Note Méthodologique

Cette note méthodologique revient sur les principales étapes nécessaires à la réalisation du modèle statistique final conçu pour l'entreprise Prêt à Dépenser. Le modèle sert à détecter les clients de l'entreprise susceptibles de ne pas rembourser leur crédit. Il renvoie un score qui donne la probabilité pour un client de ne pas rembourser son crédit.

Informations Générales

Nom : Modélisation du Défaut de Paiement Client

But : Évaluer la probabilité d'un défaut de paiement pour un client de Prêt à Dépenser

Contributeur : Eloi Le Quilleuc

Date : 24-11-2021

Description : Ce projet Data Science propose un outil d'aide à l'identification les clients susceptibles de ne pas rembourser leur crédit pour l'entreprise Prêt à Dépenser. Ce projet apporte également une application web d'interprétation et de calcul du score crédit.

Code source : https://github.com/EloiLQ/pretadepenser-score

Application web : https://share.streamlit.io/eloilq/webapp-banking/main/app.py

Jeu de Données

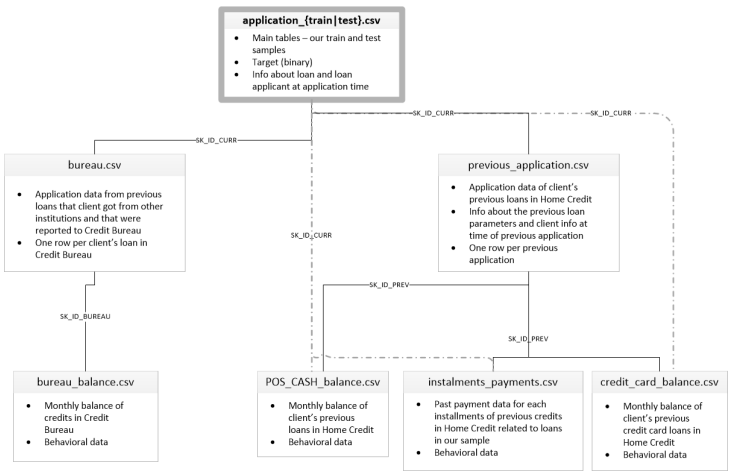
Chemin : https://www.kaggle.com/c/home-credit-default-risk/data

Origine : Home Credit

Description : informations bancaires des clients de Home Credit, réparties en sept tables.

Variable cible : TARGET

Description variable cible : variable binaire indiquant si le client a déjà eu un défaut de paiement ou non



Préparation des Données

Contributeur : Ekrem Bayar

Chemin : https://www.kaggle.com/ekrembayar/homecredit-default-risk-step-by-step-1st-notebook/notebook

Description : cette étape de préparation des données consiste principalement à agréger les sept tables tout en conservant un maximum d'information.

Filtre des clients : sélection des clients présent dans le fichier principal application_{train|test}.csv

Filtre des Variables : aucun

Valeurs manquantes : prises en compte

Encodage : One Hot pour les variables catégorielles

Feature Ingineering : moyennes de scores, moyennes, min, max sur des quantités temporelles, etc

Algorithme d'Apprentissage Machine

Algorithme : LightGBM

Home-page : https://github.com/microsoft/LightGBM

Version : 3.1.0

Type : Arbre de Décision Boosté

Tâche : Classification

Fonction de coût : régression logistique

Hyperparamètres : optimisés sur recherche sur grille (180 arbres de décision, 60 feuilles par arbre, taux d'apprentissage de 0.05, is_unbalance = True, autres hyperparamètres par défaut).

Données pour la Modélisation

Modélisation : sur validation croisée à partir d'un jeux d'entrainement

Nombre de strates de la validation croisée : 4

Mesure des performances : jeu de test

Utilisation du modèle : jeu d'évaluation

Taille du jeu d'entraînement : 285 979 clients

Taille du jeu de test : 21 526 clients

Nombre de variables d'entrée : 37 variables les plus importantes selon LightGBM

Variable cible : TARGET

Résultats

Sortie du modèle : score crédit ou probabilité de défaut de paiement

Score crédit médian : 5 %

Métrique d'évaluation : aire sous la courbe ROC (ROC AUC)

ROC AUC : 0.783

ROC AUC incertitude : [0.771 - 0.794] (intervalle de confiance 95 %), i.e 0.783 ± 1.5 %

Les clients aux scores crédit supérieurs à 13 % sont identifiés comme non payeurs. Ce seuil sur le score est estimé à partir d'une fonction de coût métier proportionnelle au solde de l'entreprise.

Coût Métier : - Np + alpha x Nnp

Np : nombre de clients payeurs
Nnp : nombre de clients non payeurs
alpha : perte associée à l'acceptance d'un non payeur divisée par le gain associé au rejet d'un client payeur (hypothèse alpha = 8)

On fait l'hypothèse d'un alpha égale à 8, c'est-à-dire qu'un client non payeur fait perdre 8 fois plus d'argent que n'en fait gagner un client payeur.

	precision	recall
0.0	0.96	0.82
1.0	0.22	0.58

classe 0 : clients payeurs
classe 1 : clients non payeurs

Si on considère que les clients identifiés comme 'non payeurs' sont rejetés, on obtient les chiffres suivants :

- 21.4 % : la proportion totale de clients rejetés
- 18 % (1 - recall classe 0) : la proportion de clients payeurs rejetés
- 58 % (recall classe 1) : la proportion de clients non payeurs rejetés

Interprétation

Algorithme : Shap

Home-page : http://github.com/slundberg/shap

Version : 0.39.0

Locale : coefficients de Shapley (en pourcentage) pour chaque variable et chaque client.

Globale : moyenne des coefficients de Shapley sur l'ensemble des clients.

Limites et Améliorations

Améliorations : marge d'amélioration possible à partir de feature ingineering.

Limites : la séparation des clients payeurs et non payeurs à partir des données bancaires reste une tâche difficile pour le modèle. Les probabilités de défaut de paiement des clients non payeurs sont principalement inférieures à 50%. Il est donc difficile d'identifier un client non payeur qu'identifier un client payeur. Une sélection des clients aux probabilité de défaut de paiement inférieurs à 2 % permet d'obtenir une bonne identification des clients payeurs.