Note METHODOLOGIQUE

1. Contexte

Ce document est un des livrables du projet «Implémentez un modèle de scoring » du parcours Data Scientist d'Openclassrooms. Le projet consiste à développer pour la société « Prêt à Dépenser », une société de crédit de consommation, un modèle de scoring de la probabilité de défaut de paiement d'un client. Le jeu de données utilisé pour ce projet est une base de données de 307 000 clients comportant 121 features (âge, sexe, emploi, logement, revenus, informations relatives au crédit, notation externe, etc.)

2. Méthodologie d'entraînement du modèle

Le modèle mis en place a été entraîné après analyse exploratoire et création de nouvelles features. Le jeu de données initial a été séparé en plusieurs parties de façon à disposer : D'un jeu de training (75% des individus) qui a été séparé en plusieurs folds pour entraîner les différents modèles et optimiser les paramètres (cross validation) sans overfitting. D'un jeu de test (25 % des individus) pour l'évaluation finale du modèle

C'est un problème de classification binaire avec une classe sous représentée (9 % de clients en défaut contre 91 % de clients sans défaut). Ce déséquilibre des classes doit être pris en compte dans l'entraînement des modèles d'ou la mise en place d'un Under Sampling.

Deux modèles ont été testés avec recherche d'hyperparamètres avec GridSearchcv :

- Logistic Regression
- Decision Tree

3. Fonction coût, algorithme d'optimisation et métrique d'évaluation

Dans le jeu de données de base, il y a une part de 92 % des clients qui n'ont pas d'incident de paiement, tandis que 8 % des clients ont eu des incidents.

Du point de vue d'une banque, on cherchera à éviter de mal catégoriser un client avec un fort risque de défaut (pertes financières et frais de recouvrements qu'on imagine importants). On cherche donc à minimiser le pourcentage de faux négatifs et à maximiser le pourcentage de vrais positifs. Autrement dit, on va chercher à maximiser le recall. Par ailleurs on cherche à maximiser le nombre de clients potentiels donc à ne pas tous les classer en défaut. On essaiera donc d'éviter d'avoir un trop grand nombre de faux positifs. On cherche donc à maximiser la précision.

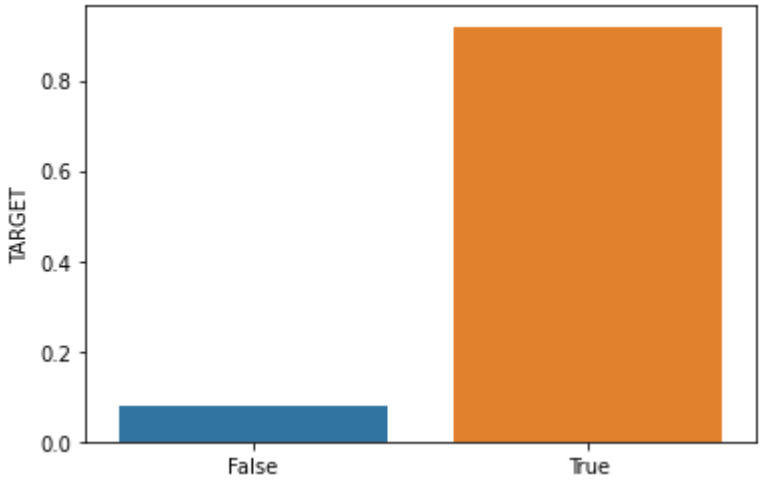
Pour notre problématique métier, le RECALL est plus important que la PRECISION car on préfèrera vraisemblablement limiter un risque de perte financière plutôt qu'un risque de perte de client potentiel. On cherche donc une fonction qui optimise les 2 critères en donnant plus d'importance au recall. Fonction permettant de faire cela : F Beta Score : https://en.wikipedia.org/wiki/F1_score) avec Beta le coefficient d'importance relative du recall par rapport à la précision.

Algorithme d'optimisation

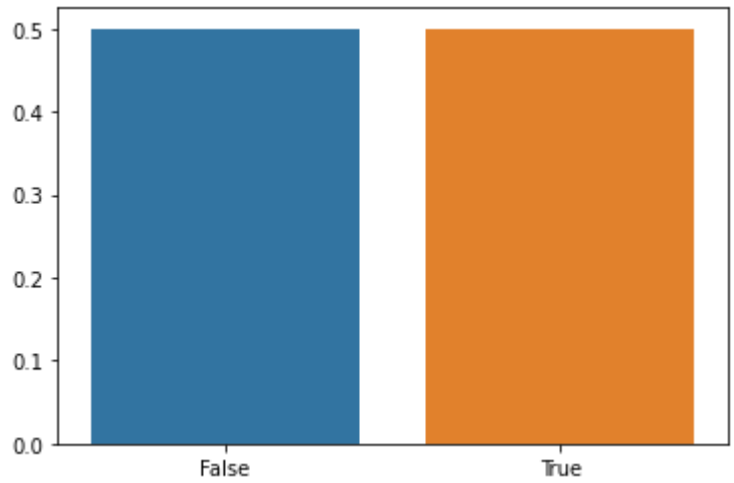
Tout d'abord, nous avons réalisé une ANOVA pour la sélection des 50 features avec f-score élevé.

Une fois la feature selection effective, nous avons mis en place un Under Sampling avec la méthode RandomUnderSampler qui nous permet d'échantillonner n'importe quelle classe aléatoire avec ou sans remplacement, ce qui équilibre nos données.

Avant



Après



La meilleure combinaison d'hyperparamètres a été retenue pour le modèle de Logistic Regression. Le modèle a obtenu un meilleur score en cross validation. Il dispose dispose d'une bonne performance avec un ROC AUC: 73.655%

4. 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.

L’ANOVA nous a permis de connaitre les 50 meilleurs features. Donc interpréter notre modèle équivaut juste à expliquer l’importance et l’utilité de ces features.

5. Limites et améliorations possibles

La modélisation effectuée dans le cadre du projet a été effectuée sur la base d’une hypothèse forte : la définition d’une mérique d’évaluation : le F Beta Score avec Beta fixé suivant certaines hypothèses non confirmées par le métier. L’axe principal d’amélioration serait de définir plus finement la métrique d’évaluation en collaboration avec les équipes métier. L’interprétabilité du modèle pourrait être étoffée en considérant les variables issues du one hot encoding comme une seule et même variable dans la perturbation (un client ne pouvant cumuler plusieurs caractéristiques dans la logique du jeu de données initial. Ajouter les autres d’autres données en plus d’Application_train , bureau balance et bureau pourrait nous permettre d’avoir un modèle plus performant.