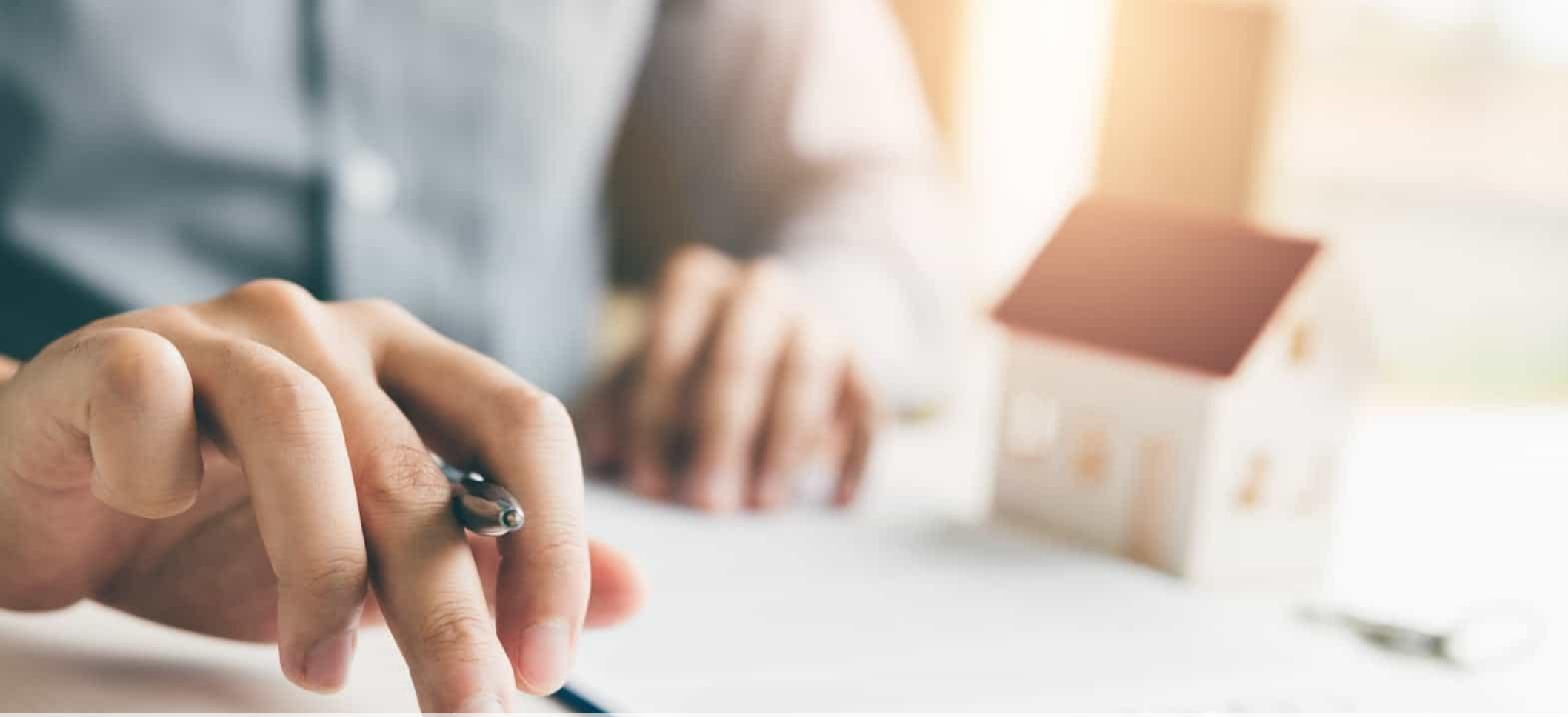


Projet 7 – Parcours Data Scientist

Implémentez un modèle de scoring

OPENCLASSROOMS

Robin Perbet – octobre 2022

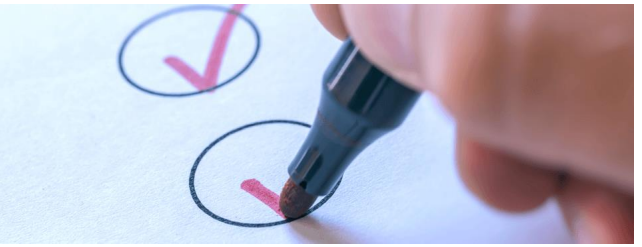
Sommaire

1. Introduction

Introduction

Problématique du projet

Présentation de la problématique

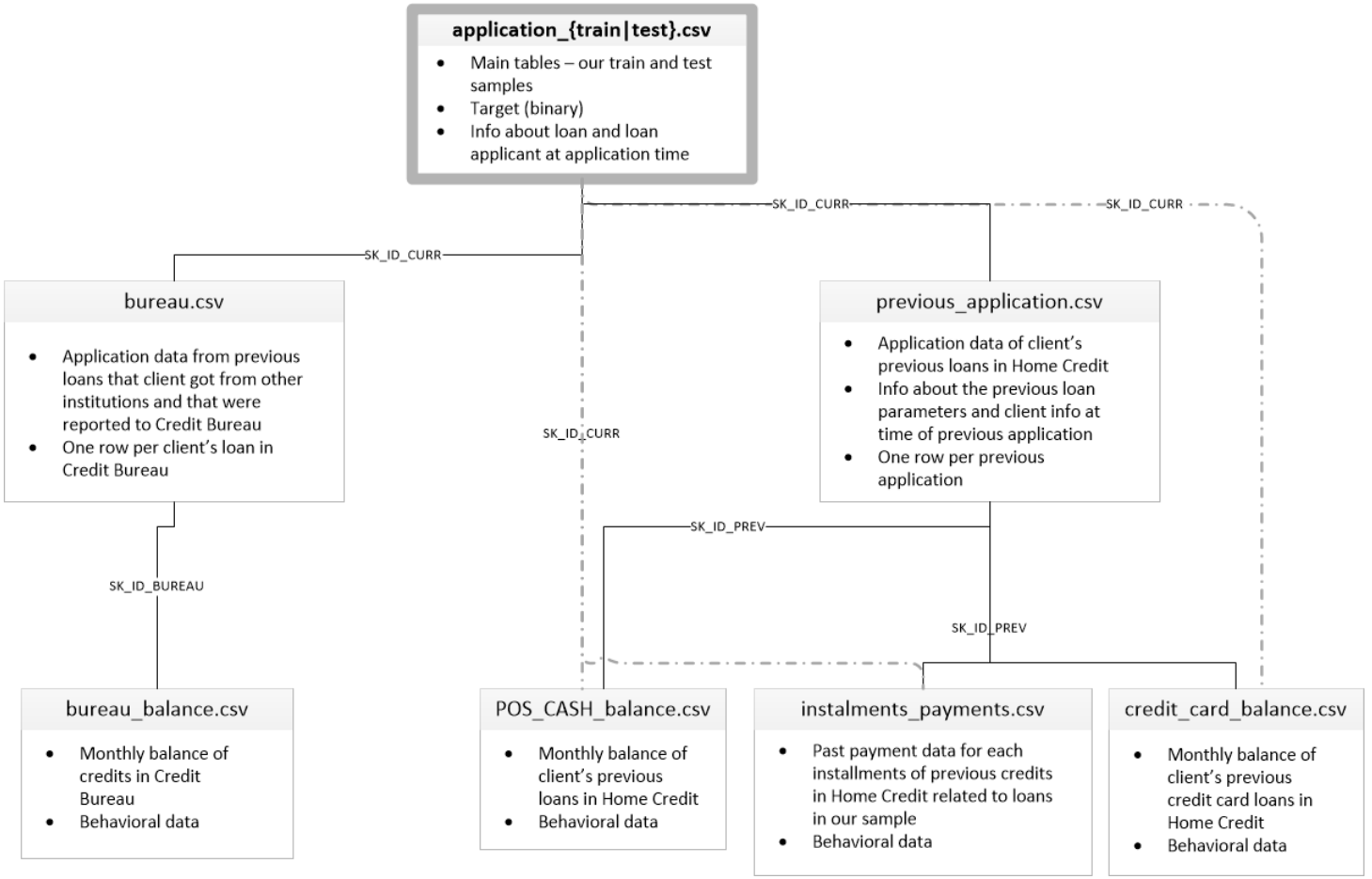
Une entreprise de crédits à la consommation	L’entreprise "Prêt à dépenser” propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt	
La mise en place d’un scoring crédit…	Afin de déterminer l’octroi ou non d’un crédit à un potentiel client, l’entreprise souhaite **mettre en œuvre un outil de “scoring crédit”** afin calculer la probabilité de capacité de remboursement du crédit à travers un algorithme de classification	
…et d’un dashboard afin d’accéder aux résultats	Afin que les chargés de clientèle puissent accéder facilement aux informations du modèle et de l’interprétation des résultats, l’entreprise souhaite également développer un dashboard interactif	

2. Nettoyage, pre-processing et feature engineering

Nettoyage, pre-processing et feature engineering

Jointure des tableaux dans un jeu de données unique

Présentation du schéma du jeu de données



Objectifs

1. Conserver une ligne par client (clé primaire SK_ID_CURR) avec toutes les informations associées
2. Repérer les éventuelles incohérences afin que la jointure soit effectuée proprement
3. Synthétiser certaines variables afin d'avoir une seule ligne par client via des regroupements sur la clé primaire :
   - Variables numériques : sommes, moyennes, médianes, variances, maximums et minimums
   - Variables catégorielles : mode

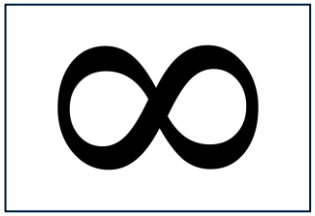
Nettoyage, pre-processing et feature engineering

Nettoyage et preprocessing

Méthodologie

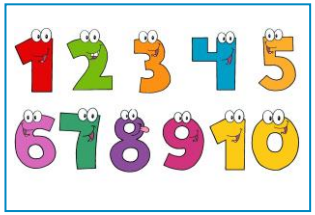
1 Suppression des valeurs infinies

- Les opérations de regroupement et certains calculs ont créé des valeurs infinies dans notre jeu de données
- Ces valeurs sont remplacées par NaN



2 Traitement des valeurs numériques

- Imputation par la moyenne des valeurs manquantes
- Normalisation (centrage – réduction)

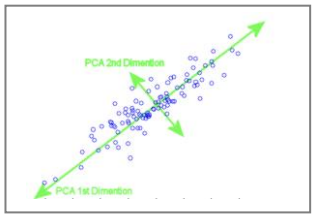


3 Traitement des valeurs catégorielles

- Imputation par le mode des valeurs manquantes
- One-hot-encoding

4 Réduction de dimension

- Afin d'accélérer l'entraînement de nos modèles, nous avons également procédé à une ACP en conservant 99% de la variance afin de réduire le nombre de variables tout en conservant le maximum d'information
- Pour l'optimisation finale, cette étape sera supprimée afin de conserver les variables originales



Nettoyage, pre-processing et feature engineering

Feature engineering

Présentation du feature engineering et de la méthodologie de test

Objectif

- Créer des variables synthétiques par client unique afin de pouvoir effectuer une classification métier pertinente

Test des résultats

- Nous avons créé une fonction permettant d'entraîner différents algorithmes de classification et d'observer la qualité des prédictions sur un jeu de validation (20% du jeu de données) en calculant plusieurs scores (aire sous la courbe, F1-score, accuracy, précision, recall, cross entropy et temps de calcul)
- Cette approche simple, sans optimisation des paramètres ni validation croisée à ce stade, nous permet d'obtenir rapidement des résultats sur le jeu de données et d'observer l'impact positif ou négatif du feature engineering sur les scores moyens des modèles
- Le calcul des variables ci-dessous a amélioré les performances moyennes des algorithmes, elles ont donc été conservées

Variables créées

- Taux d'effort : ratio annuité / revenu d'un client
- Loan-to-value (LTV) : ratio montant du crédit / prix du bien financé
- Ratio montant de l'emprunt demandé / montant reçu
- Ratio montant payé / montant dû lors des précédentes échéances de crédit
- Différence montant payé avec le montant dû lors des précédentes échéances de crédit
- Ratio nombre de jours de retard (ou d'avance) du paiement des précédentes échéances
- Ratio revenu annuel / montant du crédit perçu
- Ratio annuité / montant du crédit
- Ratio nombre de jours en situation d'emploi / nombre de jours vécus par un individu
- Ratio revenu total / nombre de personnes dans le ménage

Nettoyage, pre-processing et feature engineering

Equilibrage des classes

Présentation de la méthodologie

- Nous faisons face à une problématique de déséquilibre des classes avec près de 92% des données de la classe négative
- Il s'agit d'une problématique fréquente dans les projets de détection de cas positifs, mais ce déséquilibre peut amener l'algorithme à être moins performant dans la détection des cas positifs
- Les méthodes décrites ci-dessous ont été utilisées afin de pallier ce problème

Over-sampling	Under-sampling
■ L'over-sampling consiste à créer des points artificiels de la classe sous-représentée à partir des points existants ■ Nous avons utilisé la méthode SMOTE qui réalise pour cela un KNN et créé un point à une distance aléatoire d'un point sélectionné et de ses plus proches voisins ■ Nous avons appliqué un over-sampling de 10%	■ L'under-sampling consiste à supprimer aléatoirement des points de la classe sur-représentée ■ Nous avons appliqué un over-sampling de 50%

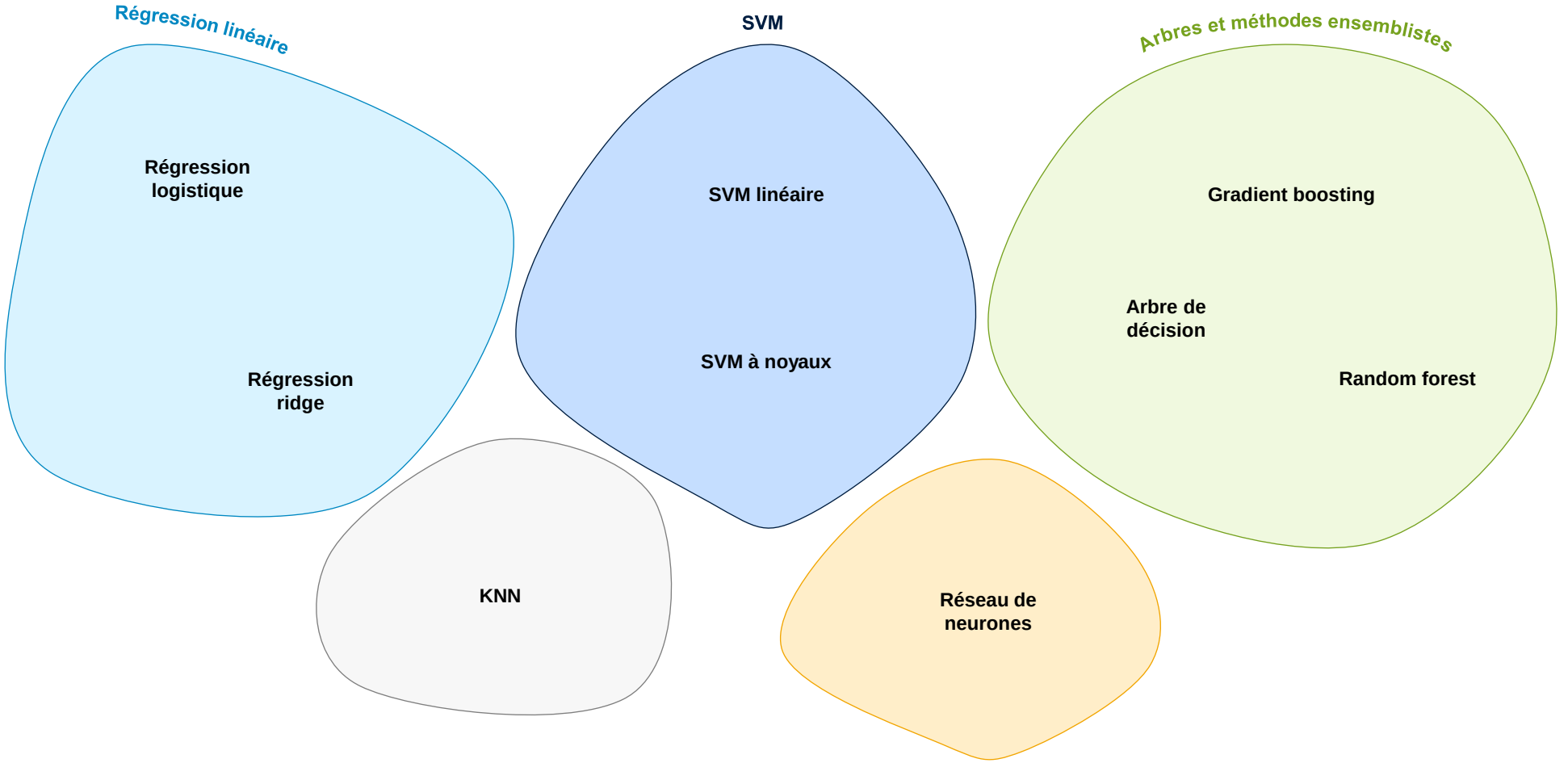
L'application de ces méthodes d'équilibrage ont amélioré les performances moyennes des modèles, elles ont donc été conservées

3. Sélection du modèle

Sélection du modèle

Modèles utilisés

Présentation des modèles testés



Sélection du modèle

Comparaison des performances

Performances des modèles utilisés après optimisation en validation croisée

	Regression logistique	Regression ridge	KNN	SVM linéaire	SVM à noyaux	Decision tree	Random Forest	Gradient boosting	Neural network
Score roc_auc cross-validation	0,7680	0,7678	0,6460	0,7681	0,6852	0,5752	0,7289	0,7601	0,7638
Paramètres optimisés	■ C	■ alpha	■ k	■ C	■ C	■ min_samples_split ■ min_samples_leaf	■ estimators	■ estimators	■ learn_rate ■ n_hidden ■ n_neurons
Meilleurs paramètres	■ 1	■ 10	■ 10	■ 0,1	■ 0,4	■ 5 ■ 2	■ 500	■ 500	■ 0,00097 ■ 0 ■ 13
Temps d'entraînement cross-validation	21s	31s	215s	2628s	3122s	177s	1642s	43891s	2813s

- Plusieurs algorithmes affichent des scores très proches : régression logistique, régression ridge, SVM linéaire, gradient boosting et neural network
- Nous allons donc déterminer notre choix sur le temps d'entraînement et la facilité d'interprétation du modèle
- La régression logistique semble donc être le meilleur candidat, nous le retenons pour les étapes suivante

4. Optimisation métier

Optimisation métier

Méthodologie d'optimisation

Présentation des contraintes métiers et des solutions utilisées

- Nous avons utilisé jusqu'ici le score roc_auc, qui est pertinent dans le cadre de classes déséquilibrées. Néanmoins, nous devons également tenir compte des contraintes de notre client d'un point de vue métier
- Les modèles de classification génèrent deux types d'erreurs, et d'un point de vue métier ces dernières n'ont pas le même impact sur la banque

Erreurs de type I (faux positif)

- Prédire des difficultés de paiement d'un client et ne pas lui accorder de prêt alors que le client était en réalité solvable présente un coup d'opportunité pour la banque qui ne prend pas un bon client et ne réalise pas sa marge de crédit

Erreurs de type II (faux négatif)

- Prédire que le client est solvable et lui accorder le prêt alors qu'il ne l'était pas en réalité présente une perte nette du montant non remboursé pour la banque

- La magnitude de la perte dans les deux situations n'est pas la même, la marge de crédit perdue présente un pourcentage limité du montant du prêt tandis que le non-remboursement d'un emprunt peu représenter une perte nette pouvant aller jusqu'au montant total du crédit accordé
- Nous allons donc tenter d'optimiser notre modèle afin de limiter le nombre de faux négatif, ce qui va en revanche augmenter le nombre de faux positif, il s'agit d'un arbitrage entre 2 scores que sont la précision et le recall

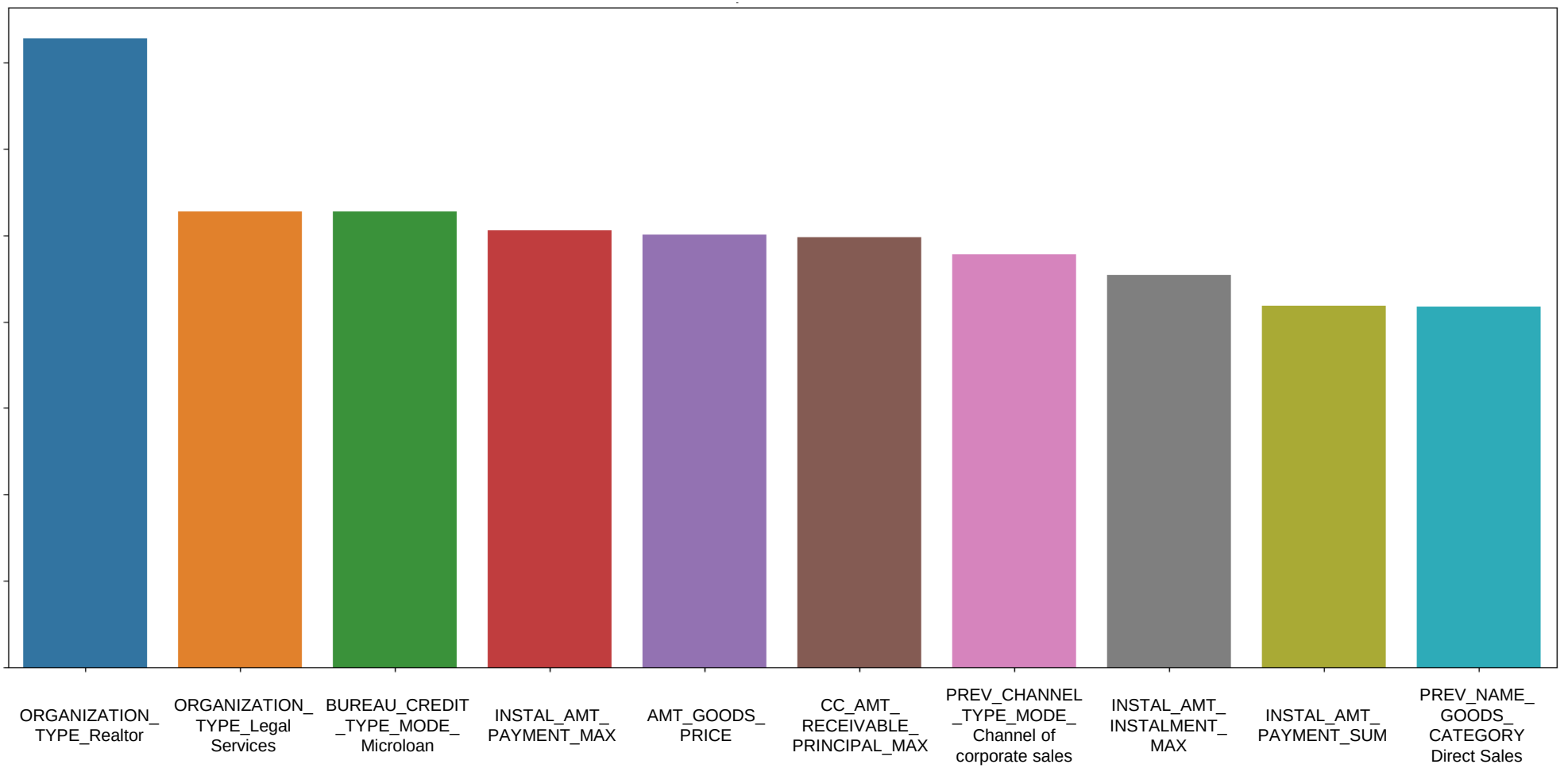
- Afin d'atteindre cet objectif, nous allons optimiser l'algorithme en fonction du F-beta-score qui permet d'allouer un poids plus important au recall
- Par ailleurs, nous allons également optimiser le seuil de prédiction qui impacte directement l'arbitrage precision/recall

5. Interprétation

Interprétation

Feature importance globale - coefficients

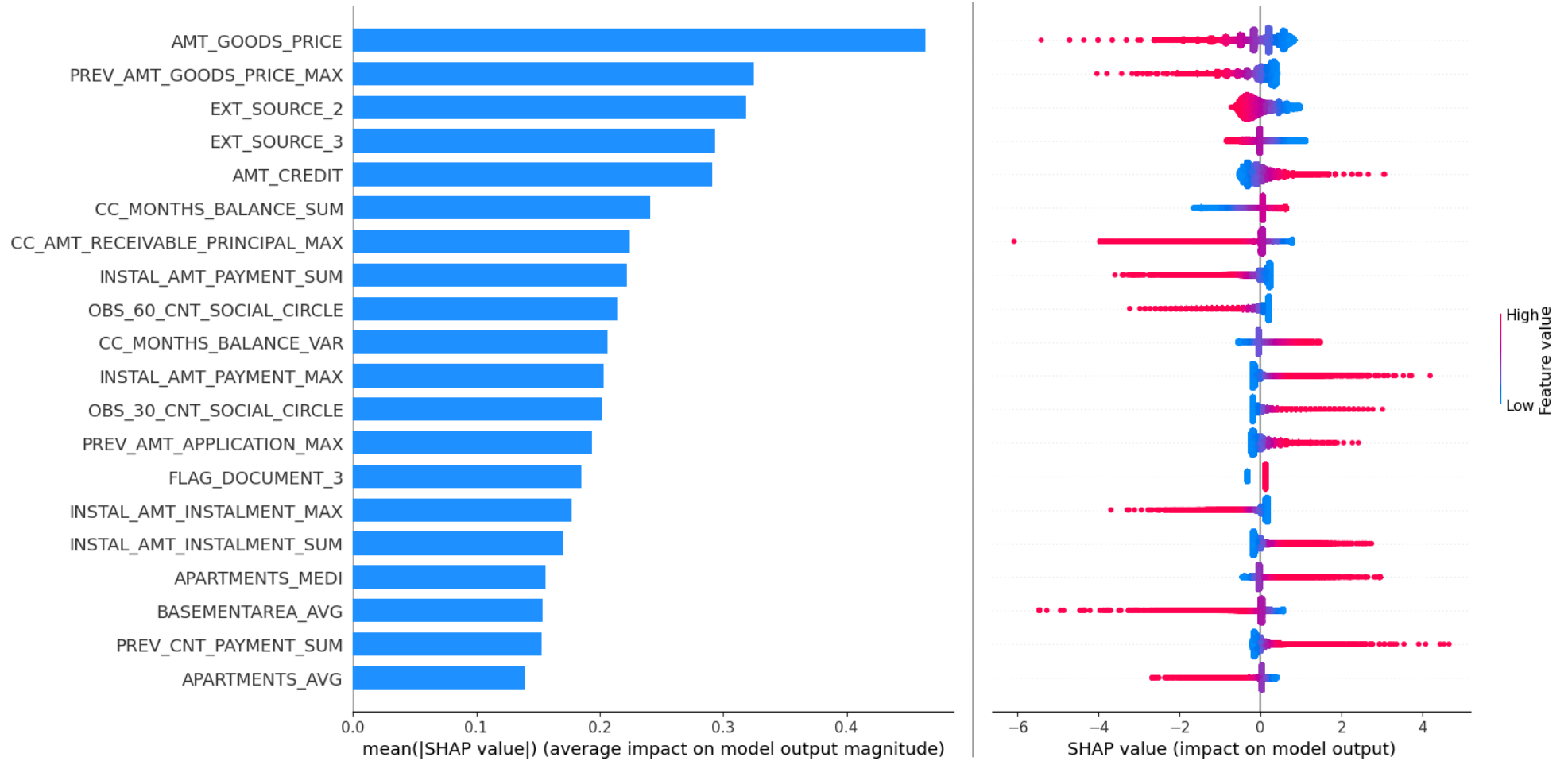
Top 10 features d'après les coefficients de la régression logistique (valeurs absolues)



Interprétation

Feature importance globale – SHAP values

Top 20 features d'après les valeurs SHAP (moyennes des valeurs absolues pour chacune des variables)



6. Mise en place du dashboard

Mise en place du dashboard

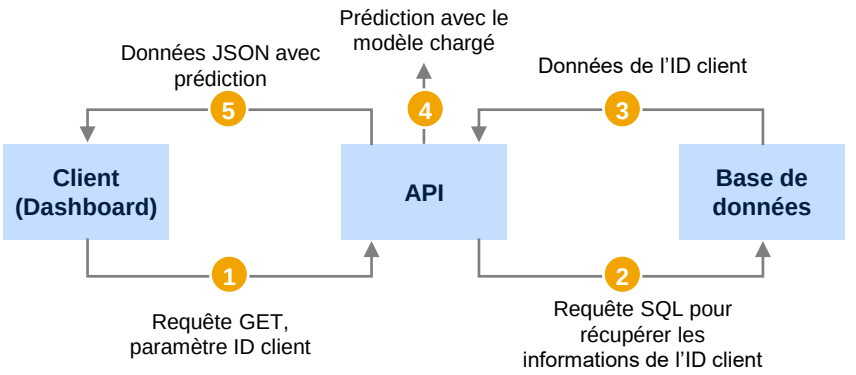
Mise en place d'une API avec Flask

Présentation de l'API mise en place

- Une API est une interface permettant de retourner un résultat à une requête HTTP sous forme de données JSON
- Dans notre cas, l'objectif était de mettre en place une API retournant les prédictions réalisées par notre modèle
- Nous avons utilisé 2 URLs différentes en fonctions des scénarios présentés ci-dessous

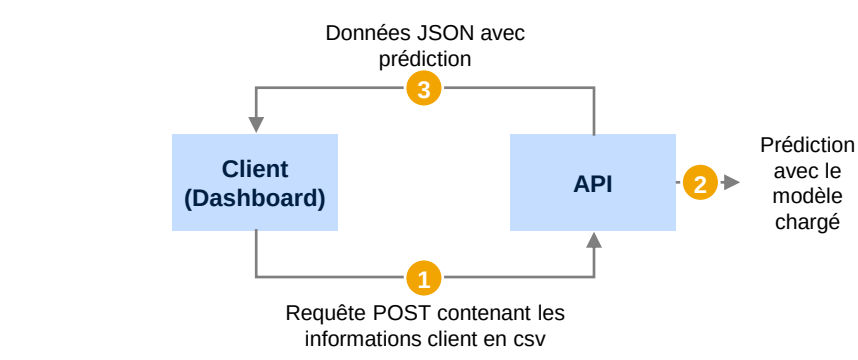
Prédiction pour un client présent dans notre jeu d'entraînement

- Afin de déterminer les prédictions, nous disposons des données
- La requête sera de forme GET et comportera en paramètre l'identifiant unique du client
- L'API va à son tour envoyer une requête à une base de données que nous avons créé et contenant le jeu d'entraînement (après opérations de preprocessing) afin d'obtenir les informations sur le client concerné
- L'API va utiliser ces informations afin de réaliser une prédiction grâce au modèle final chargé au format pickle
- Cette prédiction sera renvoyée au format JSON



Prédiction pour un nouveau client

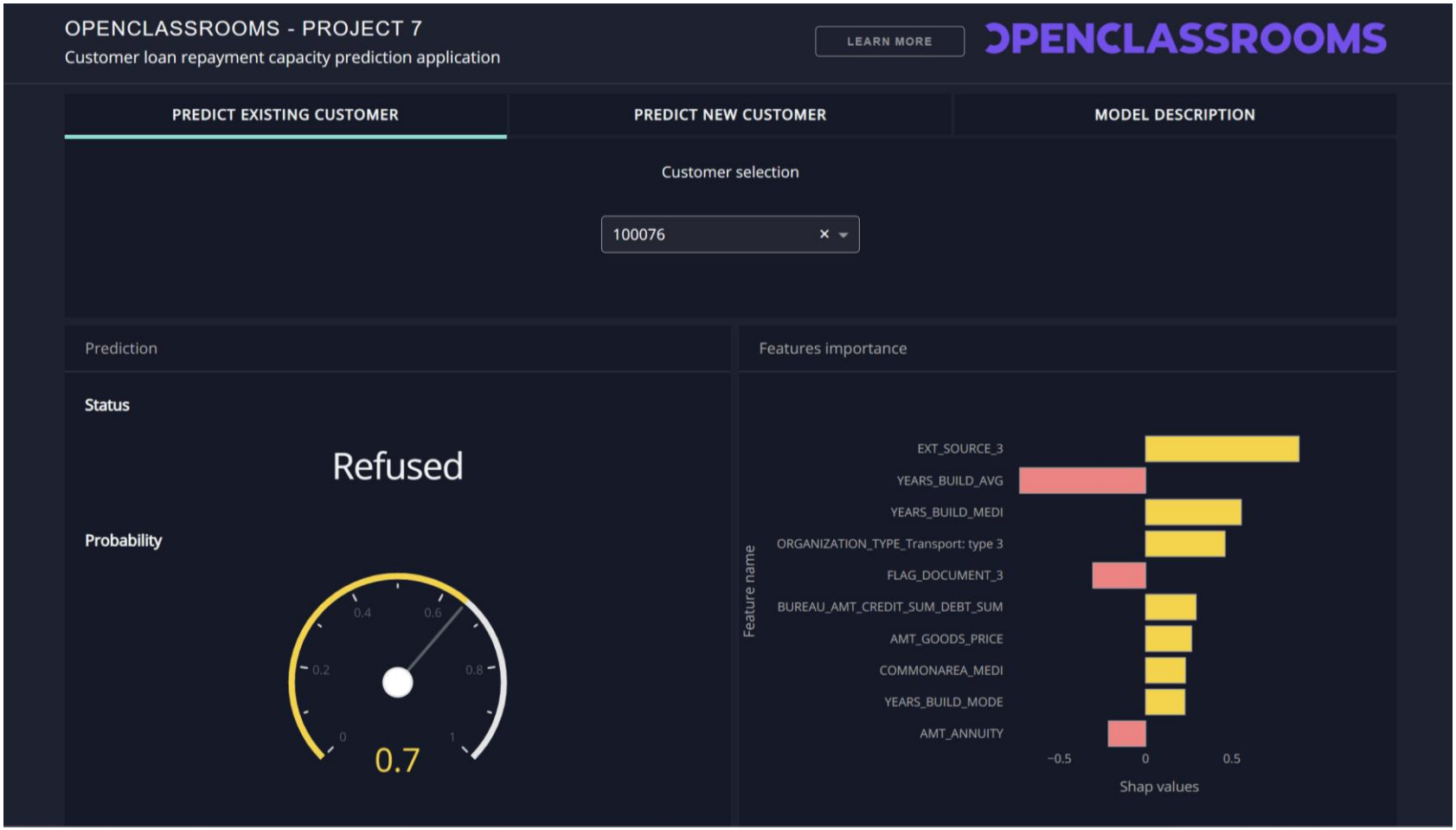
- Ici, les informations ne sont pas présentes dans la base de données
- La requête sera de forme POST et enverra les informations de ce nouveau client (après opérations de preprocessing identiques à celle réalisées sur le jeu d'entraînement)
- L'API va simplement faire la prédiction grâce au modèle final chargé au format pickle et renvoyer les données JSON



Mise en place du dashboard

Dashboard avec Dash (1/3)

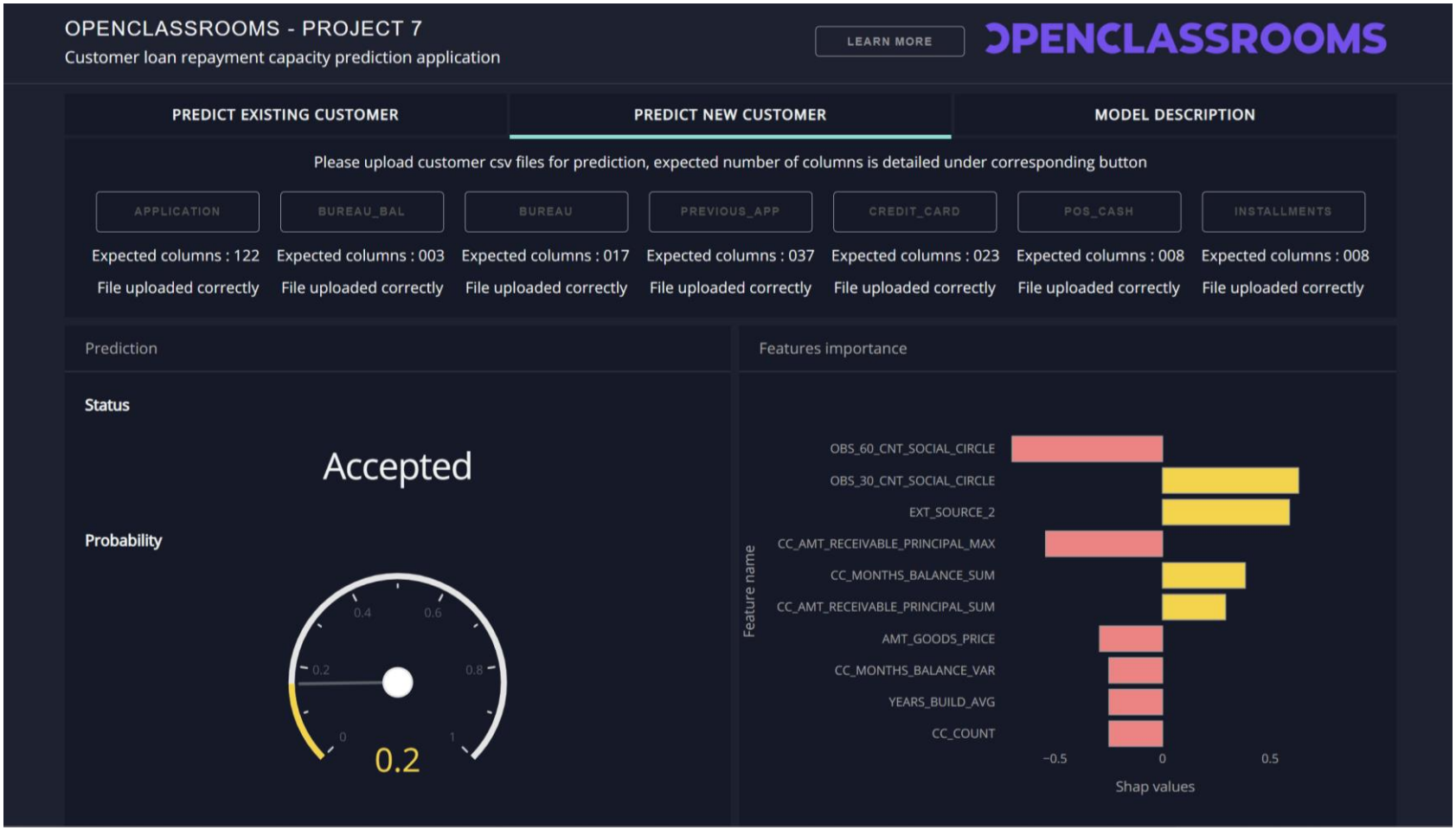
Page « predict existing »



Mise en place du dashboard

Dashboard avec Dash (2/3)

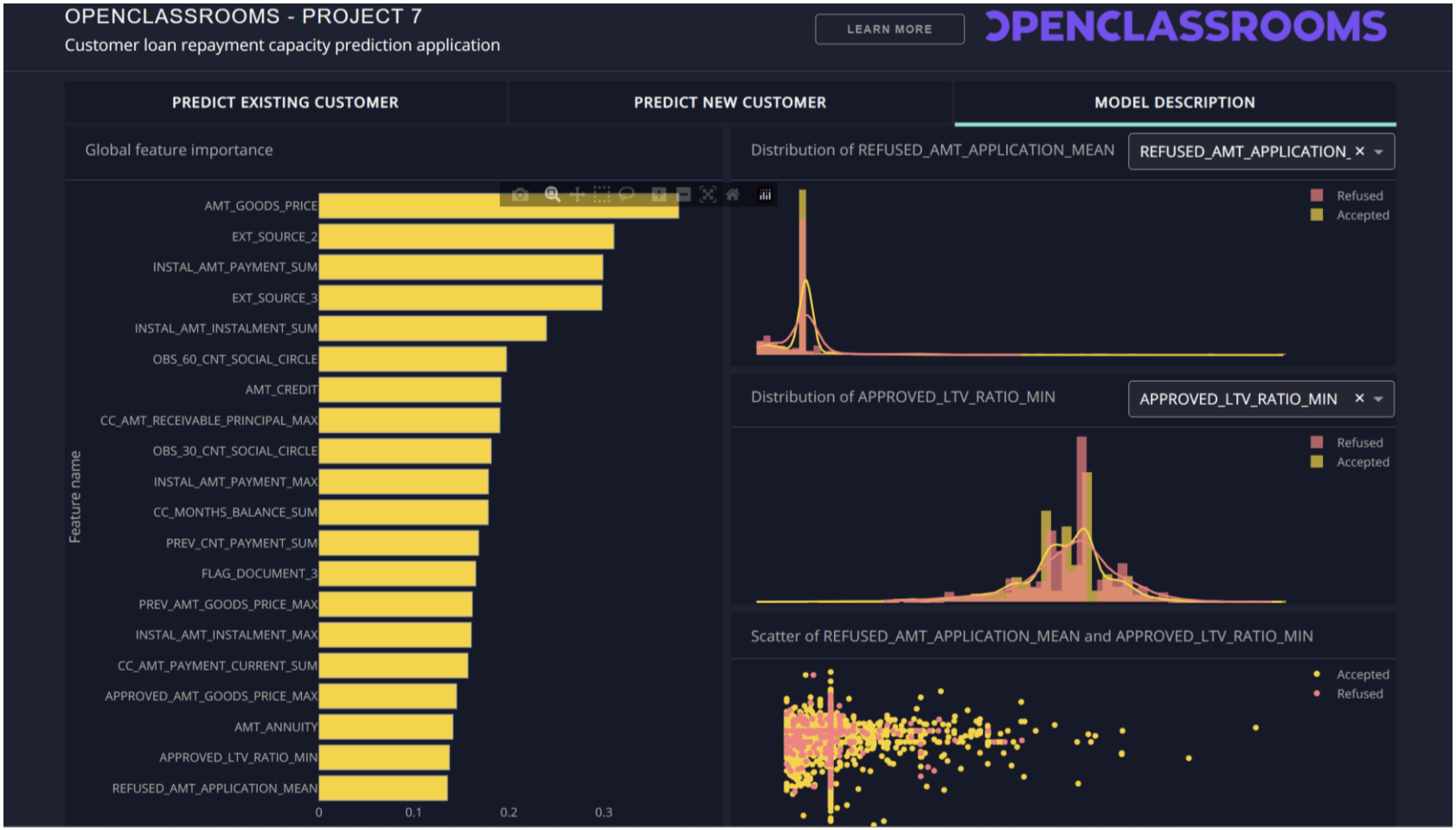
Page « predict new »



Mise en place du dashboard

Dashboard avec Dash (3/3)

Page « model description »



7. Conclusion

Conclusion

Présentation de la méthodologie d'évaluation de la stabilité du modèle

Une mise en œuvre possible et intéressante

- ✓ Notre algorithme affiche des performances satisfaisantes et peut être encore adapté en fonction des préférences du client
- ✓ L'algorithme est rapide lors de l'entraînement et de la prédiction, facilitant la mise en production
- ✓ L'algorithme est simple et permet une interprétation plus aisée
- ✓ La mise en place d'une application dashboard permettra aux chargés de clientèle de s'appuyer sur cet outil pour rapidement et visuellement accéder aux résultats du modèle. Il sera ainsi possible d'argumenter en direct avec un client potentiel sur sa candidature
- ✓ Cela répond bien aux attentes de notre client

Des points d'attention

- × Le modèle retenu présente un nombre important de variables. Cela nous permet d'afficher de meilleures performances, mais complexifie l'interprétation
- × Certaines variables sont difficilement interprétables de manière intuitive, ou tout du moins difficile à justifier auprès d'un client se voyant refuser son crédit
- × Parmi les variables les plus importantes détectées lors de la prédiction, nous retrouvons les 3 variables EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3. Malheureusement, nous ne savons pas précisément à quoi correspondent ces variables
- × Nous avons dû synthétiser certaines variables lors de l'étape de preprocessing, ce qui entraîne une perte d'information. A l'inverse, ces regroupements peuvent provoquer des informations redondantes
- × A noter également que l'optimisation du modèle n'était pas le cœur de ce projet. Une optimisation plus poussée aurait pu être réalisée en consacrant plus de temps à cette tâche, mais les efforts ont été plus particulièrement concentrés sur les étapes de déploiement

Liens vers le code, le dashboard et l'API

- L'API est disponible aux liens suivants :
  - Prédiction client existant (avec requête GET comportant id client) : https://p7-openclassrooms-api.herokuapp.com/api/predict-existing/
  - Prédiction nouveau client (avec requête POST comportant fichier client) : https://p7-openclassrooms-api.herokuapp.com/api/predict-new/
- Le dashboard est disponible au lien suivant : https://p7-openclassrooms-dash.herokuapp.com/
- Le code est disponible dans les 3 répertoires Github suivants :
  - Modélisation : https://github.com/RobinPbt/P7-Openclassrooms
  - API : https://github.com/RobinPbt/P7-Openclassrooms-API
  - Dashboard : https://github.com/RobinPbt/P7-Openclassrooms-Dashboard