des jeux. Lorsque l'in regarde globalement l'importance des features avec l'outil SHAP, nous retrouvons les cinq features déterminées précédemment par la fonction intégrée dans notre modèle ainsi que les features représentant le sexe du client et le prix du bien acheté par des prêts de consommation qui sont, selon moi des variables discriminantes et que j'ai décidé de ne pas prendre en compte. SHAP nous apporte des informations sur le sens de l'importance de chaque feature.

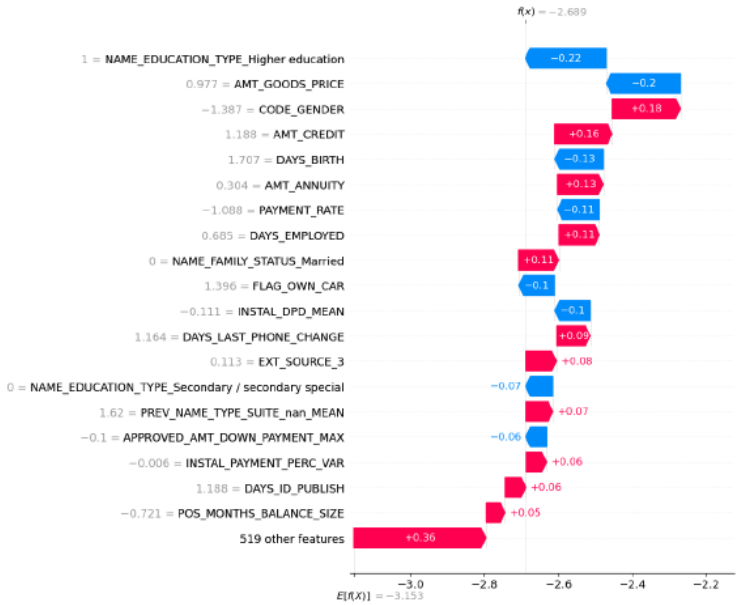
Nous traçons les valeurs SHAP de chaque caractéristique pour chaque échantillon sur l'axe x, puis nous laissons s'accumuler. Si nous colorons ensuite chaque point selon sa valeur, nous pouvons vois comment une valeur faible ou élevée affecte la sortie du modèle.

Pour EXT_SOURCE_2, EXT_SOURCE_3 et EXT_SOURCE_1 on peut voir que de faibles valeurs augmentent de manière significative la sortie de probabilités du modèle et donc le fait d'être non solvable et le non accord du prêt.

Au contraire, pour CREDIT_REFUND_TIME, plus le temps de remboursement des crédits est grand plus il y a de chance que le prêt ne soit pas accordé.



Interprétabilité locale (pour un seul client)



8) Limites et amélioration possibles

La modélisation à été effectué sur la base d'une métrique personnelle crées pour répondre au mieux au besoin de gain d'argent d'une banque. Les coefficients de cette métrique ont été choisis arbitrairement selon le bon sens. L'axe principal d'amélioration serait donc de définir plus précisément ces coefficients associés à chaque combinaison classe prédite/classe réelle car le modèle déterminé ici ne sera pas obligatoirement le meilleur.

Le modèle créé ici tend à être un modèle "éthique" car il ne tient pas compte de variables telles que le genre du client qui peut être une information discriminante dans l'attribution d'un prêt. Cependant toutes les variables discriminantes n'ont peut-être pas été enlevées. Il serait intéressant de comparer un modèle "éthique" à un modèle prenant en compte toutes les variables pour voir si le fait de supprimer ces variables discriminantes amène une perte d'information et de précision pour les prédictions ainsi qu'une perte de rentabilité pour la banque.

Le traitement des données a été réalisé de façon superficielle car une seule table de données a été prise en compte alors que les données sur les précédents prêts étaient disponibles. Il y a donc très probablement la possibilité d'améliorer la modélisation en utilisant d'autres features provenant des autres bases de données ainsi qu'en créant de nouvelles features grâce à l'équipe métier.

Enfin le Dashboard interactif pourra être amélioré pour répondre au mieux aux attentes et besoins des conseillers clients.

9) Analyse du Data Drift

347	326	0	21	0
Tests	Success	Warning	Fail	Error

120	9	0.075
Columns	Drifted Columns	Share of Drifted Columns

Drift is detected for 7.5% of columns (9 out of 120).

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
AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.161102
AMT_REQ_CREDIT_BUREAU_WEEK	num			Detected	Wasserstein distance (normed)	0.15426
NAME_CONTRACT_TYPE	cat			Detected	Jensen-Shannon distance	0.14755
DAYS_LAST_PHONE_CHANGE	num			Detected	Wasserstein distance (normed)	0.138977
FLAG_EMAIL	num			Detected	Jensen-Shannon distance	0.122121

Le seuil de détection de la dérive de l'ensemble de données est de 0,5

La dérive est détectée pour 7,5 % des colonnes (9 sur 120).