NOTE METHODOLOGIQUE

Cette note méthodologique est un des livrables attendus pour le projet « implémentez un modèle de scoring », du parcours Openclassroom data scientist.
Elle décrit la méthodologie d'entrainement, d'optimisation, d'évaluation et d'interprétabilité du modèle.

L'objectif du projet est de construire pour l'entreprise « Prêt à dépenser » un dashboard interactif qui fournit une prédiction sur le défaut de paiement de remboursement de prêt pour un client. La donnée provient de Home Crédit et est disponible sur Kaggle.

Maxime Dupouy
16/12/2022

1. METHODOLOGIE D'ENTRAINEMENT DU MODELE

1.1 La donnée :

La donnée utilisée provient d'un notebook de Kaggle, lui même inspiré de différents notebooks. J'ai opté pour ce traitement de la donnée au vu des résultats obtenus avec une ingénierie des données et un nettoyage très complets, réalisé à l'aide de « *Featuretools* » et « *Deep Feature Synthesis* ».
La donnée présente 15 colonnes, avec des caractéristiques clients variées tels que des sources externes, l'âge d'obtention de la première voiture, la date d'anniversaire, le nombre de jours employés et 307 511 observations.

La tâche à réaliser est une classification binaire. En effet, à partir de données en entrée, il faut prédire si le client va présenter un défaut de paiement (classe 1) ou non (classe 0).
Une des problématiques majeures est le déséquilibre des classes entre la répartition des clients en défaut de paiement contre les clients qui ne présentent pas de défaut de paiement. Ce rapport est de d'environ 9% pour la classe 1 contre 91% pour la classe 2.

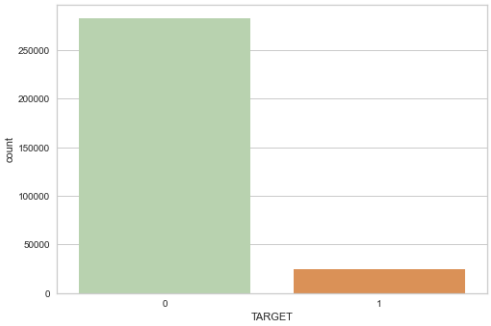

Figure 1 : distribution des clients en défaut de paiement (orange) et des clients sans défaut de paiement

1.2 Sélection des modèles :

La démarche à suivre classique en Machine learning est d'essayer en premier lieu des modèles dit « simples » puis tester des modèles plus élaborés.
J'ai donc choisi comme première approche de tester :

- Logistic Regression
- RandomForestClassifier
- XGBoost (google colab)
- LGBM Light (google colab)

Note : le notebook sélectionné propose aussi un RandomForest dont les hyper-paramètres sont « pré-affinés »).

1.3 Méthodologie d'entrainement :

Afin d'entrainer et d'évaluer le modèle en classification supervisée, il est préconisé de séparer la donnée en :

- donnée d'entrainement (généralement 75% de la donnée) afin d'entrainer le modèle
- donnée de test afin d'évaluer le modèle

Au vu du déséquilibre inhérent à notre donnée, un risque d'une répartition non homogène des classes est possible (par exemple très peu de classe 1 dans la donnée de test). Pour palier à cette problématique, j'ai utilisé un algorithme de validation croisée stratifié, avec un 5 répétitions. Cette algorithme a pour avantage de préserver la répartition des classes et donc évite de créer un lot ou la donnée de test serait erronée. De plus, le k_fold a 5 permet de répéter 5 fois l'opération afin de diminuer le biais et renforcer la légitimité des métriques obtenues.

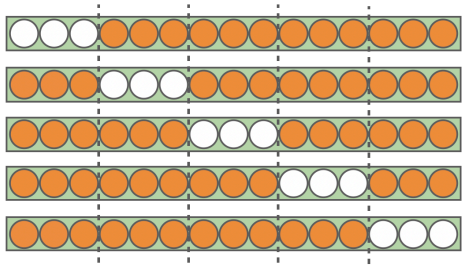

Figure 2 : représentation de la validation croisée stratifiée. En orange donnée d'entraînement, en blanc donnée de test

1.4 Optimisation du modèle :

Plusieurs approches ont été testées pour palier au déséquilibre des classes:

- oversampling, à l'aide d'un algorithme nommé SMOTE. Les résultats sont assez mitigés. De plus, changer la donnée en créant de nouveaux points peut amener un biais.
- undersampling, en diminuant le nombre de clients qui ne présente pas de défaut de paiements. Cette approche a également un impact très faible sur les résultats. De plus, cela entraîne de la perte de données, donc d'informations.
- modifier le threshold de la probabilité. Ceci permet de mettre plus de points dans la classe « client en défaut », et donc de mieux répondre à la problématique. Cependant, si on augmente trop le threshold, on a tendance à catégoriser trop de client en « défaut ».. Il faut donc ajuster grâce aux métriques afin d'obtenir le meilleur résultat.

Une fois un modèle et une méthodologie définis, un gridsearching est réalisé pour attribuer les meilleurs hyper-paramètres au modèle. L'optimisation se fera sur la métrique sélectionnée (cf paragraphe suivant). Une validation croisée est effectué par l'algorithme (ici 5 validations) afin d'obtenir des résultats robustes.

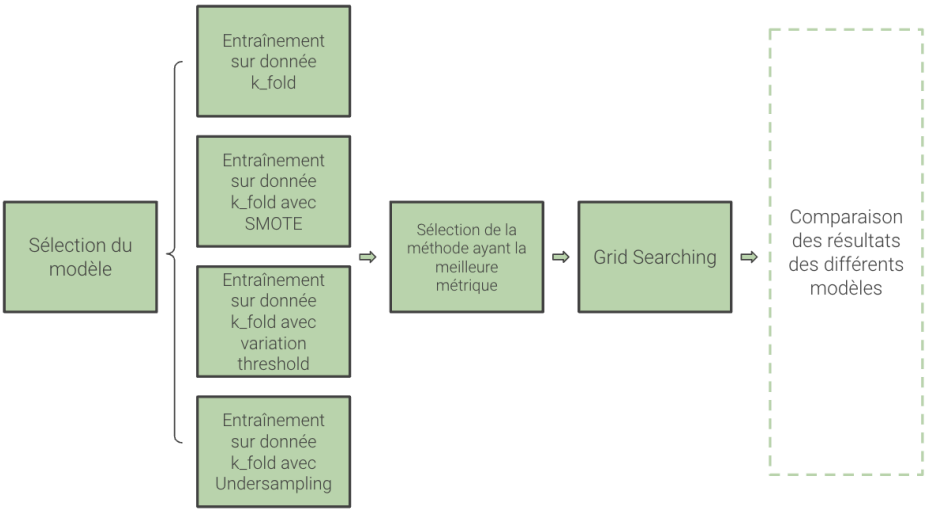

Figure 3 : résumé de la méthodologie d'entraînement

2. FONCTION-COÛT METIER, MÉTRIQUE ET OPTIMISATION

Afin de répondre à la problématique, il semble important de prédire qu'un client à risque appartient à la classe 1 pour éviter un crédit en défaut de paiement. Cependant, il s'agit de garder en tête qu'on ne veut pas être trop sévère afin de ne pas perdre de client qui ne présenterai pas de défaut de paiement. Pour visualiser la problématique, il peut être intéressant de regarder la matrice de confusion.

	Prédiction clients non défaut (0)	Prédiction clients en défaut (1)
Clients non défaut (0)	TN	FP
Clients en défaut (1)	FN	TP

Figure 4 : matrice de confusion avec TN : True Negative, FN : False Negative, FP : False positive, TP : True positive

L'objectif est donc de diminuer les FN. Cependant, il faudra faire attention de ne pas classer tous les clients en classe 1. Il faut donc mettre en regard les FP.

La métrique qui exprime les FN est le recall.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

La métrique optimisant les FP est la précision.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Il est donc intéressant de suivre ces métriques, mais le plus intéressant serait un compromis entre les deux: le F1 score. Cependant, le F1 score ne permet pas de mettre plus en avant le recall.

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

Le **F ß score** a été créé pour répondre à cette problématique. L'hyper-paramètre sera fixé à 2 afin de « privilégier » le recall par rapport à la précision, tout en gardant cette deuxième dans la balance. C'est donc sur cette métrique que la sélection du modèle a été réalisée.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

L'accuracy peut aussi être regardée, mais ce ne sera pas la métrique d'évaluation du modèle.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

L'optimisation du modèle afin de trouver un équilibre entre client à accepter et à refuser ce fera donc sur le F ß score.

Le modèle final sélectionné est le RandomForestClassifier *« pré-affiné »*, sans réalisation d'oversampling ou d'undersampling. En effet, le F ß2 score de ce modèle est de 0,4 et l'accuracy de 0.68. Le choix s'est porté non seulement sur le F ß score, mais aussi sur le recall.
A noter que le XGBoost dispose de performance un peu plus élevées, mais c'est un modèle plus couteux en performance qui ne fonctionne pas sur mon poste de travail.

3. INTERPRÉTABILITÉ

L'interprétabilité d'un modèle de machine learning est importante afin de pouvoir comprendre les résultats. L'interprétabilité (« comment on arrive au résultat ») et explicabilité (« pourquoi on a ce résultat? ») sont étroitement liées. Dans notre cas, il s'agit de dire quelles sont les caractéristiques du client qui font le plus pencher la balance vers un refus ou une acceptation du prêt.

Remarque : ces notions d'interprétabilité/explicabilité sont de plus en plus présentes au fur et à mesure que le machine learning prend de la place dans notre société. Cela permet d'éviter des biais discriminatoires qui pourrait se glisser (par exemple ici refuser plus de prêt aux hommes qu'aux femmes ou inversement). La notion d'éthique et d'explicabilité sont de plus en plus discutées, notamment dans la data science appliquée au domaine de la santé.

3.1 Interprétabilité globale :

Basiquement , elle correspond à « combien la caractéristique est utilisée dans chaque arbre de la forêt. Formellement, il est calculé comme la réduction totale (normalisée) du critère apportée par cette caractéristique. » (source).

Elle permet d'estimer pour chaque colonne si cette dernière a une grande influence sur le modèle ou une influence moindre. On peut dire dans notre cas que les trois variables ayant le plus d'impact sont EXT_SOURCE 3, 2 et 1.

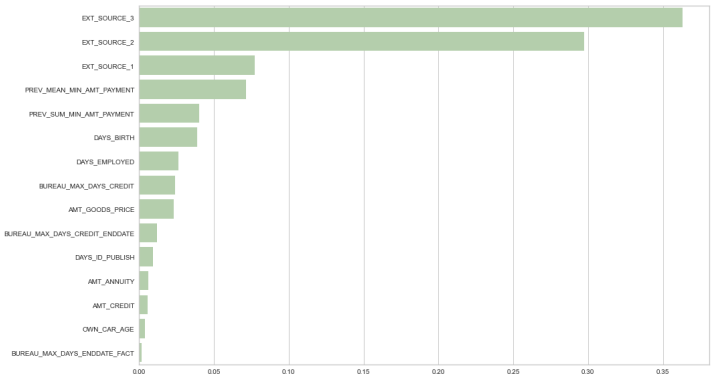

Figure 5 : représentation graphique de l'interprétabilité globale

3.2 Interprétabilité locale :

L'interprétabilité locale a pour objectif de comprendre les raisons de l'attribution d'un prédiction pour un client. Pour cela, j'ai utilisé Shap (SHapley Additive exPlanations). Le but de Shap est d'expliquer la prédiction d'une instance x en calculant la contribution de chaque caractéristique à la prédiction.

Ainsi, en saisissant un client, Shap permet de décrire un « *force plot* » qui permet de comprendre ce qui influe le plus la prédiction. Les caractéristiques ayant tendance à diminuer les probabilités sont en bleues, alors que celle en rouge ont tendance à l'augmenter. La « force » de la caractéristique est proportionnelle à la taille.

Remarque : en moyennant les valeurs absolues des valeurs de Shap pour chaque variable, nous pouvons remonter à l'importance globale des variables.

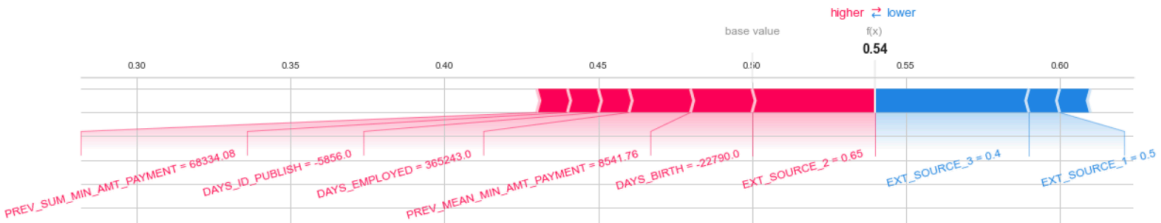

Figure 6 : représentation graphique de l'interprétabilité locale pour un client prédit « non défaut »

4. LIMITES ET AXES D'AMÉLIORATION

Bien que présentant des résultats supérieur à la baseline, la modélisation effectuée est loin d'être parfaite. La première et principale piste d'amélioration serait de travailler avec une équipe métier afin d'aiguiller les décisions. Toutes les améliorations proposées en deçà devrait passer par une analyse métier réalisée par l'équipe.

Axe donnée :

- Réaliser la partie sélection de la donnée afin de mieux comprendre la donnée que l'on manipule. En effet, lors de ce projet, cette partie est laissée de côté afin de ne pas avoir un projet trop chronophage. La sélection effectuée est assez drastique, et élimine certainement certaines caractéristiques qui pourraient être importantes.
- Réaliser une exploration de la donnée plus poussée : élimination de données aberrantes, mieux comprendre ce que l'on manipule.

Axe algorithme :

- Tester d'autres modèles plus complexe comme du deep learning
- Tester de finetuner sur sample_weight
- Tester des méthodes de bagging
- Tester les modèles sur plus de caractéristiques

Axe analyse du résultat :

- Prendre en compte d'autres paramètres : somme du crédit demandé par le client, crédit déjà en cours, … afin de proposer un résultat plus optimal.

Lien du site - ici

Lien du versioning - ici