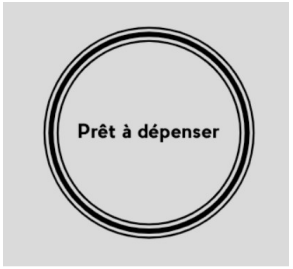
Implémentez un modèle de scoring



Meltem YILDIRIM

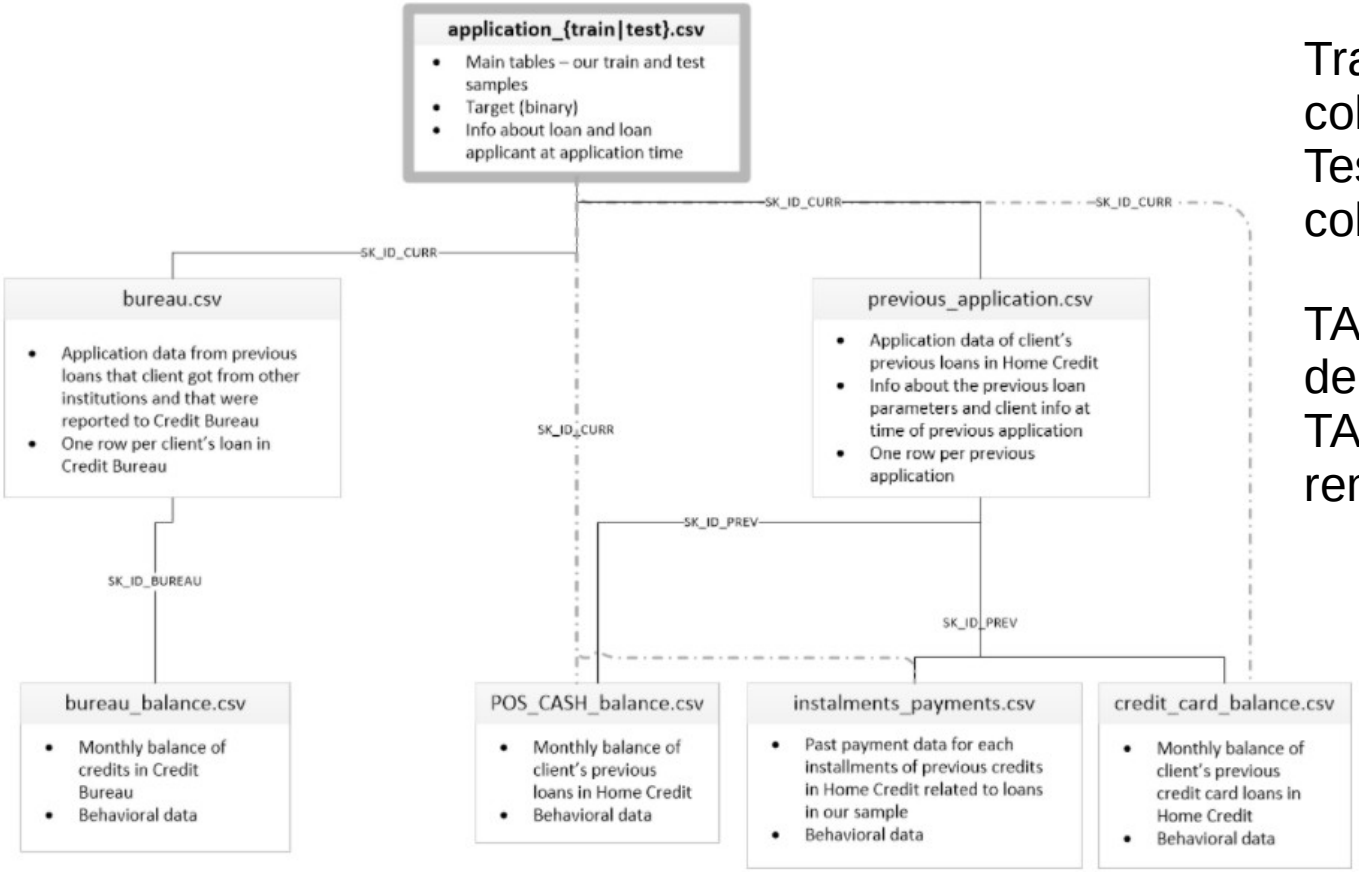
25/10/2021

Sommaire

I. Problématique

- **Contexte :**
  - **« Prêt à dépenser »** : société de crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt
- **Missions :**
  - Construction d'un modèle de scoring :
    - Traitement de données, algorithmes de classification binaire, évaluation des modèles
  - Construction d'un dashboard interactif :
    - API, visualisation du score, graphiques pour interpréter et comprendre les résultats

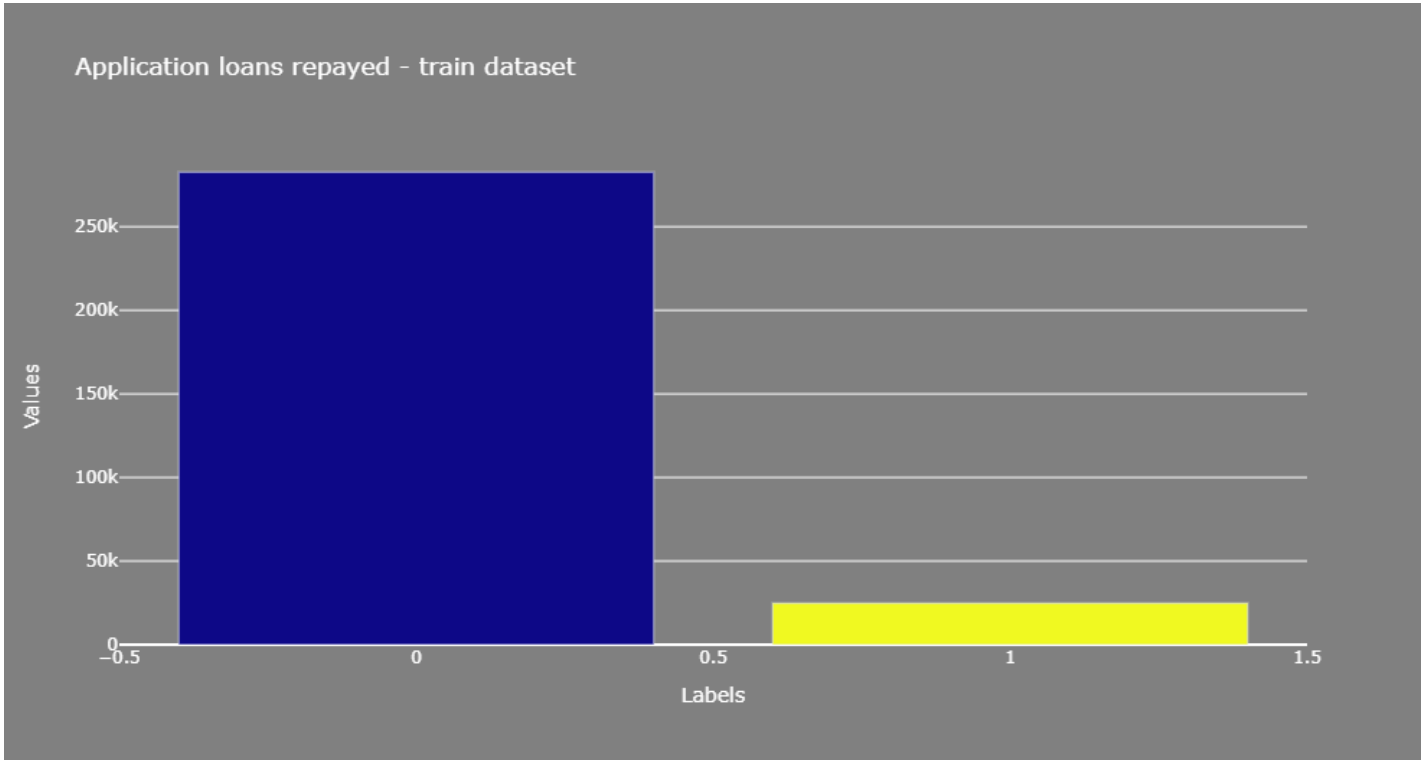
II. Présentation du jeu de données



Train : 307 511 lignes et 1 370 colonnes après la fusion.
Test : 48 744 lignes et 1369 colonnes après la fusion.

TARGET = 0 si pas de problème de remboursement
TARGET = 1 si problème de remboursement

II. Présentation du jeu de données



Problème de jeu de données déséquilibrées

III. Approche de la modélisation

- **Feature engineering :**
  - **application_{train|test} :**
    - Colonnes indiquant les valeurs manquantes s'il y a une différence dans le pourcentage de valeurs manquantes dans le cas TARGET = 0 et TARGET = 1
    - Polynomiale / plus technique (pourcentage du montant du crédit par rapport aux revenus du client, pourcentage des jours de travail par rapport à l'âge du client etc.)
  - **Tous les autres datasets :**
    - Somme, moyenne, min, max

III. Approche de la modélisation

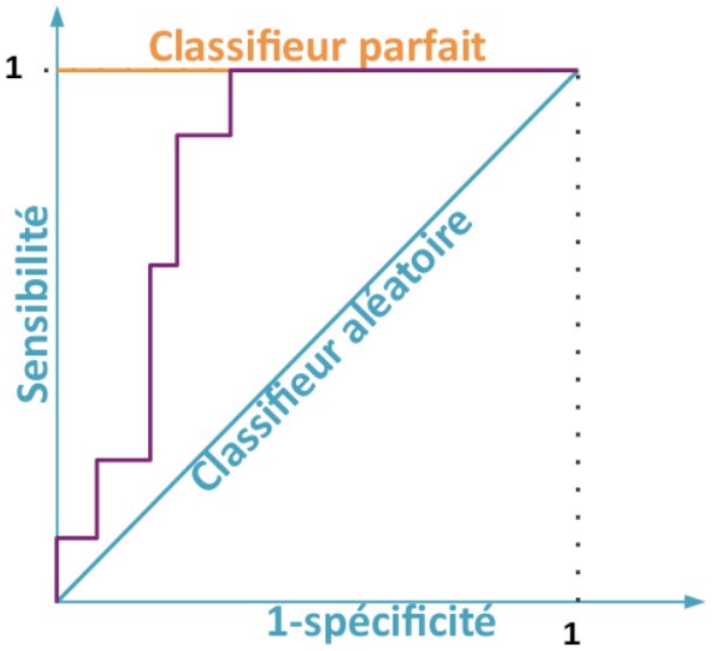
- **Prétraitement des données :**
  - Imputation des valeurs manquantes (numériques et catégorielles)
  - Ordinal Encoding et OneHotEncoding (catégorielles)
  - MinMaxScaler (numériques)
- **Données déséquilibrées :**
  - Under Sampling
  - Over Sampling
  - SMOTE

III. Approche de la modélisation

- **Différents algorithmes :**
  - Logistic Regression
  - Random Forest Classifier
  - **LGBM Classifier**
  - XGB Classifier
- **Séparation du jeu de données :**
  - 80 % train
  - 20 % test
- **Validation croisée et hyperparamètres :**
  - GridSearchCV
  - Fold = 3
  - n_estimators, num_leaves pour LGBM Classifier

III. Approche de la modélisation

- **Métrique d’évaluation :**
  - AUC, precision, recall et biais



IV. Résultats

Sans validation croisée

	AUC	AUC_Training	Precision	Recall
Logistic Regression - RUS	0.773394	0.778354	0.17248	0.707416
Logistic Regression - ROS	0.772485	0.776872	0.172354	0.701152
Logistic Regression - SMOTE	0.739308	0.782576	0.161791	0.646999
Random Forest Classifier - RUS	0.746818	1	0.161973	0.662356
Random Forest Classifier - ROS	0.738833	1	0.455696	0.0145484
Random Forest Classifier - SMOTE	0.599058	1	0.106948	0.495454
LGBM Classifier - RUS	0.779714	0.859721	0.175535	0.709234
LGBM Classifier - ROS	0.781999	0.832697	0.185091	0.692867
LGBM Classifier - SMOTE	0.694866	0.980817	0.13348	0.663973
XGB Classifier - RUS	0.764261	0.949709	0.167664	0.696706
XGB Classifier - ROS	0.765924	0.904821	0.197585	0.604971
XGB Classifier - SMOTE	0.634323	0.985832	0.0944157	0.874924

Avec validation croisée + RUS + LGBM Classifier

	AUC	AUC_Training	Precision	Recall
First	0.780591	0.854256	0.176963	0.709638
Second	0.780007	0.82656	0.176728	0.709234

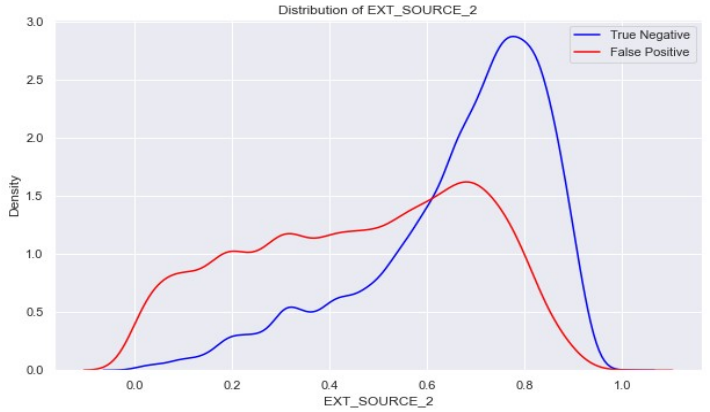
ROS + LGBM Classifier

	AUC	AUC_Training	Precision	Recall
First	0.782731	0.829656	0.185141	0.691857
Second	0.781137	0.811557	0.182729	0.699939

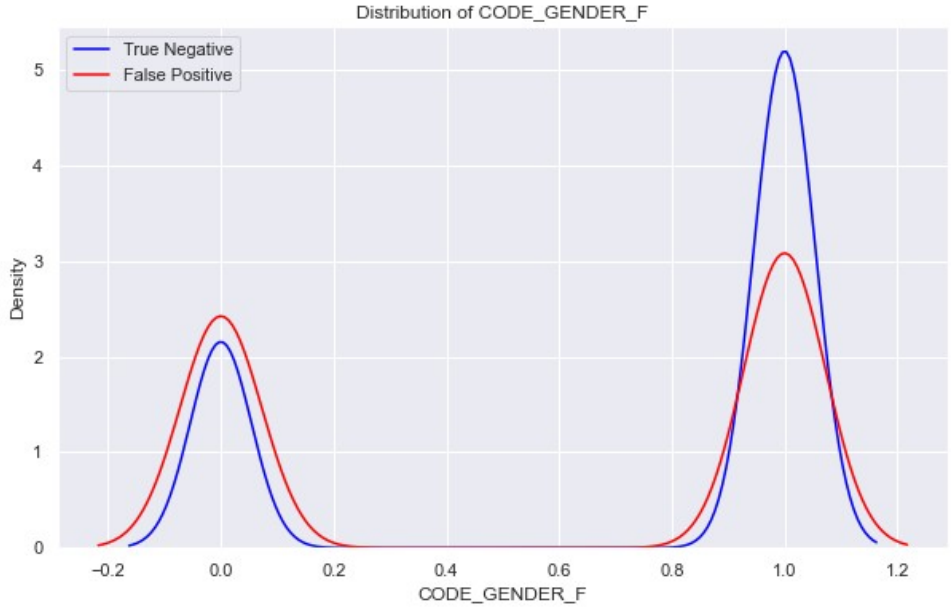
First :
- n_estimators = 150
- num_leaves = 20

Second :
- n_estimators = 200
- num_leaves = 10

IV. Résultats



Le biais du modèle



V. Présentation du dashboard

- API
- Dashboard