NOTE METHODOLOGIQUE

CAHIER DES CHARGES

Prêt à dépenser est une société financière d'offre de crédit à la consommation pour la clientèle ayant peu ou pas d'historique de prêt.

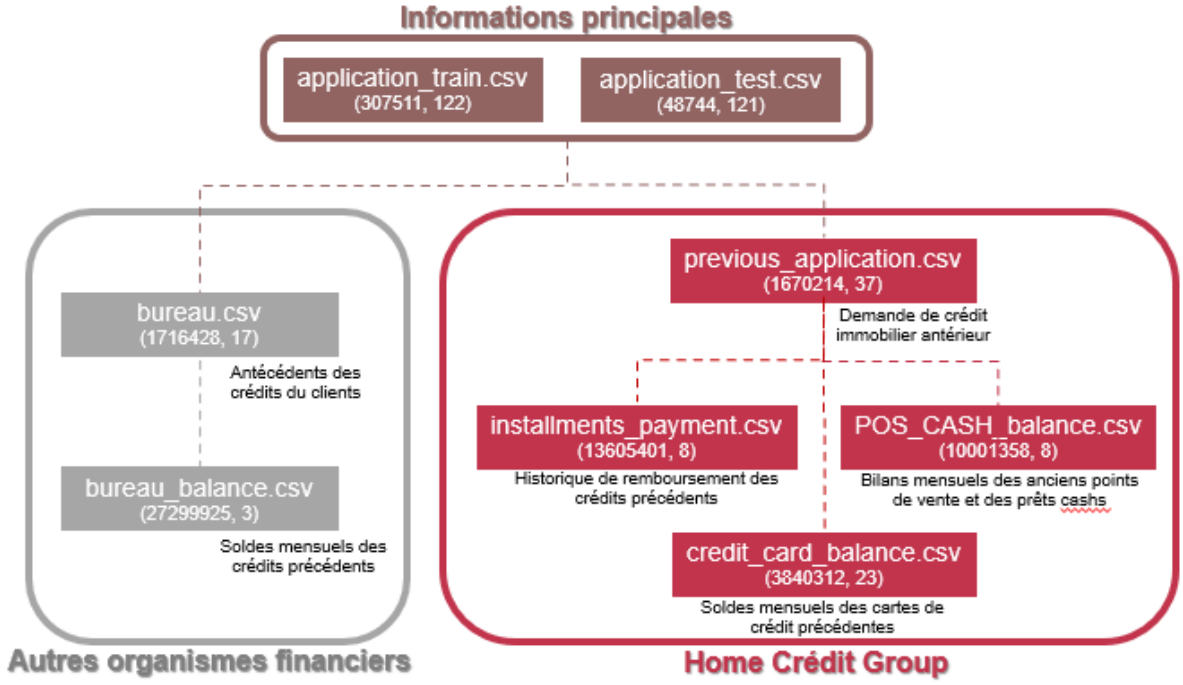
Notre mission est de développer un modèle de scoring de la probabilité de défaut de paiement du client pour étayer la décision d'accorder ou non un prêt à un client potentiel en s'appuyant sur des sources de données variées.

Le développement d'un dashboard interactif permettra aux chargés de clientèles d'expliquer avec transparence la décision d'octroi ou non du prêt et de mettre à disposition des clients l'exploration de leurs informations personnelles.

PRESENTATION DES JEUX DE DONNEES

Jeux de données fournis par HOME CREDIT GROUP

Huit fichiers au format csv sont fournis et composent notre jeu de données. Ils contiennent 218 informations bancaires et personnelles anonymisées pour 307511 clients recueillies auprès de Home Crédit Group et auprès d'autres institutions financières.



INTERPRETATION DE LA PROBLEMATIQUE :

La variable cible à prédire prend 2 valeurs et est fortement déséquilibrée (8/92) :

0 – Positive – Non défaillant :

Indique que le client a totalement remboursé son prêt

1 – Négative – Défaillant :

Indique que le client n'a pas remboursé son prêt en totalité ou en partie

PRE TRAITEMENT DES DONNEES :

Chaque fichier doit être prétraité pour que leurs données puissent être consommées par le modèle (seulement des données numériques) et rassemblées dans un seul fichier ne contenant que les variables les plus pertinentes.

La première étape consiste à transformer les types des objets pour les rendre cohérents (ex : Y/N en 0/1) et réduire leur taille de stockage dans la mémoire.

La seconde consiste à corriger les valeurs aberrantes détectées lors de l'analyse exploratoire (EDA) et d'harmoniser les valeurs uniques des données (ex : sexe Masculin, Féminin) pour les informations principales. L'analyse exploratoire a montré que plus d'un tiers des variables contiennent plus de 50% de valeurs manquantes. La double stratégie a été :

• Étape 3 : de supprimer les variables avec plus de 68% de valeurs manquantes pour conserver les variables importantes détectées lors de l'EDA et de supprimer les variables sans information pour le modèle.

• Étape 4 : d'imputer donc de remplacer les variables conservées ayant des valeurs manquantes par la valeur médiane pour toutes les variables numériques et par la valeur la plus utilisée pour les variables qualitatives.

La cinquième étape consiste à créer de nouvelles variables qui peuvent augmenter la performance du modèle selon 2 axes :

- Création automatique à partir des variables numériques en ajoutant la moyenne, le minimum, le maximum, compter leur nombre, l'écart-type…
- Création manuelle à partir de la compréhension du métier en combinant les variables. Par exemple :
    - Différence : membres de la famille - enfants (trouver les adultes)
    - Ratio : Revenu du demandeur / membres de la famille : revenu par tête
    - Différence : Revenu du demandeur - Annuité de prêt
    - Somme : flag téléphone portable ? + flag téléphone professionnel ? + flag téléphone professionnel fixe ? + flag téléphone portable joignable ? + flag adresse de messagerie électronique ?
    - Moyenne : moyenne des scores des 3 sources externes
    - …

Une fois toutes ses variables ajoutées, la sixième étape consiste à encoder les variables qualitatives pour les transformer en numériques seulement consommables par le modèle. Deux types d'encodage ont été utilisés :

- LabelEncoder : pour les variables contenant peu de valeurs différentes (ex : sexe : Féminin transformé en 0 et Masculin en 1).
- OneHotEncoder : pour les variables contenant moins de 15 valeurs différentes (ex : variable 'couleur' qui contient rouge, bleu et vert, si on crée 3 variables 'couleur_rouge', 'couleur_bleue' et 'couleur_verte', si la première ligne contient la couleur bleue, on forcera 0 'dans couleur_rouge' et 'couleur_verte' et 1 dans 'couleur_bleue' puisque la couleur est bleue…)
- Certains modèles de machine learning basés sur les distances sont sensibles aux variations d'échelle des différentes variables. Il faut alors standardiser les données en ramenant l'ensemble des valeurs d'une variable entre 0 et 1.

Une fois le nettoyage et le feature engineering terminés, la cinquième étape consiste à assembler tous les fichiers en un seul fichier en utilisant les différents identifiants lors de cette fusion, le but étant que chaque client ne soit représenté que dans une seule ligne

de ce fichier. Le nombre de variables après assemblage passe à 615 variables. Cela constitue un modèle très complexe qui peut pénaliser la performance du modèle, une phase de sélection des variables est alors nécessaire.

La sélection de variables consiste, étant donné des données dans un espace de grande dimension, à trouver un sous-ensemble de variables pertinentes. Il faut donc minimiser la perte d'information venant de la suppression de toutes les autres variables.

Plusieurs techniques ont été utilisées :

Filtrage : suppression des variables colinéaires.

Embedded : des méthodes intégrées qui apprennent quelles variables contribuent le mieux à la précision du modèle pendant sa création. Une valeur est calculée et liée à chaque variable du jeu de données servant à entraîner le modèle.

Automatiques : basées sur des librairies python (Boruta, BorutaShap, RFECV, permutation importance avec scikit-learn ou eli5)

MODELISATION :

CHOIX DES METRIQUES D'EVALUATION DE LA PERFORMANCE DU MODELE :

Le choix de la métrique est primordial, dépend de la problématique et permet d'évaluer la performance du modèle prédictif et de garantir la qualité du modèle de classification. Ces métriques permettent de comparer les classes réelles aux classes prédites par le modèle. Pour notre problématique, nous devons minimiser les pertes d'argent pour notre société financière, notre modèle doit donc :

- ne surtout pas prédire un client non-défaillant s'il est défaillant il faut minimiser le nombre de faux négatifs (erreur de type II) (prédit non-défaillant mais client défaillant dans la réalité). Dans ce cas, le groupe Home Crédit aura perdu toute la somme prêtée à l'emprunteur. Cela constitue les plus grosses pertes pour l'entreprise.

- s'efforcer de ne pas prédire en défaillant un client non défaillant il faut minimiser les faux positifs (erreur de type I) (client prédit défaillant alors que non- défaillant dans la réalité). Dans ce cas, le groupe Home Crédit aura seulement perdu les intérêts de la somme qu'il aurait prêté à l'emprunteur.

Les métriques Recall et Fbeta10 seront utilisées pour être maximales pour notre modélisation et privilégiées à la métrique Précision.
Une métrique bancaire métier a été testée mais ne donnent pas de résultats satisfaisants.

REEQUILIBRAGE DE LA VARIABLE CIBLE :

Une classe déséquilibrée peut avoir un impact négatif sur le modèle qui aura tendance à prédire la classe majoritaire (donc client non défaillant). Une modification de l'ensemble de données est possible avant d'entraîner le modèle prédictif afin d'équilibrer les données : le rééchantillonnage (re-sampling).

Deux méthodes principales existent pour égaliser les classes :

- le sur-échantillonnage (Oversampling)
- et le sous-échantillonnage (Undersampling).

Pour notre modélisation des tests ont été effectués :

- avec la librairie SMOTE et ses extensions BorderLineSMOTE et ADASYN,
- avec l'hyperparamètre class_weight du modèle LightGBM.

Le rééquilibrage via SMOTE ne donnant pas les résultats attendus, le rééquilibrage via l'hyperparamètre du modèle sera utilisé.

SEPARATION DES DONNEES EN ENTRAINEMENT/VALIDATION :

Le jeu de données est séparé en deux :

- en un jeu d'entraînement (80%) servant à entraîner le modèle,
- et en un jeu de validation permettant d'évaluer la performance des différents modèles testés.

A noter : lors de la séparation, les 2 jeux de données devront conserver la répartition de départ des classes majoritaires (les clients non défaillants) et minoritaires (les clients défaillants).

CHOIX DU MODELE :

Pour se faire une première idée des modèles de classification performants, le jeu de données a été entraîné en automatique sur tous les modèles de classification de la librairie Pycaret installés sur la machine. Les résultats montrent que les modèles ensemblistes (Catboost, Xgboost et LightGBM) semblent être plus performants sur notre jeu de données. Le modèle LightGBM non optimisé est très rapide et obtient des résultats satisfaisants qui pourront être améliorés (optimisation par réglage des hyperparamètres) et réglés (réglage de la valeur de seuil minimal de probabilité pour faire basculer la prédiction vers les classes positives = client défaillant) pour maximiser la métrique F10, c'est ce modèle qui sera retenu pour être optimisé.

OPTIMISATION DES HYPERPARAMETRES DU MODELE LIGHTGBM :

La technique retenue pour l'optimisation des hyperparamètres du modèle LightGBM est l'optimisation Bayésienne avec 3 librairies différentes (bayes_opt du MIT, skopt de scikit-learn et optuna). L'optimisation bayésienne fonctionne en construisant une distribution postérieure de fonctions (processus gaussien) qui décrit au mieux la fonction que l'on veut optimiser. Au fur et à mesure que le nombre d'observations augmente, la distribution postérieure s'améliore, et l'algorithme devient plus certain des régions de l'espace des paramètres qui méritent d'être explorées et de celles qui ne le méritent pas. L'optimisation a été effectuée sur différentes métriques (Roc Auc, PR Auc, F10, Recall et la métrique métier) pour différents jeux de données (rééquilibrés avec smote, hyperparamètre class_weight de Lightgbm ou non équilibrés, standardisés ou non…). Le but minimiser le nombre de faux négatifs tout en prédisant le plus de vrais positifs possibles tout en limitant le nombre de faux positifs. Le modèle LightGBM avec les paramètres de base sert de comparatif.

Le modèle le plus performant est le modèle optimisé avec la méthode bayésienne Optuna, avec un rééquilibrage interne (hyperparamètre class_weight='balanced') et la métrique F10. Il détecte le moins de faux négatifs, le plus de vrais positifs mais un taux plus élevé de faux positifs. Un compromis est à faire entre le taux de faux négatifs et le taux de faux positifs, une augmentation de l'un entraîne une diminution de l'autre…une collaboration sera nécessaire pour décider du réglage du seuil de décision d'un client défaillant (fixé par défaut à 0.5) et de la proportion de faux positifs acceptables (non précisée dans le cahier des charges).

INTERPRETABILITE DU MODELE :

Le modèle LightGBM optimisé final contient une méthode permettant pour chaque variable de calculer l'importance de cette variable pour le modèle. Une fois normalisées, nous pouvons comparer l'importance relative de chacune des variables et par un simple tri, afficher les 15 premières variables les plus importantes.

Parmi ces variables nous retrouvons les variables les plus corrélées avec la variable cible détectées lors de l'EDA :

- les informations bancaires en particulier CREDIT_ANNUITY_RATIO (ratio du montant du crédit du prêt sur l'annuité de prêt font partie des informations), CREDIT_GOODS_RATIO (ratio du montant du prêt sur le prix réel du bien), BUREAU_CURRENT_CREDIT_DEBT_TO_CREDIT_RATIO_MEAN (le cumul des autres prêts en cours) …,
- les données externes : EXT_SOURCE_SUM, EXT_SOURCE_1 et EXT_SOURCE_2,
- les informations personnelles : CODE_GENDER (sexe du client), DAYS_BORTH (âge du client).

LIMITES ET AMELIORATIONS :

Pour répondre au problème de classification binaire à partir des 8 fichiers fournis par Prêt à dépenser, de nombreuses techniques de Machine Learning ont été nécessaires : rééquilibrage des classes, création de nouvelles variables facilement explicables, sélection des variables pour rendre le modèle moins complexe, choix des métriques adaptées à notre problématique métier, réflexion sur le compromis taux de faux négatifs et taux de faux positifs et sur le réglage du seuil de décision.

Néanmoins une collaboration avec notre client permettrait d'améliorer le modèle : quel est l'objectif du taux de faux négatifs/positifs ? cela nous permettrait d'affiner les hyperparamètres du modèle LightGBM et de trouver le seuil optimal pour atteindre ces objectifs. Les experts métiers pourraient nous aider à créer une métrique bancaire plus efficace et adaptée et pourraient nous donner leur avis sur l'intérêt des nouvelles variables créées et pourquoi pas nous indiquer de nouvelles variables. Une explication des données externe serait un plus puisqu'il est difficile d'être transparent en utilisant ces variables importantes pour le modèle mais inexplicables pour le client. Le Dashboard pourra également être évalué par le client et les remarques prises en compte, des améliorations techniques (cache mémoire des fonctions statiques, taille des graphiques affichés, couleurs, carte graphique du client ?) pourraient également être améliorées.

PRESENTATION DU DASHBOARD :

Dépôt des sources : https://github.com/davybayet/Projet7-Datascience

Accès : https://myscoringdashboard.herokuapp.com/