horizontal line

Note Méthodologique

**18 Décembre 2021**

# Objectif

L’algorithme doit produire un score qui permet d’évaluer rapidement la capacité d’un candidat à rembourser un crédit. Ce score dépend d’une liste de variables décrivant la situation et le passé financier d’un candidat.

# Méthodologie d’entraînement

## Données déséquilibrées

Choix des données

Pour simplifier l’exécution du projet, un kernel est utilisé. Celui-ci permet de passer outre l’étape de data cleaning et de la majorité du feature engineering et de se concentrer la modélisation et le déploiement.

Le kernel sélectionné vient du site Kaggle ([lien](https://www.kaggle.com/aghandemir/home-credit-with-functions/script?scriptVersionId=32496037)) et est constitué de x variable et y observations.

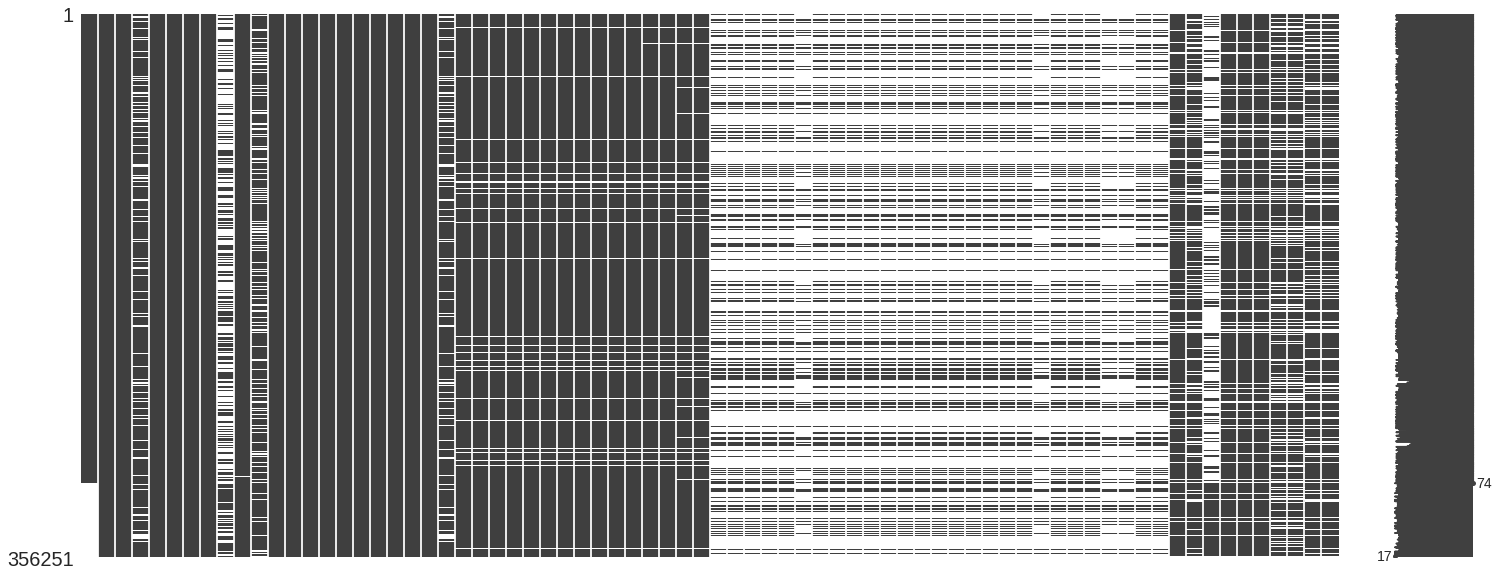
Préparation des données

Les données des dossiers disponibles sont largement déséquilibrées avec la population de dossiers qui ont fait défaut sur leur crédit représentant moins de 10% de la base totale.

Il n’y a pas de consensus autour des meilleurs approches à adopter pour les données déséquilibrées mais nombre de sources indiquent les variantes de SMOTE (Synthetic Minority Over-sampling Technique) sont parmi les plus performantes en terme de résultat de prédiction bien qu’il n’y ait pas une solution unique.

En particulier, polynom-fit-SMOTE, décrite dans [Lien](https://www.researchgate.net/publication/220933172_New_Oversampling_Approaches_Based_on_Polynomial_Fitting_for_Imbalanced_Data_Sets), paraît être une variante particulièrement performante sur une variété de données et de métriques, [Lien](https://www.researchgate.net/publication/334732374_An_empirical_comparison_and_evaluation_of_minority_oversampling_techniques_on_a_large_number_of_imbalanced_datasets). Malheureusement, cette méthode n’est pas compatible avec le pipeline de la bibliothèque imblearn donc nous nous contentons de la variante de base.

Dans le kernel utilisé, il existe des valeurs manquantes :



Notamment, on observe les données test qui ont été ajoutées à la fin de la base de données dans la première colonne.

Nous imputons donc les données avec SimpleImputer qui utilise, naîvement, la moyenne ou la médiane par défaut pour remplir les valeurs manquantes.

Ces valeurs sont ensuite normalisées avec StandardScaler.

Finalement, pour les données d’entraînement, on sur-échantillonne avec la fonction SMOTE, de la bibliothèque imblearn, les valeurs “1” dans la variable cible “TARGET” qui indiquent des observations qui ont fait défaut.

## Entraînement du score

Choix desalgorithmes

S’agissant d’une modélisation d’une issue à deux choix, un premier algorithme intéressant à explorer est la régression logistique. Modèle statistique de prédilection dans ce type de situation, il s’agit d’une approche qui reflète les méthodes classiques.

Un deuxième type de modèle intéressant à évaluer est le random forest classifier qui, avec suffisamment de forêts et de profondeur, peut atteindre des résultats satisfaisants sur les données d’entraînement. Modèle simple à mettre en oeuvre, il offre la possibilité d’optimiser des hyperparamètres afin d’explorer le potentiel d’amélioration au-delà de la préparation des données.

Finalement, relativement similaire au random forest, on explore un framework de gradient boosting, Light Gradient Boosting Machine (LGBM). Cette approche est connue pour donner des performances améliorées pour des données déséquilibrées..

Entraînement de l’algorithme

Les données issues du kernel sont séparées immédiatement en deux parties : données d’entraînement (80%) et données test (20%).

Uniquement les données d’entraînements sont modifiées avec SMOTE afin d’entraîner le modèle. Les données test n’ayant pas un impact sur l’entraînement, celles-ci sont gardées telles quelles.

Les différents modèles sont ensuite entraînés, avec optimisation des hyperparamètres pour le random forest classifier et le Light Gradient Boosting Machine.

## Classification du dossier

### Fonction du coût métier Dans un contexte réel, un dossier de crédit n’est pas évalué que sur le risque spécifique représenté par celui-ci : l’importance de l’exposition de l’institution impose des limites plus ou moins strictes en fonction du capital total à risque et des contraintes imposées par la législation.

Pour se rapprocher de ce fonctionnement, nous pourrions mesurer la quantité de capital qui est à risque à tout moment et refuser tout dossier qui augmente de manière significative ce risque.

En l’absence de directive à cet égard, nous pouvons nous contenter de classer les dossiers en deux groupes :

* Les dossiers refusés : dossier ayant une probabilité de défaut inférieur à x (déterminé en fonction de la performance des algorithmes)
* Les dossiers acceptés : dossier ayant une probabilité de défaut supérieur à x (déterminé en fonction de la performance des algorithmes).

### Algorithme d’optimisation

Comme mentionné précédemment, nous essayons trois algorithmes différents pour trouver la meilleure solution. D’après les résultats suite à une optimisation des hyperparamètres, le random forest classifier obtient la meilleure performance.

Nous procédons donc avec cet algorithme dans le déploiement du modèle.

### Métrique d’évaluation

Pour ce cas d’issues binaires, on peut se limiter aux métriques suivantes :

* La précision
* Le rappel
* La courbe AUC
* F1 Score

Dans ce cas de figure, le rappel est la métrique la plus pertinente. Dans un contexte bancaire où l’exposition au risque est extrêmement contrôlée et les pertes pénalisées, la capacité à prédire le mieux possible l’ensemble des mauvais payeurs ou risques de défaut est primordiale.

Le rappel, défini par :

Rappel = avec tp = vrai positif et fn = faux négatif

Exprime la proportion de l’ensemble des personnes à risques de défaut identifiée par l’algorithme.

### Sélection de l’algorithme

Après optimisation des hyperparamètres, nous sélectionnons le LGBM qui obtient un recall de 71,8% sur des données tests avec un seuil de 0,5 pour la classification par défaut, la meilleure performance.

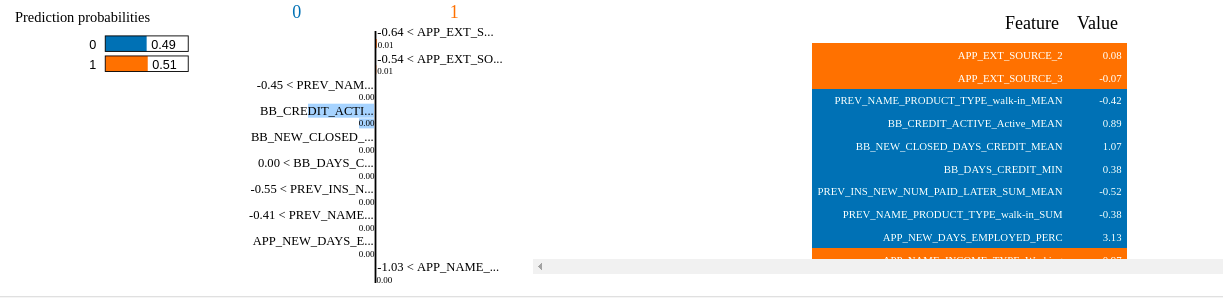
# Interprétabilité

La complexité des modèles qui sont à présent disponibles fait qu’il est possible de s’approcher de résultats très performants sans pour autant faire une sélection rigoureuse des variables qui sont incluses. Il devient donc important d’évaluer après entraînement l’importance de ces variables pour déterminer l’interprétabilité des modèles afin d’en comprendre les potentielles ramifications au niveau métier.

## Locale

La méthode LIME (Local Interpretable Model-Agnostic Explanations) génère des explications de la contribution des variables à tout type de modèles pour une observation donnée.

Pour une observation test négative :

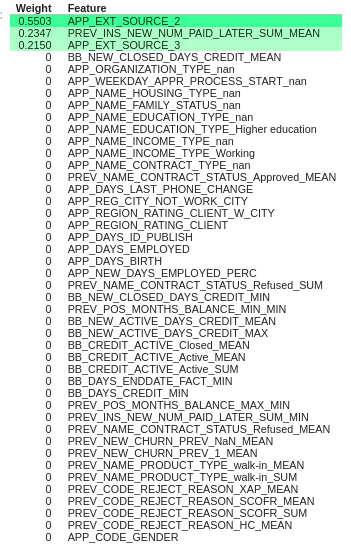


Le modèle est confiant à 51% que c’est un dossier qui fera défaut avec les features en orange qui augmente la probabilité de faire défaut tandis que les features en bleu augmente la probabilité de ne pas faire défaut.

## Globale

ELI5

La bibliothèque ELI5 permet d’obtenir une vision de l’interprétabilité globale d’un modèle, notamment en indiquant le poids d’une variable dans un modèle. Dans le modèle conservé, on constate que seules trois variables sont estimées impacter significativement la prédiction : APP\_EXT\_SOURCE\_2, PREV\_INS\_[...], et APP\_EXT\_SOURCE\_3.



# Synthèse

## Limites

### Contraintes métiers

Un outil de prédiction déconnecté de son usage métier ne pourra que satisfaire partiellement les utilisateurs. Comme mentionné précédemment, le crédit bancaire est sujet à une réglementation stricte qui impacte la capacité des agences à accepter un dossier d’emprunt. Ici, aucune contrainte n’est offerte par rapport à ce sujet pour guider les seuils à utiliser pour séparer les dossiers d’emprunts.

## Améliorations

### Gestion de données déséquilibrées

La méthode utilisée pour ré-équilibrer les catégories à travers à sur échantillonnage, SMOTE, crée de nouvelles observations afin de sur-représenter la catégorie minoritaire mais est une méthode parmi de nombreuses différentes. Plusieurs méthodes de sur-échantillonnage auraient pu être comparées afin de sélectionner la plus performante.

Une alternative aurait été de sous-échantillonner la catégorie majoritaire afin d’éviter de créer de nouvelles données artificielles qui peuvent introduire un biais.

### Algorithme d’optimisation

Les algorithmes comparés sont restés “classiques”. Le Deep Learning ayant fait de réels bonds depuis 2010, plusieurs architectures de réseaux de neurones auraient pu être essayées. Ces modèles nécessitent beaucoup de données pour être entraînés et c’est le cas ici.