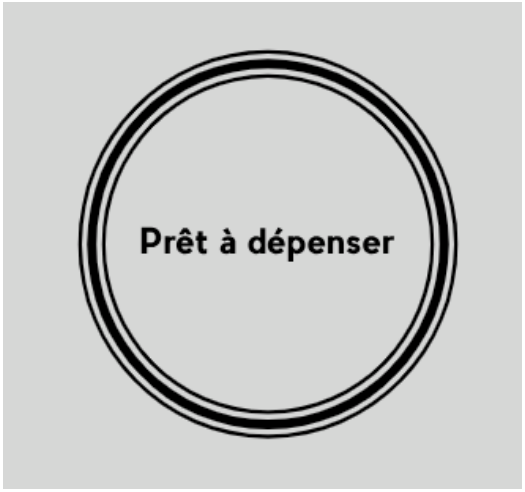
Openclassrooms

Formation Data scientist 06/2022

note méthodologique

Projet n°7 : Implémentez un modèle de scoring



Bayram DONAT

1. La méthodologie d'entraînement du modèle

1.1. Traitement

Nous avons passé les variables catégorielles en numériques par label encoder (2 classes) et one hot encoder (plus de 2 classes). Nous avons traité les données manquantes en imputant par la méthode des plus proches voisins ou la médiane de la colonne. Nous avons traité les valeurs aberrantes par la méthode des interquantiles : en imputant les valeurs max et min des 25 % à 75 % de la population aux valeurs aberrantes. Nous avons ensuite standardisé ou normalisé les valeurs. Nous avons eu de meilleurs résultats avec la normalisation. Nous avons enfin, divisé la base de donnéees en 2 : une partie pour l'entrainement et une partie pour le test (20%).

1.2. Equilibrage

Etant avec une variable TARGET très déséquilibrée, il fallait éviter le sur-apprentissage, le sous apprentissage. Nous avons opté pour le SMOTE (Synthetic Minority Oversampling Technique). Consiste à synthétiser des éléments pour la classe minoritaire, à partir de ceux qui existent déjà en choisissant aléatoirement un point de la classe minoritaire et à calculer les k plus proches voisins de ce point. Dans notre cas de figure, nous avions une base de 1 TARGET =1 pour 10 TARGET =0.

1.3. Entrainement des modèles

Le but du projet est de classifier la variable TARGET dans 2 classes : 0 ou 1. Pour ceci, nous allons utiliser 5 classifiers. Pour chacun, nous allons rechercher les hyperparamètres (1 ou 2 par modèle) à l'aide de l'outil gridsearchcv, ensuite, nous allons évaluer les performances des modèles par une matrice de confusion, une courbe roc et des métriques d'évaluation. Dans notre cas, nous avons comparer les performances de 5 classifiers et avons choisi le plus performants avec les meilleurs résultats : le classifier LGBM.

- régression logistique
- arbre de décision
- forêt aléatoire
- Gradient boosting
- Light gradient boosting machine (LGBM)

2. La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation

2.1. fonction coût métier

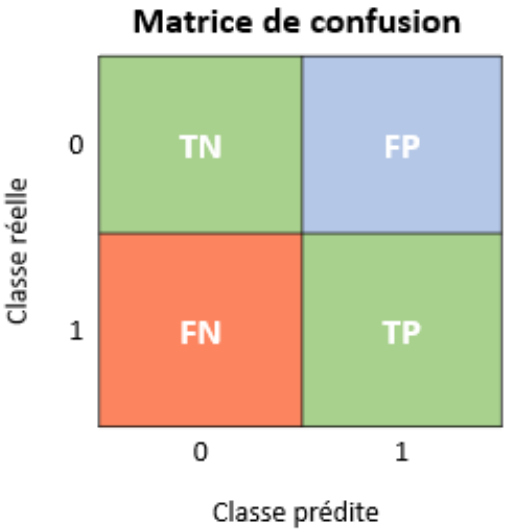
Pour ce projet, une fonction coût a été développée dans l'objectif de pénaliser des Faux Négatifs.

• TN (Vrais Négatifs) : des prêts qui **ne sont pas en défaut** et ont été **prédits correctement**

• TP (Vrais Positifs) : sont **en défaut** et ont été **prédits correctement**

• FP (Faux Positif) : des prêts qui **ne sont pas en défaut** et ont été **prédits** de manière **incorrecte**

• FN (Faux Négatif) : des prêts qui **sont en défaut** et ont été **prédits** de manière **incorrecte**



Un **Faux Positif** (FP) constitue une **perte d'opportunité** pour la banque, à la différence d'un **Faux Négatif** (FN) qui constitue une **perte pour créance irrécouvrable**.

La fonction coût sera utilisé au moment de calculer le seuil et au niveau de l'évaluation du modèle par une métrique d'évaluation

2.2. algorithme d'optimisation

Une fois notre modèle choisi. Nous allons d'abord regarder l'importance des variables (features) pour notre modèle. Ce qui nous donne un nombre de 118 variables (features) importantes. Ensuite nous allons éliminer les variables jusqu'à en obtenir 118 justement par la méthode RFE. Ceci va nous donner un dataframe optimisé qui donnera des résultats très similaires avec un nombre de features divisé par 2.

Ensuite, on va chercher à obtenir le seuil de décision en cherchant la valeur qui donne un F-beta score au maximum. Dans notre cas de figure, on l'obtient pour un seuil de décision de 0.103448

Si nous avons une prédiction de score > au seuil de décision, nous accordons le credit. Si nous n'obtenons pas ce score, nous refusons le crédit.

2.3. métriques d’évaluation

Les métriques utilisées sont les suivantes en fonction des valeurs de la matrice confusion:

- Recall score=TP/ (TP+FN)

  =>Quand le recall est **haut**, cela veut plutôt dire qu’**il ne ratera aucun positif**. Néanmoins cela ne donne aucune information sur sa qualité de prédiction sur les négatifs.

- Precision score=TP/ (TP+FP)

  => Quand la précision est **haute**, alors **la majorité des prédictions positives du modèle sont des positifs bien prédit.**

- F1 score= TP/ [TP+1/2.(FN+FP)]

- Fbeta score=TP/ [TP+1/(1+beta²).(beta².FN+FP)]

  => si beta=1 , Fbeta score=F1 score, autant d’importance pour precision et recall

  => si beta>1 , recall plus important que precision

  => si beta<1 , precision plus important que recall

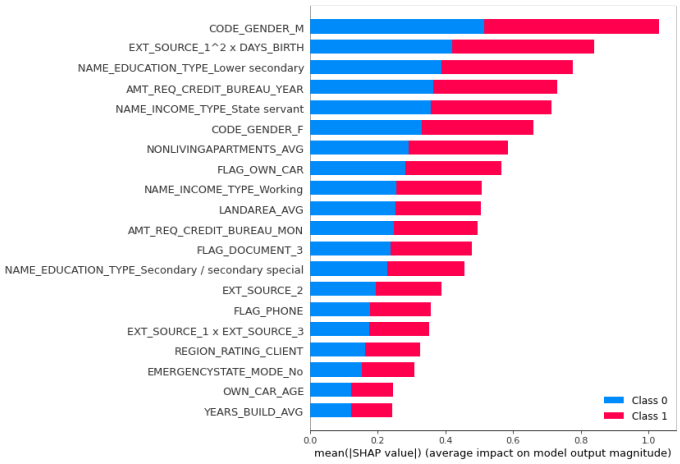
- L'AUC (l'aire sous la courbe ROC) est la mesure de la capacité d'un classificateur à distinguer les classes et correspond au résumé de la courbe ROC. Plus l'AUC est élevée, plus la performance du modèle à distinguer les classes positives et négatives est bonne.

3. L’interprétabilité globale et locale du modèle

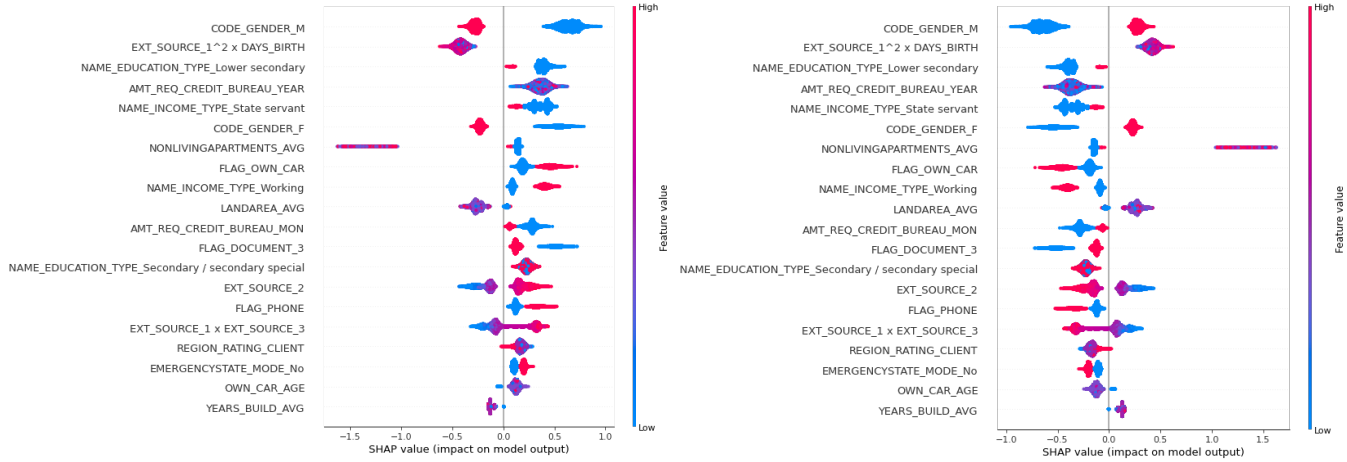
3.1. l’interprétation globale

Le graphique ci-dessous montre l’importance des variables pour notre modèle de type light gradient boosting model classifier.

Dans les variables créées, nous voyons l’importance de la variable CODE_GENDER, de la variable polynomiale créée EXT_SOURCE_1²xDAYS_BIRTH ou la variable NAME_EDUCATION_TYPE secondaire bas. Nous pouvons nous demander justement pourquoi le sexe de la personne joue sur le modèle.



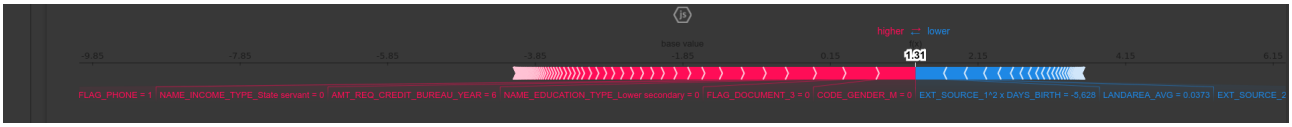
Les graphique ci-dessous montrent l’importance de la variable pour la valeur de la TARGET. A gauche, quand la TARGET vaut 0. A droite, quand la TARGET vaut 1. Si l’on prend CODE_GENDER_M (sexe masculin), nous voyons une répartition plus importante pour la TARGET nulle que pour la TARGET égal à 1.



3.2. l’interprétation locale

Pour le client ci-dessous, les variables importantes sont FLAG_ON_CAR=1,CODE_GENDER_f=0, NAME_FAMILY_STATUS_married=1, FLAG_PHONE=1, EXT_SOURCE_3, REGION_RATING_CLIENT=2, FLAG_WORK_PHONE=0OBS_30_CNT_SOCIAL_CIRCLE=0...

Les variables sans importance sont EXT_SOURCE_2=0.189, CREDIT_DURATION=256, CODE_GENDER_m=1, EXT_SOURCE_1…



4. Les limites et les améliorations possibles

La modélisation a été effectuée sur la base d'une métrique personnelle créée pour répondre au mieux au besoin de gain d'argent d'une banque. Les coefficients de cette métrique ont été choisis arbitrairement selon le bon sens. L'axe principal d'amélioration serait donc de définir plus précisément ces coefficients associés à chaque combinaison classe prédite/classe réelle car le modèle déterminé ici ne sera pas obligatoirement le meilleur.

Une connaissance approfondie sur les variables relevées dans les fichiers CSV permettra de faire un choix plus judicieux qu'en aux variables importantes et à la cration de variables fonctionnelles et polynomiales. Les échanges avec des spécialistes financiers pourra être nécessaire.

D'autre part, il est nécessaire de bien choisir les hyperparamètres pour optimiser son modèle. Pour exécuter des calculs plus poussé, il va être honorable d'utiliser des serveurs plus performants.

Enfin, le Dashboard interactif pourra être amélioré en collaborations avec les utilisateurs de premier rang : les conseillers clientèles qui pourront mieux choisir les informations à rechercher et afficher pour le travail de présentation des résultats à leurs clients.