TABLE DES MATIERES

INTRODUCTION

Le machine learning est une technique de programmation se servant des probabilités statistiques pour donner aux ordinateurs la capacité d'apprendre par eux-mêmes sans programmation explicite.

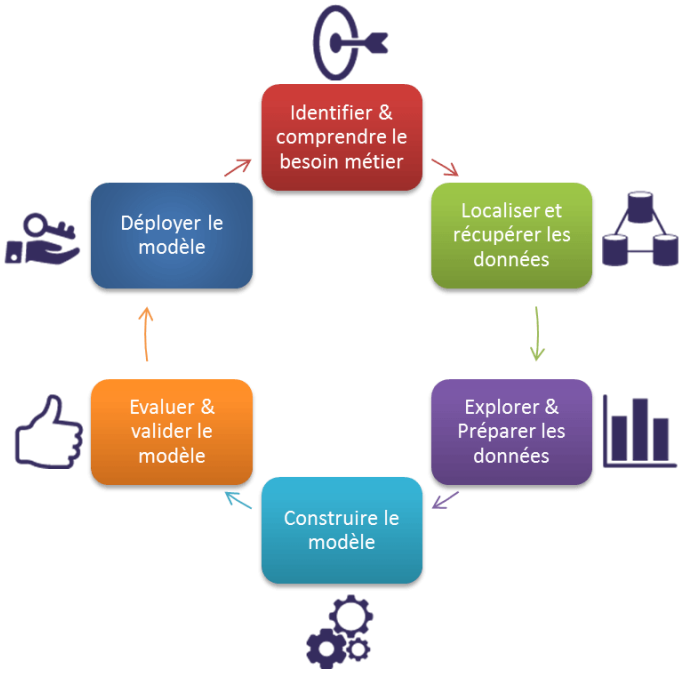
Nous sommes dans le cas d'un problème pour lequel on doit décider si l'on va accorder un crédit à un client ou non selon ses caractéristiques.

Nous considérons ici un problème de machine learning supervisé : un problème dit de classification.

Ce problème de classification comprend en entrée des données variées (données comportementales, données provenant d'autres institutions financières, etc.) et en sortie la solvabilité ou non du client.

Pour pouvoir classer au mieux les clients, il faut créer un modèle statistique associé aux données. Il faut ensuite évaluer et interpréter les résultats, évaluer la qualité du modèle.

Pour cela, il nous faut expliciter les étapes du travail de datascientist :

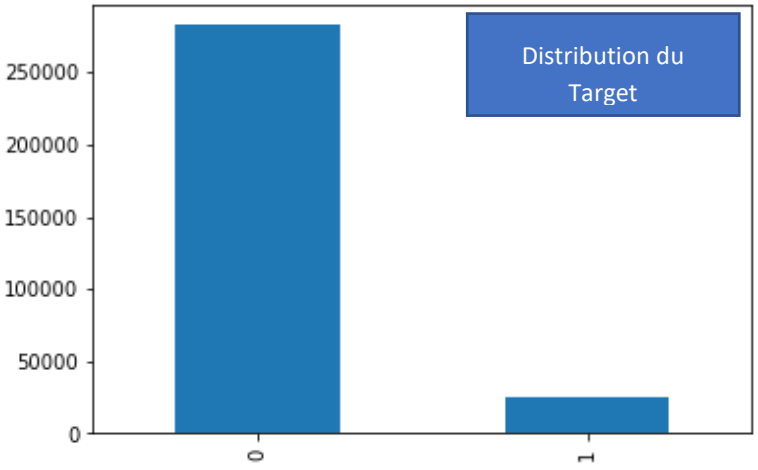


Nous allons donc présenter ici les étapes de :

- Préparation des données,
- Construction du modèle,
- Evaluation et validation du modèle

LES ETAPES DE LA MODELISATION DU PRE-TRAITEMENT A LA PREDICTION :

Nous allons dans un premier temps, nous intéresser à notre problème de classification, l'analyse exploratoire nous montre qu'il y a un fort déséquilibre de classe dans la variable cible : Target.



La classe 0 représente 92% du Target alors que la classe 1 ne représente que 8% de celui-ci. Pour faire face à ce déséquilibre, on peut utiliser la technique de rééchantillonnage en augmentant de façon artificielle le nombre d'instance de la classe minoritaire. Nous allons donc utiliser la technique du SMOTE de façon à créer pour la classe minoritaire des échantillons synthétiques.

Maintenant que nous avons identifier le déséquilibre des classes, nous allons traiter des différentes étapes de la modélisation.

L'ETAPE DU PRE-TRAITEMENT

LE NETTOYAGE ET LA TRANSFORMATION DES DONNEES :

Nous allons identifier les valeurs manquantes et les valeurs aberrantes et ne traiter que les grandes valeurs aberrantes (remplacer chacune de ces valeurs par la valeur nan).

Nous allons ensuite créer des features manuellement.

Puis nous allons créer des features automatiquement en utilisant la fonction d'agrégation « mean » (pour tous les datasets autres que les datasets train et test).

Enfin, nous fusionnons tous les datasets autres que les datasets app_train et app_test, pour obtenir un dataset intermédiaire. Nous fusionnerons celui-ci avec le dataset app_train d'un côté et avec le dataset app_test d'un autre côté de façon à n'avoir que deux datasets finaux.

Nous échantillonnons le dataset résultant de la fusion avec app_train en deux datasets pour les données d'entrainement (80%) et pour les données de test (20%). Nous appellerons le dataset d'entrainement train.

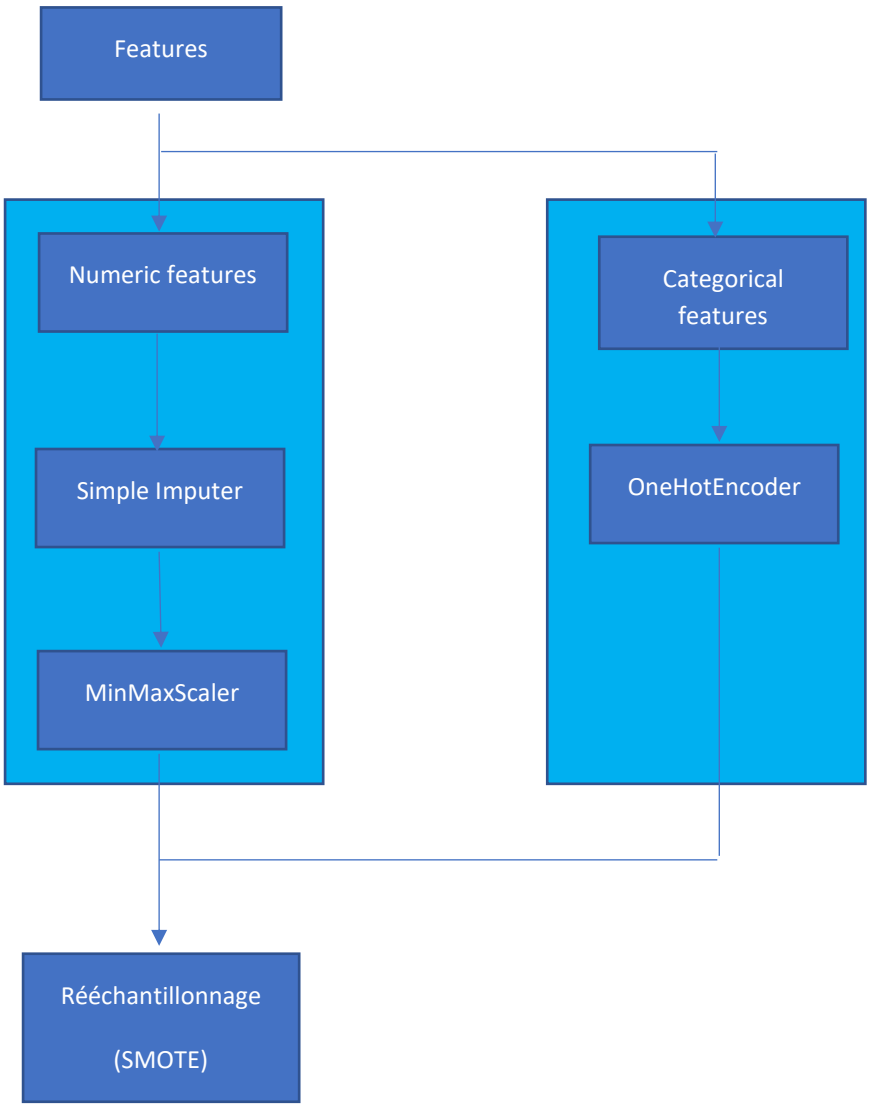
LE PREPROCESSING DES DONNEES :

Nous allons entraîner le modèle uniquement sur le dataset résultant de la fusion avec le dataset train. Nous faisons une imputation des valeurs nulles par la médiane. Nous réalisons un encodage des données catégorielles. Nous faisons ensuite une mise à l'échelle des données quantitatives

LA TECHNIQUE DE REECHANTILLONNAGE

Nous appliquons la technique de rééchantillonnage SMOTE.

Nous pouvons résumer le preprocessing des données par le schéma suivant



LE MODELE DE MACHINE LEARNING

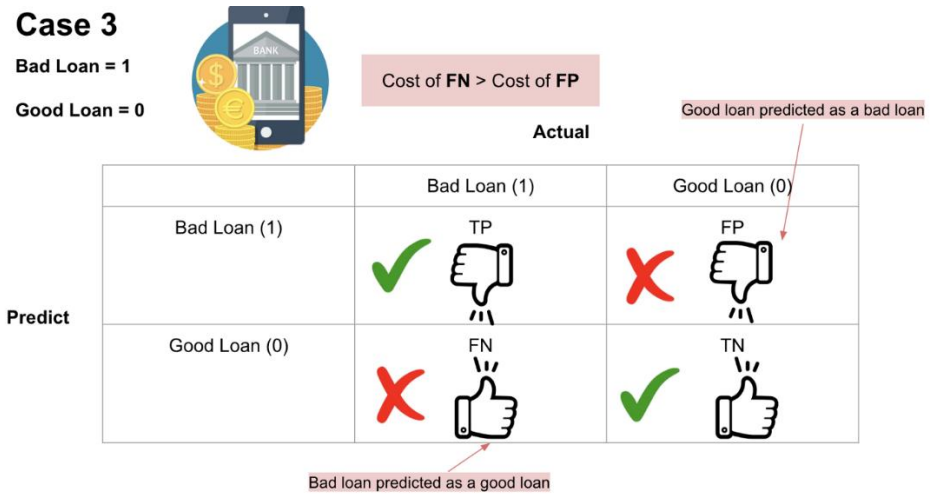
Le modèle choisi ici est le modèle de régression logistique. La régression logistique est répandue dans le domaine bancaire pour détecter des groupes à risque lors de la souscription de crédit.

EVALUATION DU MODELE DE CLASSIFICATION

Le modèle de classification classe ses résultats en deux catégories : les clients solvables et les clients non solvables.

LA MATRICE DE CONFUSION, RECALL, PRECISION

Nous allons utiliser la matrice de confusion pour évaluer le modèle de classification. La matrice de confusion consiste à afficher dans un tableau le nombre d'enregistrements en fonction de leur classe réelle et prévue.



Vrai positif (TP) : la catégorie réelle et la catégorie prédite sont toutes deux *positives*. Nombre de fois où le modèle a correctement classé un client non solvable.

Vrai négatif (TN) : la catégorie réelle et la catégorie prévue sont toutes deux *négatives*. Nombre de fois où le modèle a correctement classé un client comme *solvable*.

Faux positif (FP) : la catégorie réelle est *négative* et la catégorie prédite est *positive*. Nombre de fois où le modèle a classé un e-client comme *non solvable*, alors qu'il est *solvable*

Faux négatif (FN) : la catégorie réelle est *positive* et la catégorie prédite est *négative*. Nombre de fois où le modèle a classé un client comme *solvable*, alors qu'il est *non solvable.*

La façon dont un algorithme est optimisé dépend fortement de ce que l'entreprise essaie de réaliser. Le but de ce modèle est de trouver tous les clients ne remboursant pas leur prêt, c'est-à-dire les TP.

On cherche donc à maximiser le Recall = $\frac{TP}{TP+FN}$ (la proportion de positif réels que le modèle classe correctement) et la précision = $\frac{TP}{TP+FP}$ (mesure de l'exactitude du résultat positif prévu). Ce sont les 2 métriques qui nous intéressent.

Le coût des FN étant supérieur au coût des FP, nous cherchons en premier lieu à minimiser les FN et donc maximiser le Recall.

LA FONCTION COUT

De plus, accorder un prêt à un client non solvable (FN) et refuser un prêt à un client solvable (FP) coûte de l'argent à l'entreprise. On remarque, cependant, que l'on perd moins d'argent en prédisant positif alors que le client est négatif qu'en prédisant un client négatif alors qu'il est positif. Créons donc une fonction gain (=- fonction coût) de façon à rendre compte du coût de chaque erreur :

$$J = C_0 * TN + C_1 * FN + C_2 * FP + C_3 * TP$$

Où C_0 , C_1, C_2, C_3 représentent les gains respectifs de TN, FN, FP et TP.

(Pour retourner une prédiction binaire, il faut seuiller : si le score retourné est supérieur au seuil, alors on prédit positif ; s'il est inférieur, on prédit négatif.)

Cherchons à maximiser la fonction gain en fonction du seuil (on pourrait aussi minimiser la fonction coût).

Prenons comme valeurs les valeurs arbitraires des gains de façon que les :

- Les FN engendrent des pertes de : -10
- Les FP n'engendrent pas de gain de 0
- Les TN engendrent des gains de :1
- Les TP n'engendrent pas de gain 0

(Ces valeurs ont été choisies de manière arbitraire.)

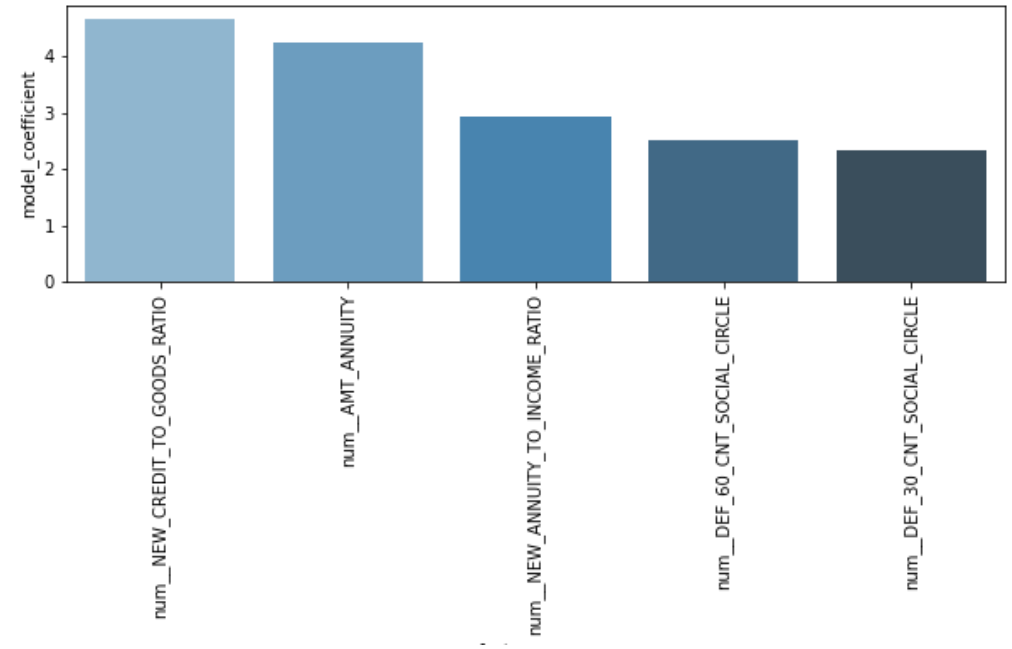
Nous obtenons, ici, un seuil optimal de 0.53.

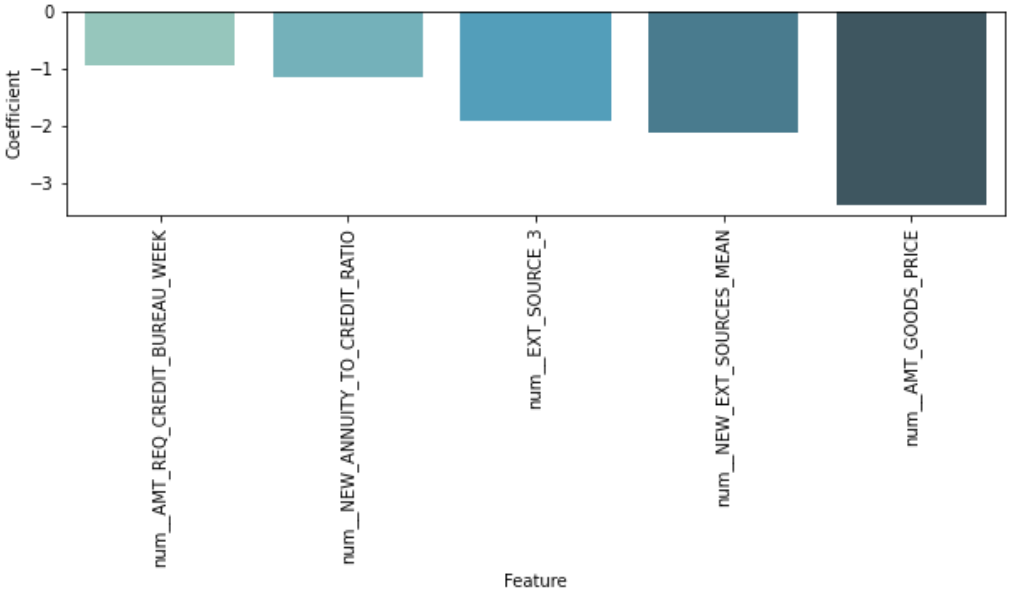
INTERPRETABILITE DU MODELE

L'une des caractéristiques déterminantes de la régression logistique est l'interprétabilité que nous avons à partir des coefficients de caractéristiques . Etant donné que nos données sont normalisées, toutes les caractéristiques varient sur la même plage. Nous pouvons comparer les amplitudes et les signes des coefficients de caractéristiques pour déterminer quelles caractéristiques ont le plus grand impact sur la prédiction de classe, et si cet impact est positif ou négatif.

- Les caractéristiques avec des coefficients positifs plus grands augmenteront la probabilité qu'un échantillon de données appartienne à la classe positive.
- Les caractéristiques avec des coefficients négatifs plus grands réduiront la probabilité qu'un échantillon de données appartienne à la classe positive.
- Les caractéristiques avec des coefficients faibles, positifs ou négatifs ont un impact minimal sur la probabilité qu'un échantillon de données appartienne à la classe positive.

Regardons dans notre cas quels sont les coefficients les plus larges positifs et négatifs :


model_coefficient
0
1
2
3
4
num__NEW_CREDIT_TO_GOODS_RATIO
num__AMT_ANNUITY
num__NEW_ANNUITY_TO_INCOME_RATIO
num__DEF_60_CNT_SOCIAL_CIRCLE
num__DEF_30_CNT_SOCIAL_CIRCLE


Coefficient
0
−1
−2
−3
num__AMT_REQ_CREDIT_BUREAU_WEEK
num__NEW_ANNUITY_TO_CREDIT_RATIO
num__EXT_SOURCE_3
num__NEW_EXT_SOURCES_MEAN
num__AMT_GOODS_PRICE
Feature

On voit ici que le rapport $\frac{Montant\ du\ crédit\ du\ prêt}{Prix\ des\ biens\ pour\ lesquels\ le\ crédit\ est\ accodé}$ est grand plus les chances d'appartenir à la classe non solvable est grande ; même chose pour les variables « annuité du prêt », le rapport $\frac{Annuité\ du\ prêt}{Revenu\ du\ client}$. On voit aussi plus la variable AMT_GOODS_PRICE est grande, plus le rapport $\frac{Annuité\ du\ prêt}{Montant\ du\ crédit\ du\ prêt}$ est grand, moins les chances d'appartenir à la classe non solvable sont grandes.

Nous calculons ensuite le nombre d'Euler à la puissance de ses coefficients ce qui nous donne un graphique représentant l'importance des fonctionnalités. La caractéristique la plus forte dans l'ensemble de données de ce jeu est le ratio new_credit_to_goods_ratio. Une augmentation de ce ratio d'une unité augmente les chances d'être de classe non solvable d'un facteur de 100 lorsque toutes les autres caractéristiques restent les mêmes.

LES LIMITES ET AMELIORATIONS POSSIBLES

Le choix du modèle de régression logistique n'est certainement pas le plus performant, il faudrait tester d'autres modèles et les comparer afin de choisir le mieux adapté.

Dans ce modèle, on a choisi de garder toutes les variables, trop de variables, cela peut jouer sur l'interprétation de celles-ci. Il faudrait donc réduire le nombre de variables en les sélectionnant pendant le travail exploratoire.

Dans ce modèle une matrice de coût arbitraire est proposée, il faudrait contacter le service métier pour avoir des valeurs plus proches de la réalité.