
Projet 7 : Implémentez un modèle de scoring

Ilham NOUMIR | Parcours Data Science | Date : /02/2021

— Sommaire

1. Présentation de la problématique et du jeu de données
2. Approche de la modélisation
3. Présentation des résultats
4. Présentation du Dashboard métier

— Problématique : ---

Description de la société:

“Prêt à dépenser” est une société financière qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt

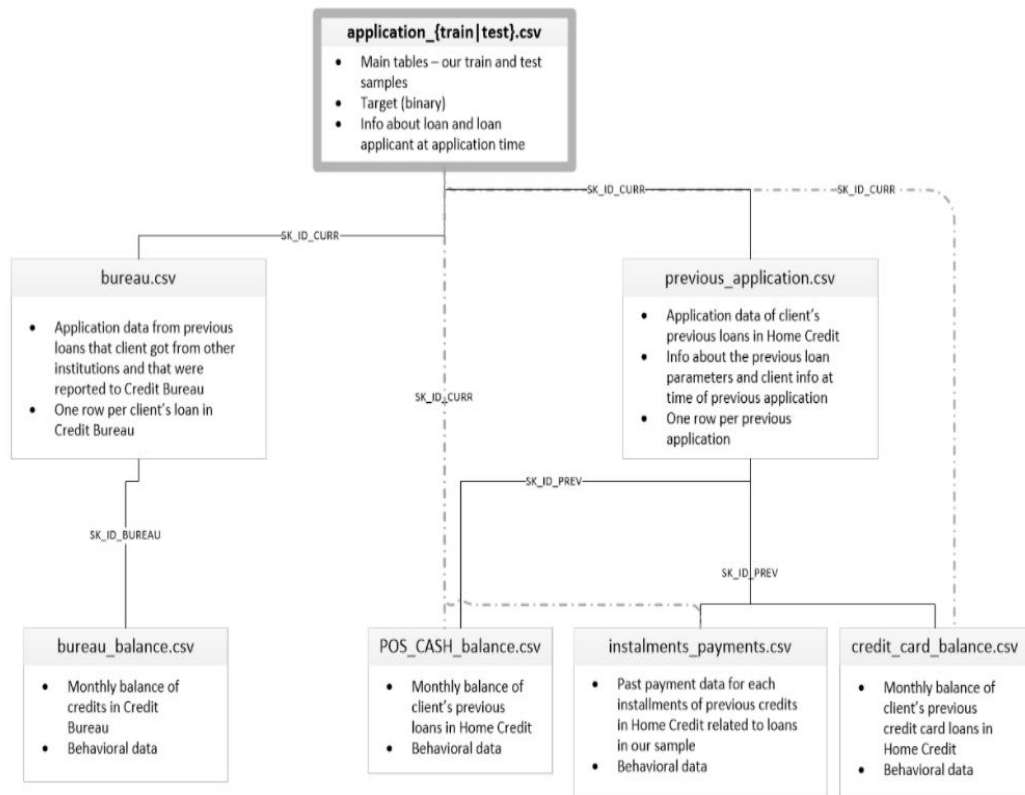
Contexte :

L'entreprise souhaite mettre en œuvre un outil de “scoring crédit” pour calculer la probabilité qu'un client rembourse son crédit

Missions :

1. Construire un modèle de scoring
2. Construire un dashboard interactif permettant d'interpréter les prédictions faites par le modèle

— Jeu de données :



Jeu de données principal :

- 307511 observations
- 121 features
- Variable dépendante :

TARGET = 0 si pas de problème de remboursement

TARGET = 1 si problème de remboursement

— Prétraitement des données : ---

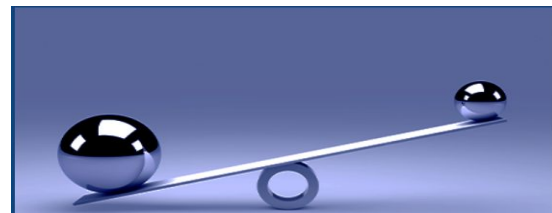
- Suppression de toutes les features ayant plus de 40% de valeurs manquantes
- Détection des outliers / anomalie
- Réduction des modalités des variables catégorielles
- Création de features métier en se basant sur ce [Kernel](#)
- Encodage des variables catégorielles (OneHotEncoder)
- Normalisation des variables numériques (StanderScaler)
- Sélection des variables pertinents grâce au features selection de Scikit learn

— Features métiers :

- ❑ **CREDIT_INCOME_PERCENT** : le pourcentage du montant du crédit par rapport au revenu du client
- ❑ **ANNUITY_INCOME_PERCENT** : le pourcentage de l'annuité du prêt par rapport au revenu du client
- ❑ **CREDIT_TERM** : la durée du paiement en mois (l'annuité étant le montant mensuel dû)
- ❑ **DAYS_EMPLOYED_PERCENT** : le pourcentage des jours d'emploi par rapport à l'âge du client

Variable dépendante :

Problématique : Jeu de données déséquilibré



92 % des clients sans défaut de paiement

8 % des clients avec défaut de paiement

Solution : SMOTE de Imblearn combinée avec la méthode cross validation avec une stratégie de stratification adaptée à ce genre de problématique qui est la méthode de RepeatedStratifiedKFold

— Métriques utilisées : ---


Problématique métier :


Dans le contexte bancaire deux types de risques à prendre en compte :

- Risque financier lié à l'insolvabilité des clients
- Risque de perte d'opportunité (des nouveaux clients)

Objectif : Minimiser les clients à risque qui font perdre de l'argent à la société

Fonction coût métier :

Réduire le risque financier  Réduire les clients à défauts qui sont prédits comme des bons clients

 Réduire le taux des faux négatifs

— Fonction coût métier :

| | Clients prédits en défaut | Clients prédits sans défaut |
|------------------------------|---------------------------|-----------------------------|
| Clients réellement en défaut | Vrais positifs | Faux négatifs |
| Clients sans défaut | Faux positifs | Vrais négatifs |

Fbeta score : La métrique qui permet de définir le poids qu'on souhaite attribuer au recall ou à la précision.

Précision = $TP / (TP+FP)$

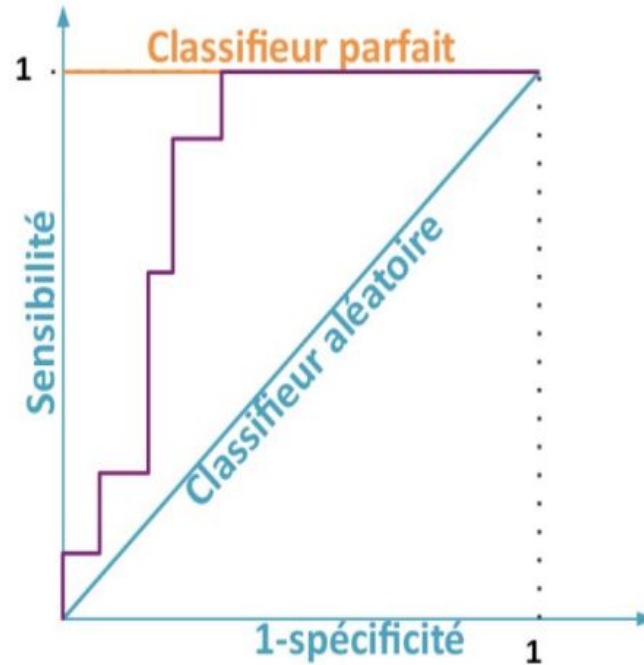
Recall = $TP/(TP+FN)$

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

— Métrique d'évaluation:

La courbe ROC représente le taux de vrais positifs (TPR) par rapport au taux de faux positifs (FPR).

La métrique ROC_AUC correspond à l'aire sous la courbe ROC, elle est comprise entre 0 et 1.



— Approche de la modélisation :

Algorithmes testés :

- LogisticRegression
- RandomForestClassifier
- GaussianNB
- DecisionTreeClassifier
- XGBClassifier
- GradientBoostingClassifier

— Modèle choisi:

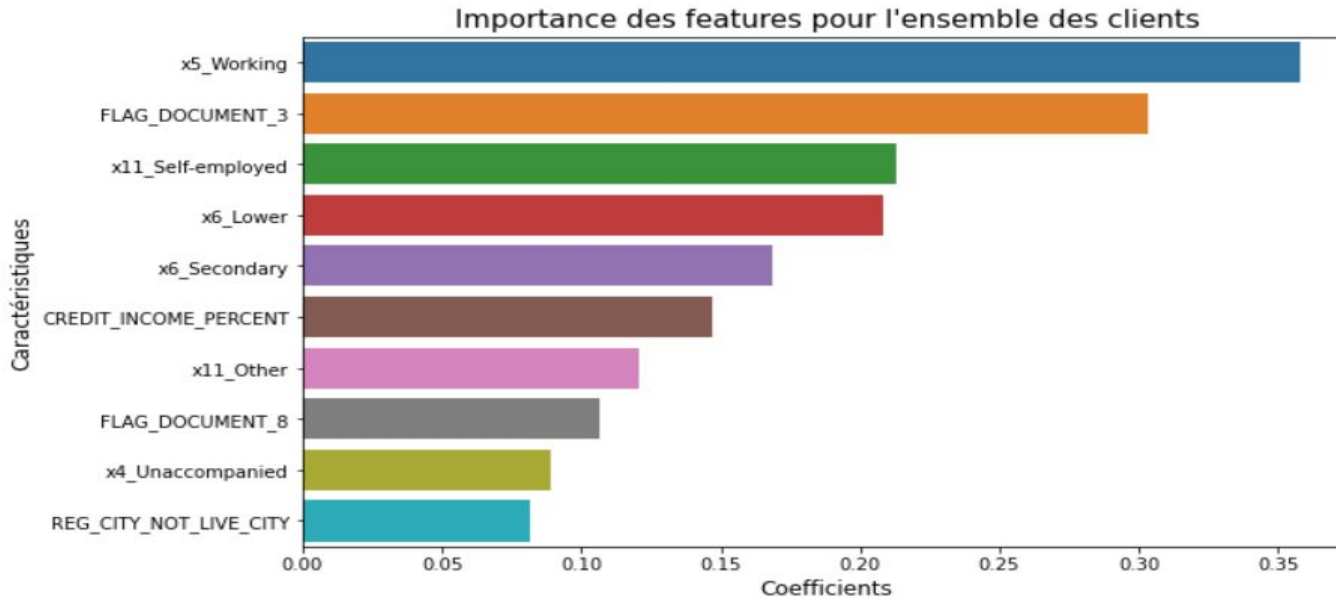
| | name | fbeta score training | AUC training | fbeta score test | AUC test |
|---|----------------------------|----------------------|--------------|------------------|----------|
| 0 | LogisticRegression | 0.450023 | 0.779952 | 0.428043 | 0.753343 |
| 1 | RandomForestClassifier | 1.000000 | 1.000000 | 0.149021 | 0.721708 |
| 2 | GaussianNB | 0.319311 | 0.531986 | 0.318506 | 0.526606 |
| 3 | DecisionTreeClassifier | 1.000000 | 1.000000 | 0.226633 | 0.568475 |
| 4 | XGBClassifier | 0.974246 | 0.999893 | 0.156041 | 0.709660 |
| 5 | GradientBoostingClassifier | 0.385635 | 0.831904 | 0.258458 | 0.726335 |

Choix des hyperparamètres: HyperOpt

```
Best: {'C': 0.053974302792412, 'fit_intercept': 1, 'max_iter': 597, 'solver': 1, 'tol': 9.755892114577838e-05, 'warm_start': 0}
```

— Interprétabilité des résultats :

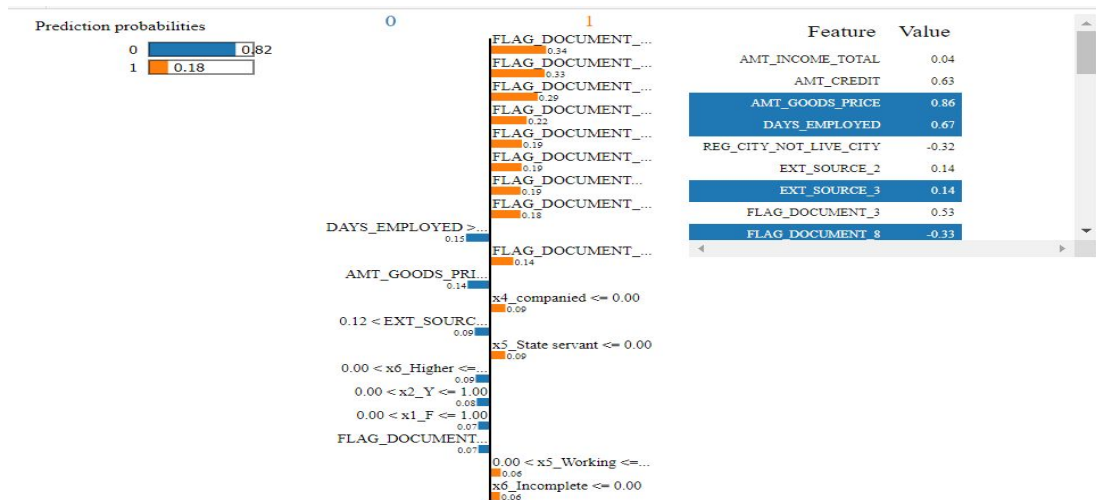
Interprétabilité globale:



Interprétabilité des résultats :

Interprétabilité locale :

LIME



SHAP



— Présentation API et dashboard :

[API](#)

[Dashboard](#)

— Conclusion et améliorations possibles : ---

- La fonction coût basée sur des hypothèses métiers confirmées
- Feature engineering plus élaboré
- Dashboard : Plus de diversification sur les graphes