NOTE METHODOLOGIQUE

Implémentez un modèle de scoring

Eric Blanvillain

18/02/2022

Diagram, shape, circle

Description automatically generated

TABLE DES MATIERES

1. SYNOPSIS…………………………………………………………………………………………………………………………….3

2. PRESENTATION DES DONNEES ................................................................................................ 1

Arborescence des tables de données...........................................................................................1

Modèle de classification et cible (TARGET)..................................................................................1

3. PREPARATION DES DONNEES .................................................................................................. 2

Pipeline de pré-traitement appliqué aux tables ......................................................................... 2

Assemblage des tables et sélection de variables..........................................................................3

4. MODELISATION ........................................................................................................................ 4

Méthodologie de rééquilibrage de la variable cible ....................................................................4

Séparation des données................................................................................................................5

Choix de l’algorithme....................................................................................................................5 Problématique « métier » : métrique « bancaire » et seuil de solvabilité……………………………….6

Méthode d’optimisation des hyperparamètres de LightGBM.....................................................6

Modèle finalisé..............................................................................................................................7

Fixation du seuil de solvabilité optimum......................................................................................9

5. INTERPRETABILITE ................................................................................................................. 10

Importance relative des variables pour le modèle ................................................................... 10

1. CONCLUSION – LIMITES ET AMELIORATIONS .............................................................. 10
2. PRESENTATION DU DASHBOARD .................................................................................. 11
3. LIENS............................................................................................................................... 12

Déploiement du dashboard ...................................................................................................... 12

SYNOPSIS

**Introduction**

Vous êtes Data Scientist au sein d'une société financière, nommée "Prêt à dépenser", qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt.

L’entreprise souhaite mettre en œuvre un outil de “scoring crédit” pour calculer la probabilité qu’un client rembourse son crédit, puis classifie la demande en crédit accordé ou refusé. Elle souhaite donc développer un algorithme de classification en s’appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).

De plus, les chargés de relation client ont fait remonter le fait que les clients sont de plus en plus demandeurs de transparence vis-à-vis des décisions d’octroi de crédit. Cette demande de transparence des clients va tout à fait dans le sens des valeurs que l’entreprise veut incarner.

Prêt à dépenser décide donc de développer un dashboard interactif pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d’octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

**Les données**

Voici les données dont vous aurez besoin pour réaliser le dashboard. Pour plus de simplicité, vous pouvez les télécharger à cette adresse.

Vous aurez sûrement besoin de joindre les différentes tables entre elles.

**Votre mission**

Construire un modèle de scoring qui donnera une prédiction sur la probabilité de faillite d'un client de façon automatique.

Construire un dashboard interactif à destination des gestionnaires de la relation client permettant d'interpréter les prédictions faites par le modèle, et d’améliorer la connaissance client des chargés de relation client.

Michaël, votre manager, vous incite à sélectionner un kernel Kaggle pour vous faciliter la préparation des données nécessaires à l’élaboration du modèle de scoring. Vous analyserez ce kernel et l’adapterez pour vous assurer qu’il répond aux besoins de votre mission.

Vous pourrez ainsi vous focaliser sur l’élaboration du modèle, son optimisation et sa compréhension.

**Spécifications du dashboard**

Michaël vous a fourni des spécifications pour le dashboard interactif. Celui-ci devra contenir au minimum les fonctionnalités suivantes :

Permettre de visualiser le score et l’interprétation de ce score pour chaque client de façon intelligible pour une personne non experte en data science. Permettre de visualiser des informations descriptives relatives à un client (via un système de filtre). Permettre de comparer les informations descriptives relatives à un client à l’ensemble des clients ou à un groupe de clients similaires.

**Livrables**

Le dashboard interactif répondant aux spécifications ci-dessus et l’API de prédiction du score, déployées chacunes sur le cloud.

Un dossier sur un outil de versioning de code contenant : Le code de la modélisation (du prétraitement à la prédiction) Le code générant le dashboard Le code permettant de déployer le modèle sous forme d'API Une note méthodologique décrivant : La méthodologie d'entraînement du modèle (2 pages maximum) La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation (1 page maximum) L’interprétabilité globale et locale du modèle (1 page maximum) Les limites et les améliorations possibles (1 page maximum) Un support de présentation pour la soutenance, détaillant le travail réalisé.

Pour faciliter votre passage au jury, déposez sur la plateforme, dans un dossier nommé “P7\_nom\_prenom”, tous les livrables du projet. Chaque livrable doit être nommé avec le numéro du projet et selon l'ordre dans lequel il apparaît, par exemple “P7\_01\_dashboard”, “P7\_02\_dossier”, et ainsi de suite.

**Soutenance**

Pendant la soutenance, l’évaluateur jouera le rôle de Michaël, à qui vous présentez votre travail.

Présentation (20 minutes) Rappel de la problématique et présentation du jeu de données (5 minutes) Explication de l’approche de modélisation (10 minutes) Présentation du dashboard (5 minutes) Discussion (5 minutes) L’évaluateur, jouant le rôle de Michaël, vous challengera sur vos choix. Débriefing (5 minutes)

À la fin de la soutenance, l'évaluateur arrêtera de jouer le rôle de Michaël pour vous permettre de débriefer ensemble.

**Skills**

1. Présenter son travail de modélisation à l'oral
2. Réaliser un dashboard pour présenter son travail de modélisation
3. Rédiger une note méthodologique afin de communiquer sa démarche de modélisation
4. Utiliser un logiciel de version de code pour assurer l’intégration du modèle
5. Déployer un modèle via une API dans le Web

PRESENTATION DE LA DONNEE

1. **Arborescence des tables de données**

Sept tables de données sont mises à disposition pour mener à bien notre démarche. Elles sont constituées de données anonymisées d’informations personnelles et bancaires des clients.

Timeline

Description automatically generated

**FIGURE 1 – Arborescence des 7 tables de données fournies par Home Crédit**

1. **Modèle de classification et cible**

La valeur à prédire est contenue dans la variable « **TARGET** ». Elle peut prendre 2 valeurs :

* 1 qui indique que le client n’a pas remboursé son crédit. (Défaillant, 8%)
* 0 qui indique que le client l’a remboursé. (Non défaillant, 92%)

Ce déséquilibre devra être pris en compte lors de la construction du modèle, car certains algorithmes sont sensibles au **déséquilibre**.

Chart, pie chart

Description automatically generated

**FIGURE 2– Répartition de la variable TARGET**

PREPARATION DES DONNEES

1. **Pipeline de pré-traitement appliqué aux tables**

Chaque table est traitée indépendamment. Des **anomalies** détectées pendant l’analyse exploratoire des données sont corrigées. Cette analyse a permis de constater qu’un tiers des variables totales ont 50% de valeurs manquantes.

Une double stratégie est appliquée **: suppression de colonnes** (NAN > 60%) et **imputation** selon le type de données.

Diagram

Description automatically generated

**FIGURE 3 – Pipeline de pré-traitement appliqué aux tables**

Deux processus de création manuelle de nouvelles variables (**feature enginneering**) sont appliqués :

Création systématique de variables statistiques créées à partir des variables existantes : (moyenne, minimum, maximum, écart-type, compte)

Création de variables « métier » basées sur les connaissances du domaine bancaire :

* **Ratios** : montant total du crédit/revenu du client, montant crédit/intérêt du prêt, ancienneté au travail/âge, nombre d’enfants/revenu, revenu par tête
* **Sommes** : des 3 sources extérieures, des docs (flags) à fournir, des contacts client (adresse, téléphones, mail)
* **Différences** : âge - ancienneté emploi
* **Booléen** : crédit > somme demandée ? (Credit > GoodPrice)
* **Variable liée à la cible** : moyenne de la cible pour les 500 voisins les plus proches de chaque client en fonction de 4 variables : les 3 sources extérieures (EXT\_SOURCE 1, 2 et 3) et un ratio créé par nos soins (ratio CREDIT/ANNUITY)

Les données sont ensuite encodées selon leur type et leur signification :

Variables qualitatives : transformation en variables quantitatives pour les rendre exploitables

* **Label encoder** : encodage binaire (0 ou 1) (modalités < 3)
* **One Hot Encoder** : pour les autres variables qualitatives

Variables quantitatives : mise à l’échelle : ensemble des valeurs ramenées de 0 à 1

* **MinMaxScaler** : pour pouvoir tester nos données sur des algorithmes de machine learning sensibles aux variations d’échelle des variables lors de la phase de sélection du meilleur modèle.

1. **Assemblage des tables et sélection de variables**

Les 7 tables sont assemblées (**merging**). Elles contenaient initialement 218 variables au total. Après la phase de création de variables, nous avons systématiquement supprimé les variables colinéaires à plus de 80%. Nous obtenons alors un **jeu de données** final de 365 variables.

Plusieurs approches ont été utilisées pour sélectionner les variables les plus importantes :

* + Méthode de filtrage (suppression des variables colinéaires)
  + Méthodes automatiques utilisant des packages python : Boruta et BoostAroota
  + Méthode « embedded » ou intégrées qui effectuent la sélection des fonctionnalités pendant la formation du modèle. Ici le modèle utilisé est LGBM non optimisé.

Table

Description automatically generated

**FIGURE 4 – Sélection de variables**

-> Après sélection, en ne conservant que les variables communes aux 3 méthodes utilisées, le jeu de données contient 109 variables et une ligne par client.

Modélisation

1. **Méthodologie de rééquilibrage de la variable cible**

L’analyse exploratoire a montré qu’il s’agissait d’un problème de classification déséquilibré : le choix d’une métrique de performance est donc essentiel pour évaluer correctement les modèles.

Une modification de l’ensemble de données est possible avant d’entraîner le modèle prédictif afin **d’équilibrer les données**. Cette stratégie est appelée **rééchantillonnage** (re-sampling) et il existe deux méthodes principales pour égaliser les classes : le sur-échantillonnage : **Oversampling** et le sous- échantillonnage : **Undersampling**.

Text

Description automatically generated with low confidence

**FIGURE 5 – Stratégies de rééchantillonnage**

Chart

Description automatically generated

**FIGURE 6 – Sur-échantillonnage SMOTE appliqué au jeu de données**

Une métrique d’évaluation quantifie la performance d’un modèle prédictif.  
La qualité d’un modèle de classification dépend directement de la métrique utilisée pour l’évaluer. Elle consiste globalement à comparer les classes réelles aux classes prédites par le modèle ou permettre d’interpréter les probabilités prédites pour ces classes.

1. **Séparation des données**

Le jeu de données est séparé en jeu d’entrainement (80% des données) et en jeu de validation (20%) en veillant à maintenir les proportions de classes 0 et 1 dans les 2 jeux. Le **jeu de validation** sera utilisé pour mesurer la performance des modèles testés.

1. **Choix de l’algorithme**

Nous avons testé un ensemble de modèles grâce à la librairie **PyCaret** : un **modèle linéaire**, **régression logistique**, deux modèles ensemblistes, **Random Forest** et **Adaboost**, et deux modèles ensemblistes « avancés » avec gradient boosting, **Catboost** et **LightGBM**.

La métrique utilisée est le score AUC ou « aire sous la courbe ROC ». Elle est exploitée en classification binaire pour mesurer la performance d’un modèle : plus il est performant, plus l’aire sous la courbe est maximisée.

Graphical user interface, application, website

Description automatically generated

**FIGURE 7 – Courbe ROC**



**FIGURE 8 – Tableau de comparaison d’algorithmes établi grâce à la librairie Pycaret**

Les deux algorithmes Catboost et LightGBM ont été **optimisés**.

-> C’est finalement **LightGBM**, plus rapide, qui a été retenu.

1. **Problématique « métier » : métrique « bancaire » et seuil de solvabilité**

Dans notre problème de classification binaire, le coût des **faux positifs** n'est pas le même que celui des **faux négatifs**. (erreurs de type 1 et de type 2)

**Graphical user interface, application, table

Description automatically generated**

**FIGURE 9 – Matrice de confusion**

Nous pouvons optimiser notre classificateur pour une valeur **seuil de probabilité** afin d'optimiser la fonction de perte personnalisée simplement en définissant le **coût** des vrais positifs, des vrais négatifs, des faux positifs et des faux négatifs séparément. Pour cela, il faut :

-> Créer une **métrique "bancaire"** qui **pénalisera** les faux négatifs, c'est à dire les clients prédits non défaillants par le modèle et qui se révèlent en réalité défaillants. (Mauvais prêts)

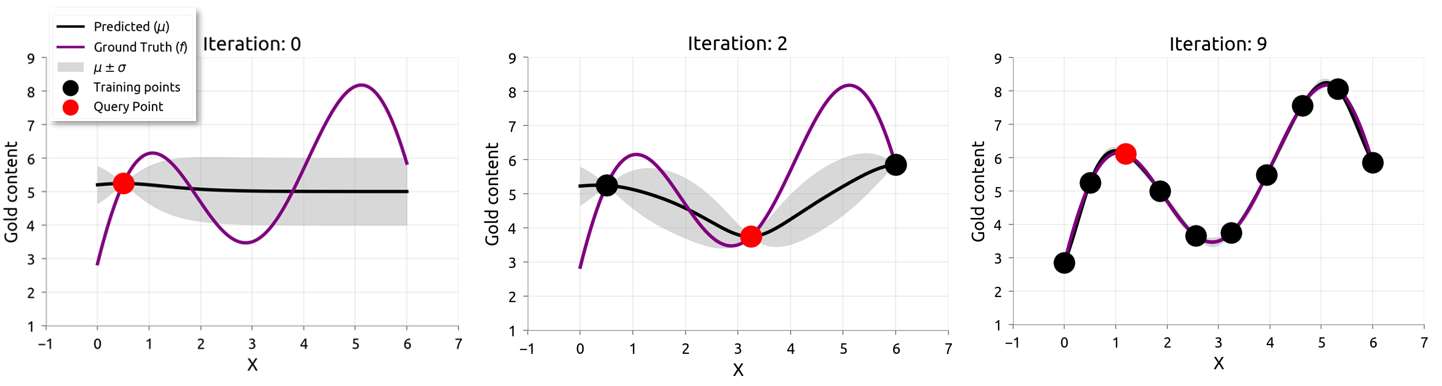
Choix arbitraire de pénalisation :

* Mauvais prêts : pénalisation de -10
* Bons prêts : gain de 1

-> Chercher et fixer le seuil de solvabilité optimum en fonction de notre métrique bancaire.

1. **Méthode d’optimisation des hyperparamètres de LightGBM**

La méthode d’optimisation des hyperparamètres choisie est l’optimisation bayésienne.

**FIGURE 10 – Optimisation bayésienne (images : https://distill.pub/2020/bayesian-optimization/)**

L’optimisation bayésienne construit un modèle de probabilité de la fonction objectif afin de proposer des choix plus intelligents pour le prochain ensemble d'hyperparamètres à évaluer. Au fur et à mesure que le nombre d'observations augmente, la distribution postérieure s'améliore et l'algorithme devient plus sûr des régions de l'espace des paramètres qui valent la peine d'être explorées et de celles qui ne le sont pas.

L’optimisation a été faite selon 2 métriques pour LightGBM :

Modèle 1 : sur la métrique roc\_auc  
**Modèle 2 :** métrique « bancaire » créée par nos soins

Et sur 2 jeux de données différents :

**Jeu 1 :** jeu de données sans rééquilibrage des données et sans standardisation (LGBM est insensible aux variations d’échelle) Lire note class\_weight ci-dessous.  
**Jeu 2 :** jeu de données obtenu après sur-échantillonnage SMOTE

Un modèle de base réglé sur les hyperparamètres par défaut de LightGBM sert de comparatif.

1. LightGBM possède un hyperparamètre « class\_weight » que l’on peut régler sur « balanced » afin qu’il tienne compte du déséquilibre des classes.
2. Un œil attentif sera gardé sur deux métriques : le rappel et la précision. Le but dans le cas d'une classification des prêts bons ou mauvais est de maximiser le rappel au détriment de la précision : diminuer les faux négatifs (FN) pour augmenter le rappel.
3. **Modèle finalisé**

Les deux modèles **LightGBM**, réglés sur les **métriques AUC** ou sur la **métrique bancaire**, sans rééchantillonnage mais avec **l’hyperparamètre class\_weight=’balanced’** se sont montrés les plus performants.

A noter : ces deux modèles ont un meilleur **score de rappel** que les modèles de base et SMOTE, mais un moins bon **score de précision**. Ils détectent mieux les **faux négatifs**, mais on observe en parallèle une augmentation des **faux positifs**.

A picture containing calendar

Description automatically generated

**FIGURE 11 – Scores LGBM class\_weight versus SMOTE**

Graphical user interface, application, website, Teams

Description automatically generated

**FIGURE 12 – Matrices de confusion LightGBM class\_weight, jeu de validation**

Un arbitrage est à faire, en collaboration avec notre client, pour décider du modèle à privilégier et en particulier quel équilibre optimum on doit trouver entre les faux négatifs et les faux positifs.

Le modèle LightGBM « métrique bancaire » détecte moins de faux négatifs que le modèle LightGBM « ROC\_AUC ». Il prédit plus souvent que le client va rembourser un prêt alors qu’il va se révéler défaillant. L’organisme prêteur perd ainsi la somme prêtée.

Il est en revanche meilleur sur les faux positifs. En ne prêtant pas à des clients finalement solvables, la banque perd les intérêts que ses clients auraient versés.

Le modèle conservé est donc le modèle LightGBM optimisé sur la métrique roc\_auc. La métrique bancaire sera utilisée pour chercher le seuil de solvabilité.

Table

Description automatically generated

**FIGURE 13 – Hyperparamètres et performance du modèle final**

1. **Fixation du seuil de solvabilité optimum**

Graphical user interface, chart

Description automatically generated with medium confidence

**FIGURE 14 – Exemple explicatif de la notion de seuil et Matrice de confusion, modèle final, seuil=0.41, jeu de validation**

Notre modèle fait la prédiction que le client demandeur a une probabilité de 50% d’appartenir à la classe 1, c’est-à-dire d’être défaillant. En plaçant le seuil à 0.41, ce client sera prédit défaillant.

INTERPRETABILITE

1. **Importance relative des variables pour le modèle**

Table

Description automatically generated

**FIGURE 15 – Les 20 variables les plus importantes pour le modèle final**

Sans surprise, la variable « TARGET\_NEIGHBORS\_500\_MEAN » créée en calculant la moyenne de la cible pour les 500 voisins les plus proches en fonction de 4 variables du jeu de données arrive en tête. Les variables que nous avions détectées durant l’analyse exploratoire comme fortement corrélées à la cible se trouvent également dans la liste : les variables sources extérieures (EXT\_SOURCE\_2, 3, VAR), les variables âge (DAYS\_BIRTH), l’ancienneté (DAYS\_EMPLOYED) ou le ratio montant crédit/annuité (CREDIT\_ANNUITY\_RATIO).

CONCLUSION – LIMITES ET AMELIORATIONS

Notre étude portait sur un problème de classification binaire présentant un déséquilibre de classe.

Nous avons mis en œuvre des stratégies propres pour optimiser le meilleur modèle et obtenir une performance maximale. Nous avons testé et comparé différentes solutions de rééquilibrage de classe. Nous avons créé de nouvelles variables en veillant à ce qu’elles restent facilement explicables et nous conformer ainsi à la demande de notre client.

Nous avons également créé une métrique métier et fixé un seuil de solvabilité optimum. Le modèle final est un LightGBM optimisé sur la métrique ROC\_AUC.

Une amélioration du modèle serait envisageable en poursuivant l’optimisation des hyperparamètres et en modifiant, avec l’aide d’un expert métier, la métrique que nous avons créée : la fixation de la pénalisation des faux négatifs est arbitraire, nous pourrions la « doser » différemment. Cet expert pourrait également nous guider dans la création de variables pertinentes auxquelles nous n’avons pas pensé, malgré le soin apporté au feature engineering.

PRESENTATION DU DASHBOARD