Note méthodologique N°2 - Optimisation du modèle

Fonction coût métier & métrique d'évaluation

L'objectif du projet est de résoudre un problème de classification. Il s'agit de prévoir si un crédit appartient à la classe 0 (crédit accepté) ou bien à la classe 1 (crédit refusé).
Nous sommes face à un problème d'optimisation dont les contraintes sont les suivantes:

- Le principal objectif est de minimiser le nombre de crédits prévus comme appartenant à la classe 0 (crédits acceptés) alors qu'ils appartiennent en fait à la classe 1 (ne seront pas remboursés). Il s'agit donc de **minimiser le nombre de faux positifs.**
- Compte tenu du caractère commercial l'activité de la société Prêt à dépenser, il faut également chercher à minimiser le taux de faux négatifs, c'est à dire le nombre de crédits prédits comme appartenant à la classe 1 (crédits refusés) alors qu"ils appartiennent en réalité à la classe 0 (ils auraient été remboursés si le crédit avait été accepté). L'objectif secondaire est donc de **minimiser le nombre de faux négatifs**.

En termes techniques, le data scientist doit chercher un modèle permettant de **maximiser le recall**. Il s'agit de trouver un modèle permettant de prévoir le plus grand nombre de crédits appartenant à la classe 1.

$$Recall = \frac{nombre\ de\ crédits\ correctement\ prédits\ comme\ \ appartenant\ à\ la\ classe\ 1}{nombre\ total\ de\ crédits\ appartenant\ effectivement\ à\ la\ classe\ 1}$$

De manière secondaire, il faut également chercher à minimiser le taux de faux négatif afin de ne pas perdre trop de clients solvables. Celà revient donc à maximiser la surface sous de la courbe ROC (AUC ROC) ainsi que le score F1.

Algorithme d'optimisation

Parmi les algorithmes testés affichant de bonnes performances (voir note N°1) , nous avons charché à optimiser les hyperparamètres de régression logistique en utilisant les fonctions RandomizedGridSearchCV et GridsearchCV de scikit learn. Les régressions logistiques présentent plusieurs avantages. Elles permettent d'obtenir de bons scores de recall, sont facilement compréhensible et présentent l'avantage d'être rapide à entraîner.
Néanmoins, les modèles XGBoost permettent d'atteindre des performances supérieures en termes de recall et d'AUC ROC curve pour des temps de calcul convenables comparé aux Random Forests.
Au final, nous sommes parvenus à atteindre un recall sur le test set d'environ 0.80 et un AUC ROC Curve de même niveau. Le score F1 est autour de 0.76.
Nous avons également cherché à optimiser le seuil de probabilité à utiliser pour prévoir la classe 1. Par défaut, ce seuil est fixé à 50%. Avec un modèle XGBoost optimisé, nous avons réussi à légèrement améliorer la performance du modèle avec un seuil très légèrement inférieur (49.55%).