P7 Data Scientist

Note Méthodologique

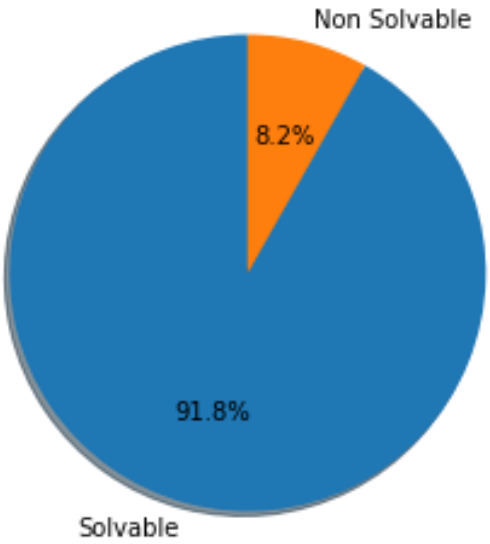
Aginth Muthulingam

I) Introduction

Nous sommes data Scientist pour la société financière "Prêt à dépenser" proposant des crédits à la consommation pour des personnes ayant peu ou pas d'historique de prêt. Notre entreprise souhaite développer un outil utilisant un modèle de scoring afin d'obtenir la probabilité de défaut de paiement d'un client potentiel en s'appuyant sur des sources de données variées (données personnelles, données provenant d'autres institutions financières, etc.). On nous demande donc de 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, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

II) Données

Nous avons à notre disposition une base de données fournit par Home Crédit, qui est un service dédié à la fourniture de crédits à des personnes ayant peu ou pas d'historique de prêt. Dans un premier temps je m'intéresse plus particulièrement à des variables essentielles tels que l'âge, le genre, le logement, le type d'emploi etc…
Je constate par ailleurs que dans ce jeu de données, il y a un fort déséquilibre entre les personnes solvable (capable de rembourser) et non solvable (incapable de rembourser). On peut l'observer sur la figure ci-dessus.



III) Traitement des données

Avant d'analyser nos données. Il faut faire un 1er pré-traitement. Pour celui-ci on va utiliser un kernel qu'on peut trouver sur le site *Kaggle* (https://www.kaggle.com/jsaguiar/lightgbm-with-simple-features). Ce kernel va assembler nos 9 fichiers, faire un encodage numérique des variables catégorielles et créer de nouvelles variables compréhensibles par les personnes qui ne sont pas du domaine bancaire comme le taux de paiement, le rapport salaire/crédit, etc… Après ce pré-traitement, on vérifie le pourcentage de valeurs manquantes selon les clients et variables. Je décide donc de

supprimer les clients présentant plus de 30% de valeurs manquantes et les colonnes présentant plus de 56% de valeurs manquantes. Après cette réduction de données, il est nécessaire d'imputer les données manquantes restantes, effectuer une standardisation des données.

IV) Méthodologie : entrainement de modèles

a) Choix d'algorithme

Un modèle de prédiction prend en entrée des données et donne en sortie une conclusion, qui dans notre cas sont « solvable » et « non solvable ». Pour déterminer l'algorithme optimal de classification, j'ai testé 3 algorithmes. Un algorithme KNN Clasifier, un algorithme de forêt d'arbre aléatoire (random forest) et un algorithme de gradient boosting (Light-GBM). L'entrainement de ces modèles nécessite auparavant une division de notre jeu de données en un jeu d'entrainement et un jeu de test. Dans mon le jeu d'entrainement représente 67% du jeu initiale et le jeu de test représente 33%.

Pour chacun de ces algorithmes, on va réaliser une recherche d'hyper paramètres afin que les modèles soit le plus adapté aux données. La version du modèle avec le score d'erreur minimisé est celui à garder pour la suite avec les paramètres optimums.

b) Gérer le déséquilibre de classe

Comme dis précédemment, nos données présentent un fort déséquilibre entre les clients solvables et non solvables. Ce déséquilibre peut avoir un impact sur la performance de l'algorithme et les prédictions peuvent être erronés. Pour contrecarrer ce problème de déséquilibrage, 3 approches sont envisagées :

- 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.

- 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.

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

V) Choix d'un algorithme adapté

a) Métrique Bancaire

Les classes prédites par nos différents modèles sont comparées aux classes réelles. On peut alors créer une matrice de confusion pour se représenter les erreurs commises par le modèle :

		Reality	
	Confusion matrix	**Negative : 0**	**Positive : 1**
Prediction	**Negative : 0**	**True Negative : TN**	**False Negative : FN**
	Positive : 1	**False Positive : FP**	**True Positive : TP**

La classe 0 correspond aux clients solvables qui sont négatifs au 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 :

- FN : Accorder un crédit à un client incapable de rembourser
- TN : Accorder un crédit à un client capable de payer
- TP : Pas accorder de prêt à un client incapable de rembourser
- FP : Pas accorder de prêt à un client capable de rembourser

Pour notre entreprise FN et FP représente une perte d'argent, TP un gain d'argent et enfin FP ne représente ni un gain ni une perte.

On cherche à maximiser nos gains du coût il est possible de créer une métrique, pour chacun de nos modèles, en pondérant chaque cas par un coefficient. J'ai considéré qu'accorder un crédit à une personne incapable de rembourser aura un impact bien plus fort sur nos gains. Du coup j'ai fixé les coefficients suivants :

- Coeff_FN = -100
- Coeff_FP = -1
- Coeff_TN = 10
- Coeff_TP = 0

On peut donc définir le score avec la formule suivante :

$$\text{Scoring} = 10 \times TN + 0 \times TP + (-100) \times FN + (-1) \times FP$$

On normalise ensuite ce score :

$$\text{Scoring normalisé} = \frac{Scoring - \min_{gain}}{\max_{gain} - \min_{gain}}$$

Avec : $\min_{gain} = (\text{TN} + \text{FP}) * (-1) + (\text{TP} + \text{FN}) * (-100)$

Et : $\max_{\text{gain}} = (TN + FP) * 10 + (TP + FN) * 0$

b) Métriques d'évaluation

Pour analyser les résultats produits et ainsi démontrer l'efficacité de chaque modèle, il est nécessaire de se baser sur différentes métriques :

Recall_1 = $\frac{TP}{TP+FN}$; Precision_1 = $\frac{TP}{TP+FP}$; F1score _1 = $\frac{2 \times precision\_1 \times recall\_1}{precision\_1 + recall\_1}$

Recall_0 = $\frac{TN}{TN+FP}$; Precision_0 = $\frac{TN}{TN+FN}$; F1score_0 = $\frac{2\times precision\_0\times recall\_0}{precision\_0+recall\_0}$

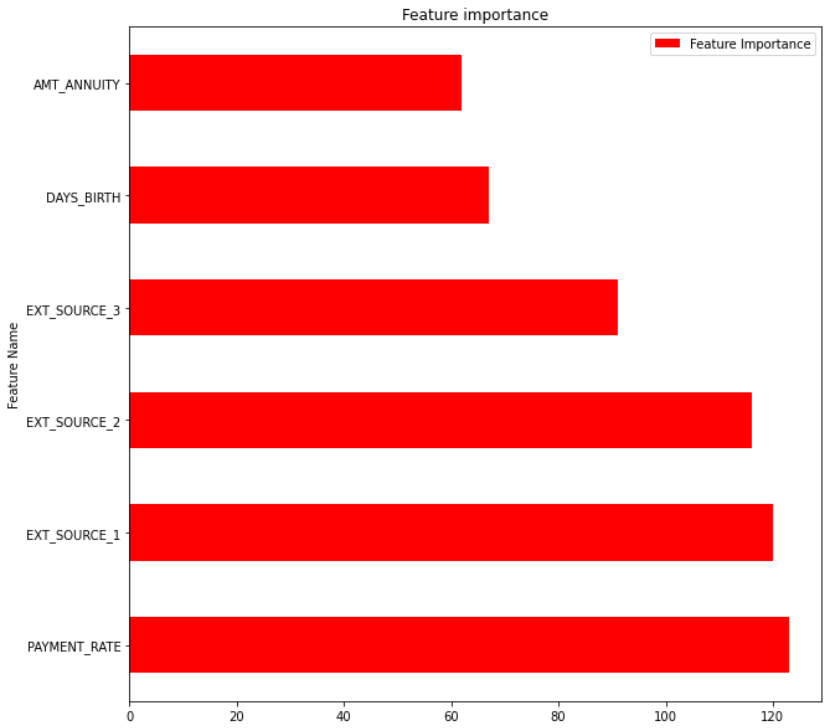
Le modèle utilisé pour l'outil doit maximiser chacune de ces métriques ainsi que la métrique bancaire. Chaque modèle doit donc maximiser ce gain et remplir les contraintes des métriques d'évaluation.

	TN	FP	FN	TP	precision_1	recall_1	f1-score_1	precision_0	recall_0	f1-score_0	scoring
LGBM Classifier & Over Sampling	65186	22831	2610	5394	0.191107	0.673913	0.297773	0.961502	0.740607	0.836721	0.710424
LGBM Classifier & Under Sampling	62085	25932	2336	5668	0.179367	0.708146	0.286234	0.963739	0.705375	0.814561	0.706629
Random forest Classifier & Over Sampling	52106	35911	2810	5194	0.126359	0.648926	0.211529	0.948831	0.591999	0.729097	0.617762
Random forest Classifier & Under Sampling	51349	36668	2775	5229	0.124806	0.653298	0.209575	0.948729	0.583399	0.722508	0.615033
KNN Classifier & Under Sampling	58168	29849	3595	4409	0.128700	0.550850	0.208651	0.941794	0.660872	0.776713	0.611080
Random forest Classifier & SMOTE	64908	23109	4791	3213	0.122065	0.401424	0.187205	0.931262	0.737448	0.823100	0.585376
KNN Classifier & Over Sampling	55211	32806	4000	4004	0.108775	0.500250	0.178694	0.932445	0.627277	0.750007	0.569789
LGBM Classifier & SMOTE	87801	216	7793	211	0.494145	0.026362	0.050053	0.918478	0.997546	0.956381	0.558022
KNN Classifier & SMOTE	21153	66864	1017	6987	0.094609	0.872939	0.170717	0.954127	0.240329	0.383947	0.526625

Toutes les conditions sont validées par le modèle Light Gradient Boosting Machine avec la méthode de l'over-sampling.

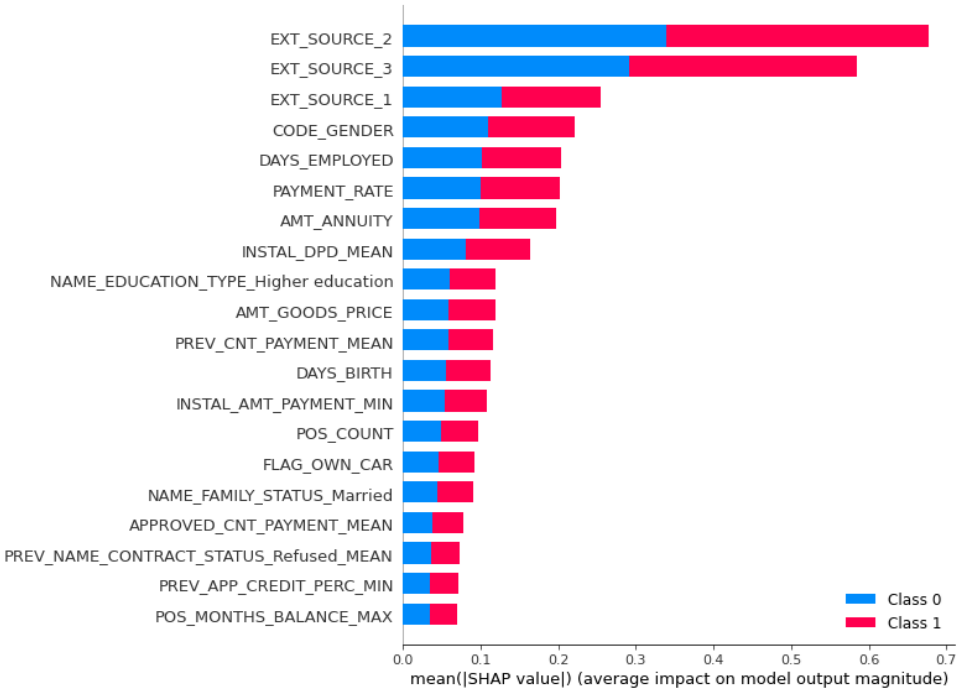
VI) Importance des features

Maintenant que le modèle est choisi, 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.



Avec cette représentation, on peut dire que les features les plus importantes pour la prédiction d'accord d'un prêt sont le taux de paiement et les sources extérieures 1, 2 et 3 qui sont des scores normalisés créés à partir de sources de données externes. Puis on trouve l'âge du client et l'annuité du crédit.

Pour plus de précision et pour connaître l'influence des variables sur chaque classe, il est possible d'utiliser la librairie SHAP.



Lorsque l'on regarde globalement l'importance des features avec l'outil SHAP, nous retrouvons les cinq features déterminées précédemment ainsi que les features représentant le sexe du client et le nombres de jours ou le client travaille. Ces 2 derniers semblent cependant être discriminantes et je ne décide donc de ne pas les prendre en compte.

VII) Limites et améliorations

Un des premiers axes qu'on pourrait dans notre modèle, serait de définir plus précisément les coefficients utilisés dans notre métrique. En effets celles-ci ont été définis arbitrairement et une expertise d'une personne spécialisée dans le bancaire serait judicieuse.

L'autres axe à améliorer serait de rendre notre modèle plus éthique. En effet notre jeu de données possède plusieurs variables pouvant être discriminante comme le genre ou encore le lieu d'habitat du client. Plusieurs de ces variables n'ont pas pu être enlevés du jeu de données. Cependant en enlevant ces données, le modèle peut être mois précis. Il faudrait donc vérifier si en les enlevant, il n'y a pas de perte de rentabilité pour la banque.

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