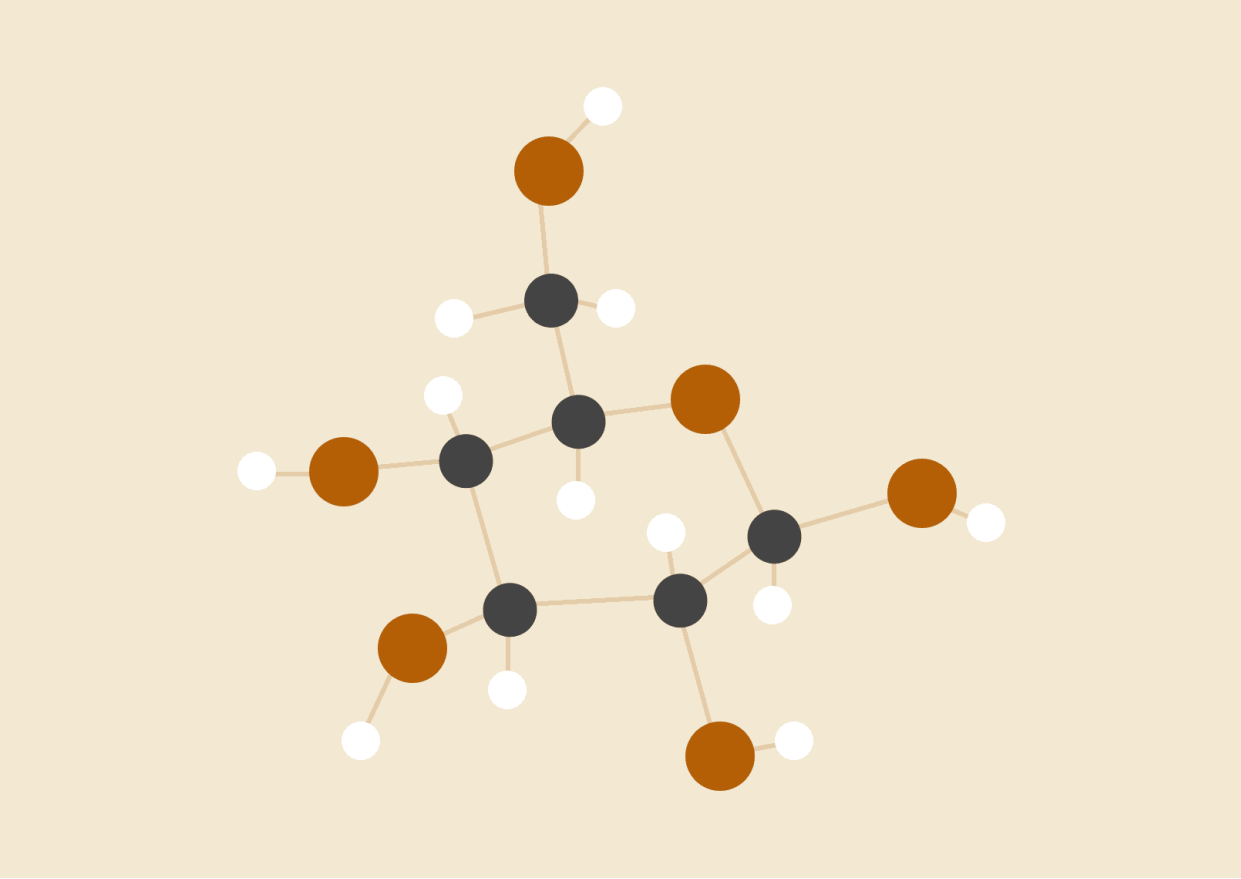
NOTE MÉTHODOLOGIQUE
PROJET 7

IMPLÉMENTEZ UN MODÈLE DE SCORING



Saidali Bacar Abdallah

02/2024

OPENCLASSROOMS

SOMMAIRE

INTRODUCTION

Présentation du sujet

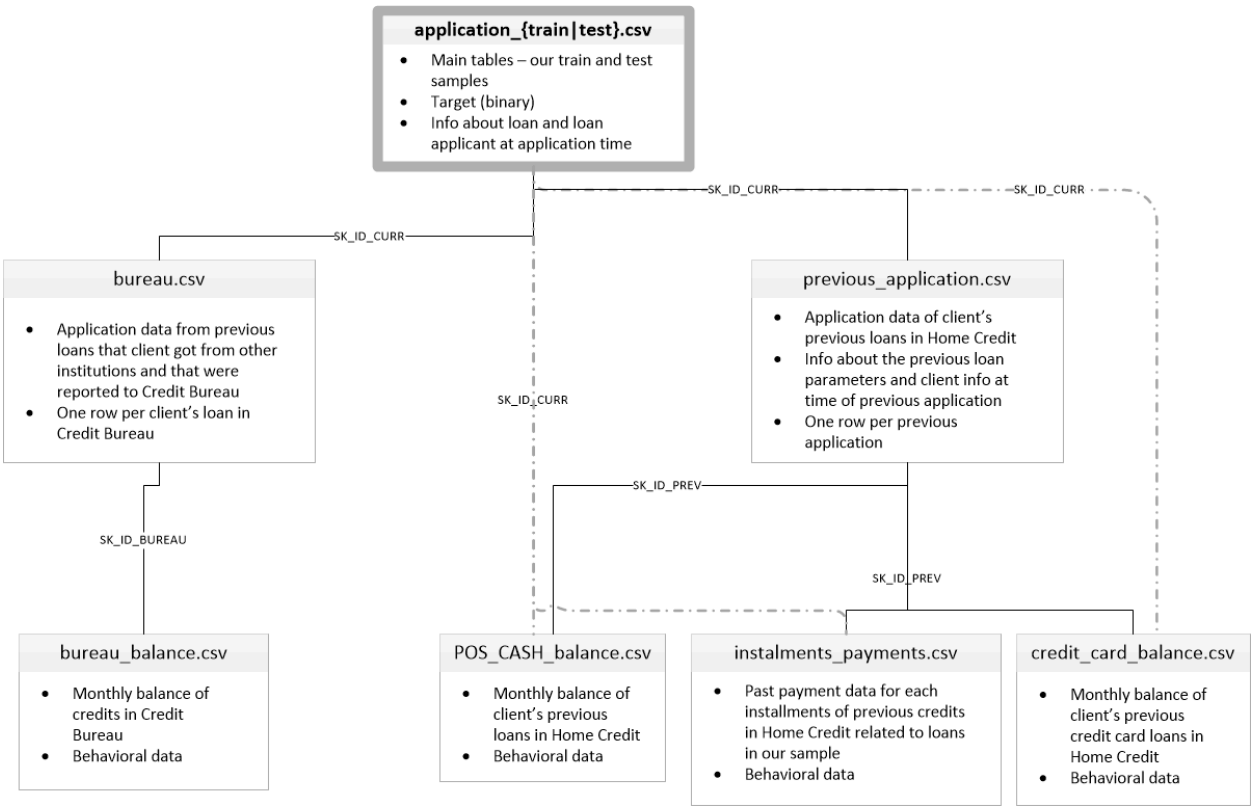
Position: Data Scientist au sein d'une société financière "Prêt à dépenser". Cette société propose des crédits à des personnes ayant peu ou pas de d'historique de prêt.

Mission: Elle consiste à construire un outil de **scoring crédit** dans l'objectif de déterminer la probabilité de remboursement d'un client et ensuite classifier la demande **en crédit accordé ou refusé**.

Contraintes: Dans un souci de transparence, on nous demande de construire un dashboard avec des paramètres clé et compréhensibles permettant d'expliquer au client le motif d'acceptation ou de refus de crédit.

Les données:

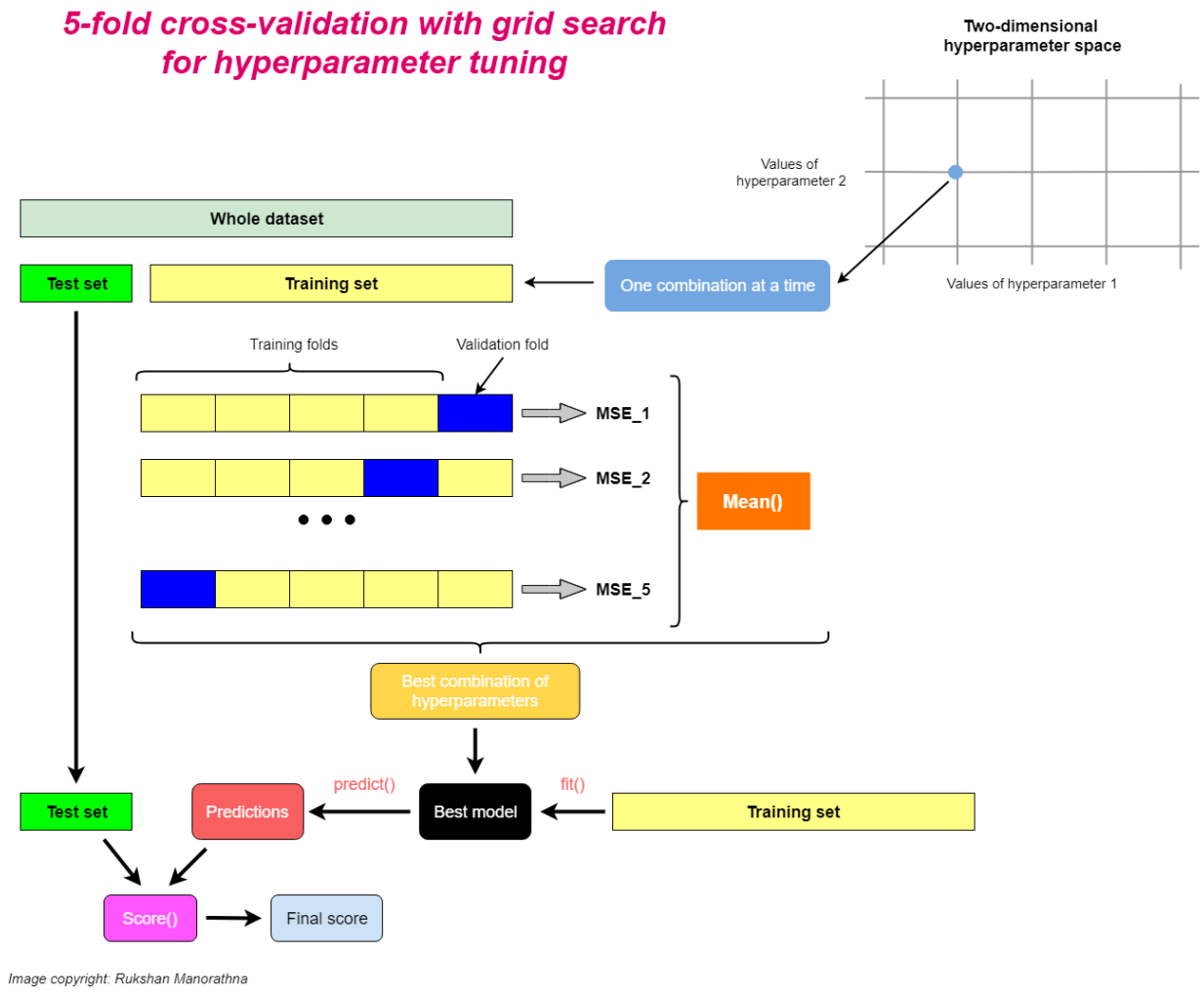
Un corpus de 9 fichiers organisé comme suit:



I. ENTRAINEMENT DU MODÈLE

Prérequis: choix du Kernel Kaggle, analyse exploratoire et création de nouvelles features.

Les données utilisées suite à ce processus initiale sont composées de 322060 clients et 189 features représentant (sexe, type d'emploi, catégorie de logement, niveau de revenu, montant du crédit en cours, etc…)



Les données:

Le jeu de données initial est partagé est jeu de données d'entraînement (60% des clients), de validation et de test, 20% des clients chacun.

Le jeu de données d'entraînement sert à l'entraînement des modèles, celui de validation à

valider les modèles et celui de test sert en toute fin comme un jeu de données neuf à l'évaluation modèle qui aura obtenu les meilleures performances.

Les modèles utilisés:

- Logistic regression
- Decision Tree Classifier
- RandomForest Classifier
- XGB Classifier
- LightGBM Classifier

Ici on a des modèles de type régression, arbre de décision et ensembliste.

GridSearchCV et pipeline:

Dans le pipeline on retrouve **SMOTE** et le modèle.

Mise en place d'un GridSearchCV avec 5 folds qui inclut le pipeline, param_grid, et le scoring.

II. TRAITEMENT DU DÉSÉQUILIBRE DES CLASSES

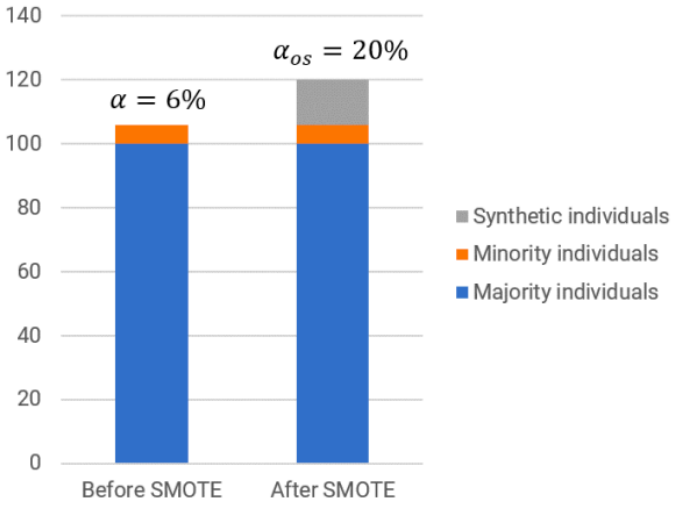
Notre problématique relève d'une classification, il s'agit ici de prédire la probabilité d'un client à rembourser son crédit. Dans notre jeu de données initial lorsque l'on regarde la variable TARGET (variable binaire avec 1 le client est capable de rembourser son crédit et 0 le client ne pourra pas rembourser son crédit). Il y a un grand déséquilibre de classe de l'ordre de 9% contre 91%.

Ce déséquilibre de classe doit être pris en compte lors de l'entraînement du modèle dans la mesure ou la sous représentation d'une classe par rapport à l'autre va induire une mauvaise prédiction sur cette même classe (dans notre cas, que le client soit capable de rembourser son crédit).

Quel outil ?

On utilisera ici la librairie SMOTE (Synthetic Minority Oversampling TEchnique) implémentée dans le package imbalanced-learn.

SMOTE est une méthode de suréchantillonnage des observations minoritaires, qui permet de générer de nouveaux individus minoritaires qui ressemblent aux autres, sans être strictement identiques. Proche de l'algorithme k-NN



Exemple simple: considérons des données avec 6 observations minoritaires et 100 majoritaires, ici SMOTE a généré 14 observations minoritaires.

De fait, SMOTE prévient de l'overfitting.

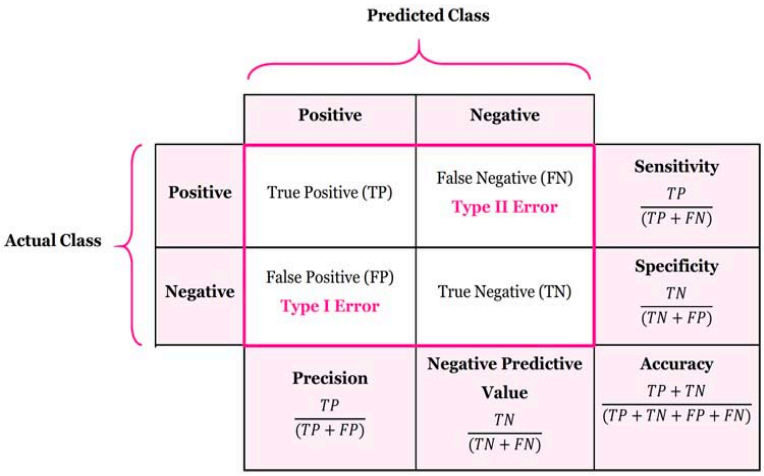
III. LA FONCTION COÛT MÉTIER

La fonction coût métier vient prendre en compte les spécificités de notre dataset et le contexte.

En effet, on considère l'hypothèse suivante: le coût d'un faux négatif est dix fois supérieur au coût d'un faux positif. Autrement dit, il est dix fois plus coûteux de prédire qu'un client a la capacité de rembourser son crédit qu'il ne l'est pas que l'inverse.

Pour implémenter cette hypothèse, on va utiliser la matrice de confusion et make scorer tous deux intégrer dans sklearn.metrics.

Rappel sur la matrice de confusion:



Le choix de la métrique sera donc le custom_scorer que nous aurons définis et intégré au GridSearchCV.

Du point de vue d'un créancier, on cherche à éviter de mal catégoriser un client à risque c'est-à-dire à maximiser le pourcentage de vrai positif (sensitivité ou recall) sans rogner sur la précision (on ne souhaite pas un trop grand nombre de faux positifs). Le recall est donc plus important que la précision.

L'algorithme d'optimisation: le modèle retenu est le lightgbm avec comme paramètre de boosting gbdt (gradient boosted decision trees). Il est basé sur trois grands critères: les arbres de décision, **l'optimisation du gradient** et la technique de boosting.

https://www.sicara.fr/blog-technique/mastering-lightgbm-unravelling-the-magic-behind-gradient-boosting

IV. LE TABLEAU DE SYNTHESE DES RESULTATS

Les résultats:

Modèles	Custom Score	ROC-AUC	Accuracy	Recall	Precision	F1-Score
Logistic Regression	-27125	0.58	0.46	0.66	0.08	0.15

Decision Tree Classifier	-25632	0.69	0.84	0.20	0.12	0.15
Random Forest Classifier	-22240	0.69	0.8	0.37	0.15	0.21
LightGBM Classifier	-26613	0.77	0.93	0.04	0.47	0.08

V. INTERPRÉTABILITÉ GLOBALE ET LOCALE DU MODÈLE

L'interprétabilité des modèles est très importante en Machine Learning. En effet, il faut expliquer et justifier la prise de décision induite par le modèle prédictif.

Une des questions fondamentales est d'identifier les variables qui ont influé la prédiction de notre modèle. Le jeu d'entraînement permet de mettre en avant le rôle des variables dans l'étape de la modélisation alors que le jeu de test permet de souligner l'influence de la variable lors du déploiement du modèle (généralisation).

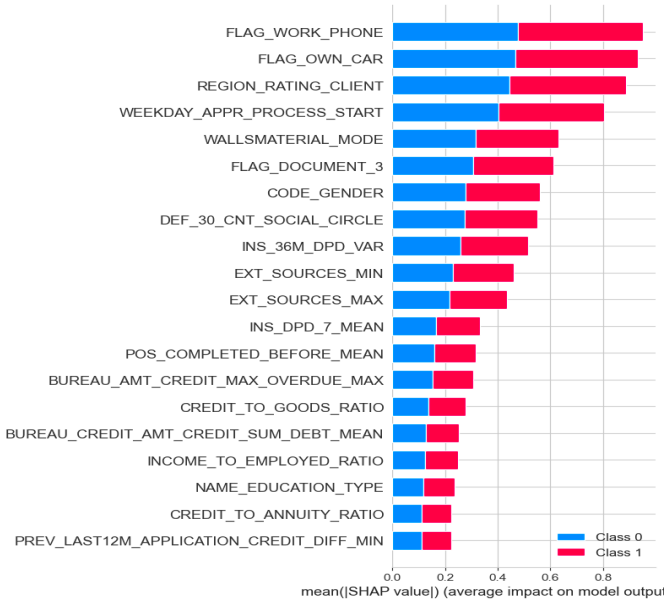
La librairie utilisée ici est la librairie SHAP (SHapley Additive exPlanations). Elle est basée sur les valeurs de Shapley (Shap values) en comparant ce qu'un modèle ce qu'un modèle prédit avec et sans cette variable.

Interprétabilité globale.

On cherche à expliquer le modèle dans sa globalité. Autrement dit, quelles sont les variables les plus importantes en moyenne pour le modèle.

Interprétabilité locale.

Elle consiste à expliquer la prévision f(x) d'un modèle pour un individu donné. Dans notre cas on souhaite savoir pourquoi la demande de crédit a été acceptée ou refusée?



VI. LES LIMITES ET LES AMÉLIORATIONS POSSIBLES

Les limites:

- La connaissance métier, le milieu bancaire

Chaque domaine a son propre vocabulaire et ses propres spécificités et le milieu bancaire n'est pas en reste. Cela peut influer directement sur la sélection des variables du point de vue "métier". Il aurait été intéressant de disposer de disposer d'un véritable glossaire des variables.

- La profusion de nombre de variable
- Le choix et l'optimisation du modèles

Le choix et l'optimisation du modèle se base sur une hypothèse forte concernant la métrique d'évaluation en rapport avec la fonction coût métier. Une discussion plus approfondie avec les

équipes de “Prêt à dépenser” serait la bienvenue.

Les améliorations possibles:

- Collaboration avec les équipes métier
- Dashboard

Il serait sans doute intéressant de développer un dashboard avec une page “client” et une page “banque”.

De plus, il serait également intéressant de rajouter un onglet intéractif dans la partie “client” de simulation et de pouvoir tester différents scénarios de crédit.

VII. L’ANALYSE DU DATA DRIFT

CONCLUSION

Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici Insérez votre texte ici.