Projet 7 : Implémenter un modèle de scoring

Note méthodologique

Contexte :

"Prêt à dépenser" est une société financière qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de crédit.

L'entreprise souhaite développer un outil de “scoring crédit” et un dashbord pour restituer les résultats de cette modélisation. L'objectif est de disposer d'un score (probabilité) qui permettra d'accorder ou pas un crédit et d'expliquer cette décision au client.

Les données utilisées pour ce projet sont constituées de 307 000 observations (clients) et de 121 variables.

Problématique :

Il s'agit d'un problème de classification avec deux classes déséquilibrées (9 % de clients en défaut contre 91 % de clients sans défaut de paiement). La notion de solvabilité d'un client sera définie selon un seuil de classification.

Pour une banque, il est important de minimiser le taux de faux positifs afin d'éviter des pertes financières engendrées par l'insolvabilité d'un client.

Afin d'intégrer ces contraintes, l'évaluation des différentes modélisations sera basée sur les métriques ROC_AUC et F_beta score.

La courbe ROC représente le taux de vrais positifs (TPR) par rapport au taux de faux positifs (FPR). La métrique ROC_AUC correspond à l'aire sous la courbe ROC, elle est comprise entre 0 et 1.

Un modèle de classification, dont les prédictions sont 100% fausses, a une ROC_AUC égale à 0; celui dont les prédictions sont correctes à 100 % a une ROC_AUC égale à 1.

La ROC_AUC évalue la qualité des prédictions du modèle quel que soit le seuil de classification choisi. Or, la problématique traitée intègre deux contraintes, le seuil de classification afin de déterminer la solvabilité d'un client et la minimisation des faux positifs afin de limiter les pertes financières. C'est pourquoi, la ROC_AUC sera complétée par le F_beta score dans l'évaluation des modèles.

Le F_béta score est une métrique qui permet de définir l'importance (le poids) que l’on souhaite accorder au Recall (rappel) ou à la Précision. Dans notre problématique, on cherche à maximiser le Recall.

Pour rappel, la matrice de confusion est la suivante :

	Clients prédits en défaut	**Clients prédits sans défauts**
Clients réellement en défaut	Vrais positifs	**Faux négatifs**
Clients sans défaut	Faux positifs	Vrais négatifs

Le Recall et la Précision :

$$Recall = \frac{vrais\ positifs}{frais\ positifs + faux\ négatifs} \qquad Precision = \frac{vrais\ positifs}{frais\ positifs + faux\ positifs}$$

Enfin, le F_beta score :

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

Dans la formule mathématique du F_beta score, le paramètre beta est un paramètre de pondération. La valeur de beta doit être fixée en collaboration avec les équipes métiers.

En effet, il faudra estimer le coût moyen d'un défaut de paiement (coût de faux positifs) et le coût d'opportunité d'un client potentiel écarté à tort (faux négatif). On peut ainsi calculer deux coefficients : un coefficient Recall et un coefficient précision. Béta serait, par exemple, égal au rapport entre ces deux coefficients.

Dans le cadre de ce projet, nous n'avons pas de consignes quant aux coûts de défaut et d'opportunité. C'est pourquoi, le travail de modélisation a été élaboré sous l'hypothèse : béta=3.

Méthodologie d'entraînement du modèle

Les étapes de préparation des données et de features engineering ont été réalisées sur la base des kernels Kaggle consultables sur les liens suivants : 1,2,3.

Plusieurs classiffiers ont été testé afin d'en sélectionner le meilleur selon les métriques ROC_AUC et F_beta score. Ces modèles sont : RandomForestClassifier, LGBMClassifier, XGBClassifier et la régression logistique. Ils ont été entrainés via une validation croisée (5 folds) avec leurs hyperparamètres définis par défaut.

L'optimisation des hyperparamètres du modèle choisi a été réalisée via une validation croisée (5 folds). Enfin, le modèle a été évalué, toujours selon les métriques ROC_AUC et F_beta score, avec ces meilleurs hyperparamètres.

Traitement des classes déséquilibrées :

Le déséquilibre des classes détériore la qualité de prédiction du modèle puisque à l'étape d'entrainement, l'apprentissage se fera au détriment de la classe minoritaire (clients en défaut) qui sera mal détectée par le modèle à l'étape de prédiction.

Trois méthodes de traitement des classes déséquilibrées ont été testées : SMOTE, SMOTE&RandomUnderSampler et ADASYN. L'évaluation de ces méthodes, combinées au meilleur modèle (LGBMClassifier), a été réalisée sur la base des métriques ROC AUC et F_beta score.

La méthode SMOTE (Synthetic Minority Oversampling Technique) consiste à générer, selon un taux d'échantillonnage, des données synthétiques très proches (très similaires) des données de la classes minoritaire.

La méthode RandomUnderSampler vise à réduire les données de la classe majoritaire en supprimant des observations de cette classe.

La combinaison des méthodes SMOTE et RandomUnderSampler peut s'avérer efficace pour traiter le déséquilibre des classes.

Enfin, la méthode ADASYN (Adaptive Synthetic Sampling), selon cette article, consiste à générer des données synthétiques de la classes minoritaire de manière "adaptative" en fonction de leurs distributions.

"This paper presents a novel adaptive synthetic (ADASYN) sampling approach for learning from imbalanced data sets. The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn. As a result, the ADASYN approach improves learning with respect to the data distributions in two ways: (1) reducing the bias introduced by the class imbalance, and (2) adaptively shifting the classification decision boundary toward the difficult examples. Simulation analyses on several machine learning data sets show the effectiveness of this method across five evaluation metrics."

Dans ce projet, la combinaison des méthodes SMOTE et RandomUnderSampler donne les meilleures métriques.

Interprétabilité du modèle

Le modèle étant destiné à des équipes opérationnelles devant être en mesure d'expliquer les décisions de l'algorithme à des clients réels, le modèle est accompagné d'un module d'explicabilité.

La méthode Shap (SHapley Additive exPlanations) a été utilisée afin de connaître, globalement, les variables les plus influentes sur les prédictions du modèle. De la même manière, Shap permet d'identifier, localement, les variables influant la prédiction pour une observation (un client) donnée.

Limites et améliorations possibles

La modélisation réalisée a été effectuée sur la base d'une hypothèse forte : la définition d'une métrique d'évaluation, le F Beta Score avec Beta fixé suivant certaines hypothèses non confirmées par les équipes métier.

L'axe principal d'amélioration serait de définir plus finement la métrique d'évaluation en collaboration avec ces équipes.

De la même manière, le seuil de solvabilité d'un client a été fixé de manière arbitraire. Il conviendrait de le fixer après consultation des équipes métier.

Enfin, les étapes de traitements préalables à la modélisation ont été réalisées en réutilisant un notebook issu de Kaggle. Il y a très probablement l'opportunité d'améliorer la modélisation en affinant le travail de features engineering en collaboration avec les équipes métier.