PHAM Jimmy CALVEZ Corentin TERRIEN Lubin

Data engineering and analysis

1)Démarche de préparation des données et de modélisation

Afin d'atteindre l'objectif de prédire le risque de défaut, nous avons organisés notre approche en deux étapes : la première étant de faite du Data engineering pour reconstruire l'historique client, puis faire une modélisation adaptée au classes et coûts financiers.

1.1) Data Engineering

Le Dataset "Home Credit Default Risk " était séparé en plusieurs tables, il a fallu les rassembler ensemble afin d'obtenir une profondeur historique de chaque client sans pour autant dupliquer les lignes.

- Traitement des anomalies : L'analyse exploratoire a révélé des valeurs aberrantes, notamment dans la variable DAYS_EMPLOYED où une valeur d'environ 1000ans apparaissait fréquemment. Ces valeurs ont été remplacées par des valeurs nulles pour ne pas fausser le modèle.
- Réduction du bruit : Afin d'optimiser les performances et de réduire la dimensionnalité inutile, nous avons supprimé toutes les variables contenant plus de 65% de valeurs manquantes.
- Nettoyage des catégories : Les entrées non informatives ont été filtrées.

1.2) Feature Engineering

Nous avons les données par client néanmoins nous avons perdu des informations après l'étape précédente, afin de compenser cela, nous avons créé de nouvelle feature tel que les identifiant de crédit ou des statistiques de moyenne, somme, max et min afin d'obtenir des indicateurs comportementaux. Cela nous permet d'obtenir au final un dataset qui possède une vue d'ensemble avec plus de 300 features prêt pour la modélisation.

2)Résultat

Une fois les données préparées, nous avons entraîné et comparé plusieurs algorithmes (Régression logistique, Random Forest, LightGBM et MLP) en utilisant une validation croisée stratifiée (StratifiedKFold) pour garantir la robustesse des résultats. Les performances ont été suivies via MLflow. Le modèle LightGBM a été sélectionné comme modèle final car il offrait la meilleure capacité de généralisation avec 0.70 pour le test AUC contrairement au Random Forest qui montrait des signes d'overfitting avec 1.0 pour AUC Train et 0.54 pour le test AUC.

3)Conclusion

A l'issue de la phase comparative, le modèle LightGBM a été retenu pour ses performances et sa stabilité.

- Robustesse : contrairement aux autres algorithmes testés (notamment le Random Forest qui fait de l'overfitting), le LightGBM démontre une excellente capacité de généralisation avec un AUC Test de 0.70 très proche de l'AUC Train de 0.74. Cela garantit que le modèle sera fiable sur de nouveaux clients.
- Performance Métier : Avec une Accuracy de 72%, le modèle parvient à classifier correctement une large majorité des dossiers.

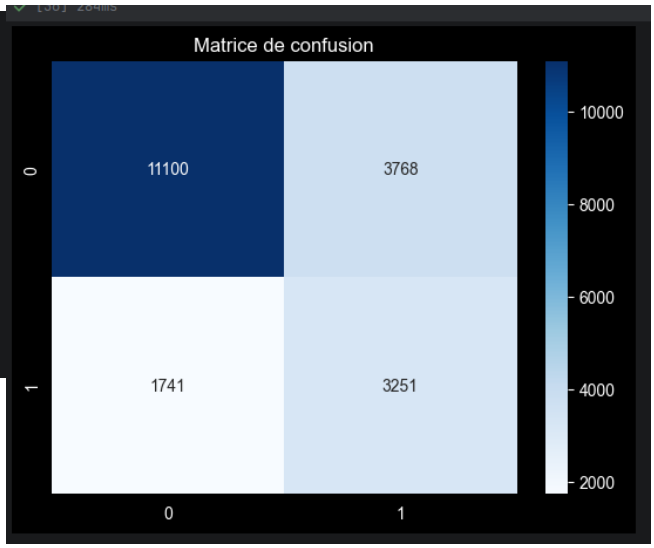
L'objectif prioritaire étant de minimiser le risque financier, l'analyse de la matrice de confusion révèle :

- Détection du risque : Le modèle parvient à identifier correctement 3251 clients à risque, évitant ainsi des pertes potentielles pour l'organisme de crédit.
- Couverture : Avec un Rappel de 0.65 sur la classe cible, nous captons 65% des défauts potentiels. Bien que 1741 dossiers à risque n'aient pas été détectés, ce résultat constitue un premier filtre efficace pour assister les organismes de crédit.

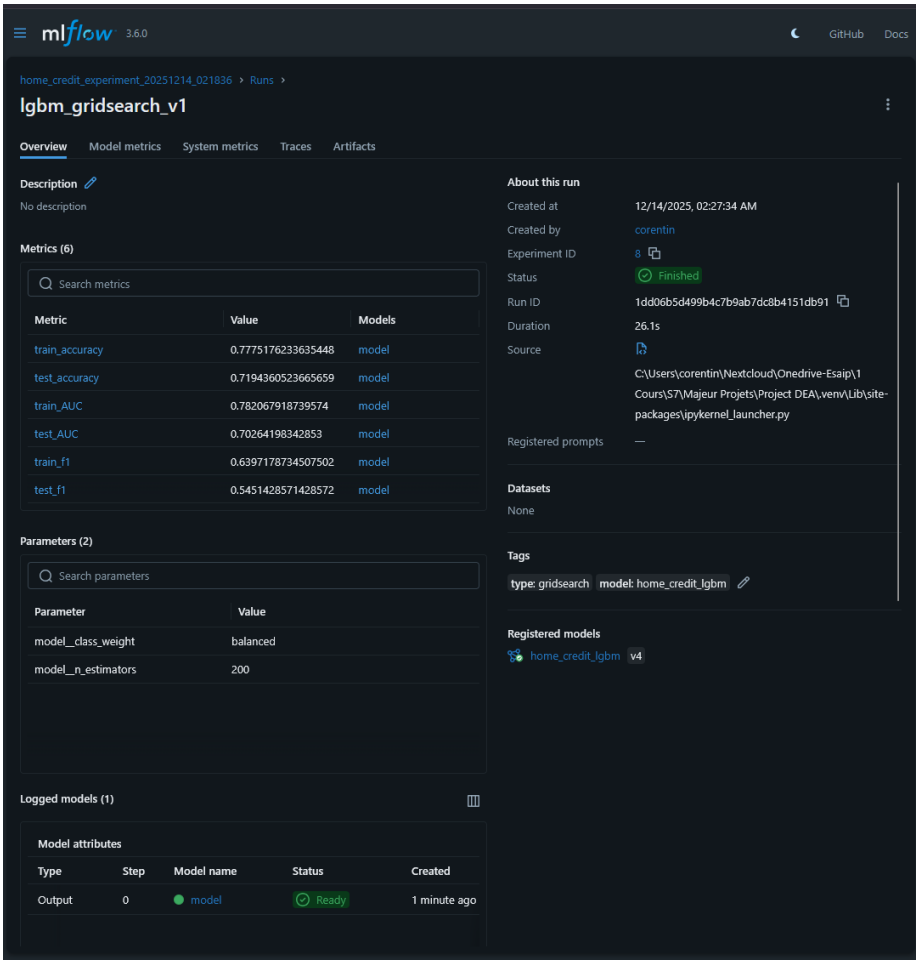
Néanmoins, même si ce modèle constitue une base solide, plusieurs pistes pourraient être explorées pour augmenter encore la détection des défauts :

- Optimisation du seuil : Affiner le seuil de décision pour capturer davantage de Faux Négatifs, quitte à augmenter légèrement le taux de refus.
- Feature Engineering : Intégrer de nouvelles variables ou des données comportementales plus fines pour aider le modèle à mieux discriminer les profils.

	precision	recall	f1-score	support
0	0.86	0.75	0.80	14868
1	0.46	0.65	0.54	4992
accuracy			0.72	19860
macro avg	0.66	0.70	0.67	19860
weighted avg	0.76	0.72	0.74	19860



4)Capture


mlflow 3.6.0
GitHub
Docs
home_credit_experiment_20251214_021836 › Runs ›
lgbm_gridsearch_v1
Overview
Model metrics
System metrics
Traces
Artifacts
Description
No description
Metrics (6)
Search metrics
Metric
Value
Models
train_accuracy 0.7775176233635448 model
test_accuracy 0.7194360523665659 model
train_AUC 0.782067918739574 model
test_AUC 0.70264198342853 model
train_f1 0.6397178734507502 model
test_f1 0.5451428571428572 model
Parameters (2)
Search parameters
Parameter
Value
model__class_weight balanced
model__n_estimators 200
Logged models (1)
Model attributes
Type
Step
Model name
Status
Created
Output 0 model Ready 1 minute ago
About this run
Created at 12/14/2025, 02:27:34 AM
Created by corentin
Experiment ID 8
Status Finished
Run ID 1dd06b5d499b4c7b9ab7dc8b4151db91
Duration 26.1s
Source
C:\Users\corentin\Nextcloud\Onedrive-Esaip\1 Cours\S7\Majeur Projets\Project DEA\.venv\Lib\site-packages\ipykernel_launcher.py
Registered prompts —
Datasets
None
Tags
type: gridsearch
model: home_credit_lgbm
Registered models
home_credit_lgbm v4