Note méthodologique N°1 - Entraînement du modèle

Les contraintes

Le jeu de données analysé après les opérations de preprocessing est de **grande taille**. La dataset utilisé pour entraîner les modèles comporte plus de **300,000 lignes**. Suite à l'encodage One Hot des variables catégorielles, le nombre de **features** est **proche de 800**.
A celà s'ajoute un nombre conséquent de **valeurs manquantes**. Enfin, les **classes** sont particulièrement **déséquilibrées** avec une majorité de crédits appartenant à la classe 0 (crédits accordés). Les crédits refusés sont donc minoritaires, ce qui biaise les modèles qui ont tendance à prévoir la classe 0 dans la majorité des classes.

Nettoyage des données

Etapes

Afin d'obtenir des modèles pertinents, une phase de nettoyage rigoureuse des données en plusieurs étapes est nécessaire. Cette phase est décrite dans la présentation et les notebooks de la phase 1.
Voici un récapitulatif du preprocessing:

- Imputation des valeurs manquantes
- Mise à l'échelle des données
- rééquilibrage des classes

Équilibrage des classes

Compte tenu de la forte disparité entre le nombre d'échantillons de la classe 0 (crédit accepté) et classe 1 (crédit refusé), il est impératif de rééquilibrer les classes afin d'obtenir des modèles de bonne qualité.
On voit bien dans le notebook 7 que, sans cette opération de rééquilibrage, la performance des modèles est très mauvaise puisque l'algorithme va chercher à prévoir la classe majoritaire.
De nombreuses méthodes sont disponibles sous python afin de procéder au rééquilibrage des classes. Nous en avons testé plusieurs en utilisant la librairie Imbalanced Learn.: undersampling, oversampling, SMOTE. Nous avons également testé les paramètres de rééquilibrage spécifiques aux méthodes d'ensemble (XGBoost, Random Forest).

Modélisation

Types de modèles testés

Compte tenu des caractéristiques propres au jeux de données (grande dimension, classification supervisée), nous avons retenu des modèles adaptés à ces contraintes. Voici la liste des modèles testés:

- régression bayésienne
- régression logistique (avec et sans régularisation)
- arbres de décision
- méthodes d'ensemble (XGBoost, RandomForest, LightGBM).

Procédure d'entraînement des modèles
Les données d'origine sont séparées en train et test sets.
A noter que les données du test set ne disposent pas de targets. En d'autres termes, ils ne peuvent pas être utilisés pour le calibrage des modèles.
Compte tenu du déséquilibre entre les classes, il est impératif de créer un train set et un test set comportant une proportion de classes équivalente à celle du dataset d'origine. C'est ce qu'on appelle la stratification.

Pour la phase d'entraînement, nous utilisons uniquement les données du train set, c'est-à-dire celles qui ont été labellisées. Ce jeu de données d'entraînement est lui même divisé en deux sous-échantillons:

- un train set (70% des données) utilisé uniquement pour entraîner les modèles
- un test set (30%), aussi appelé validation set, utilisé pour tester la performance des modèles sur des échantillons nouveaux pour lesquels la target est connue.

Nous sommes donc bien dans un cadre d'apprentissage supervisé.

Sélection des modèles
Afin de sélectionner un modèle final parmi tous ceux testés, il faut convenir de critères d'évaluation en cohérence avec les objectifs. Il est également important de prendre en compte le temps d'entraînement nécessaire dans une optique de passage en production.
Les modèles sélectionnés doivent être performants selon certains critères spécifiques mais également rapides à entraîner. C'est l'objet de la note méthodologique N°2.