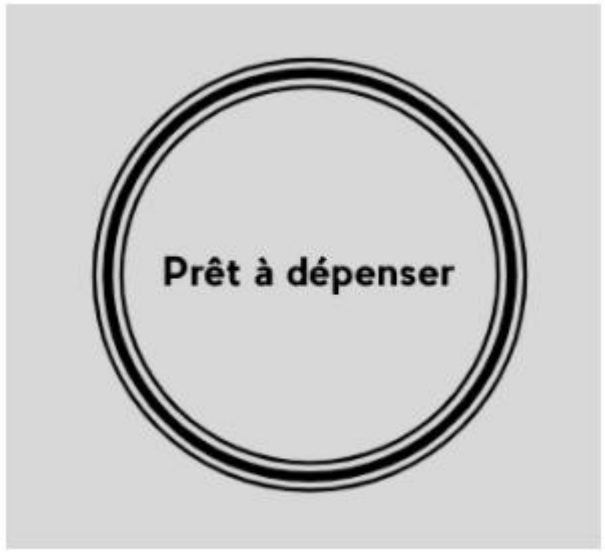
IMPLÉMENTEZ UN MODÈLE DE SCORING



Parcours Data sciences | ABBOUbD Marwa

Encadrant : Bertrand Beaufils

Présentation de la problématique

Data Scientist au sein d'une société financière nommée «prêt à dépenser», qui propose des crédits à la consommation pour des personnes avec peu d'historique de prêt

Objectifs de l'étude

- ❑ Construire un modèle de scoring de la probabilité de défaut de paiement d'un client afin de prendre la décision d'accorder ou non le prêt au client

- ❑ Deployer un dashboard : Construire un tableau de bord interactif et simple de compréhension et le mettre à la disposition du service client

Source de données

https://www.kaggle.com/moltean/fruits

Plan

1. Présentation de la problématique
2. Présentation de données
3. Prétraitement des Données - Notebook Kaggle
4. Modélisation : Machine Learning
5. Déploiement : API et Dashboard interactif
6. Conclusion et Perspectives

PRESENTATION DES DONNEES

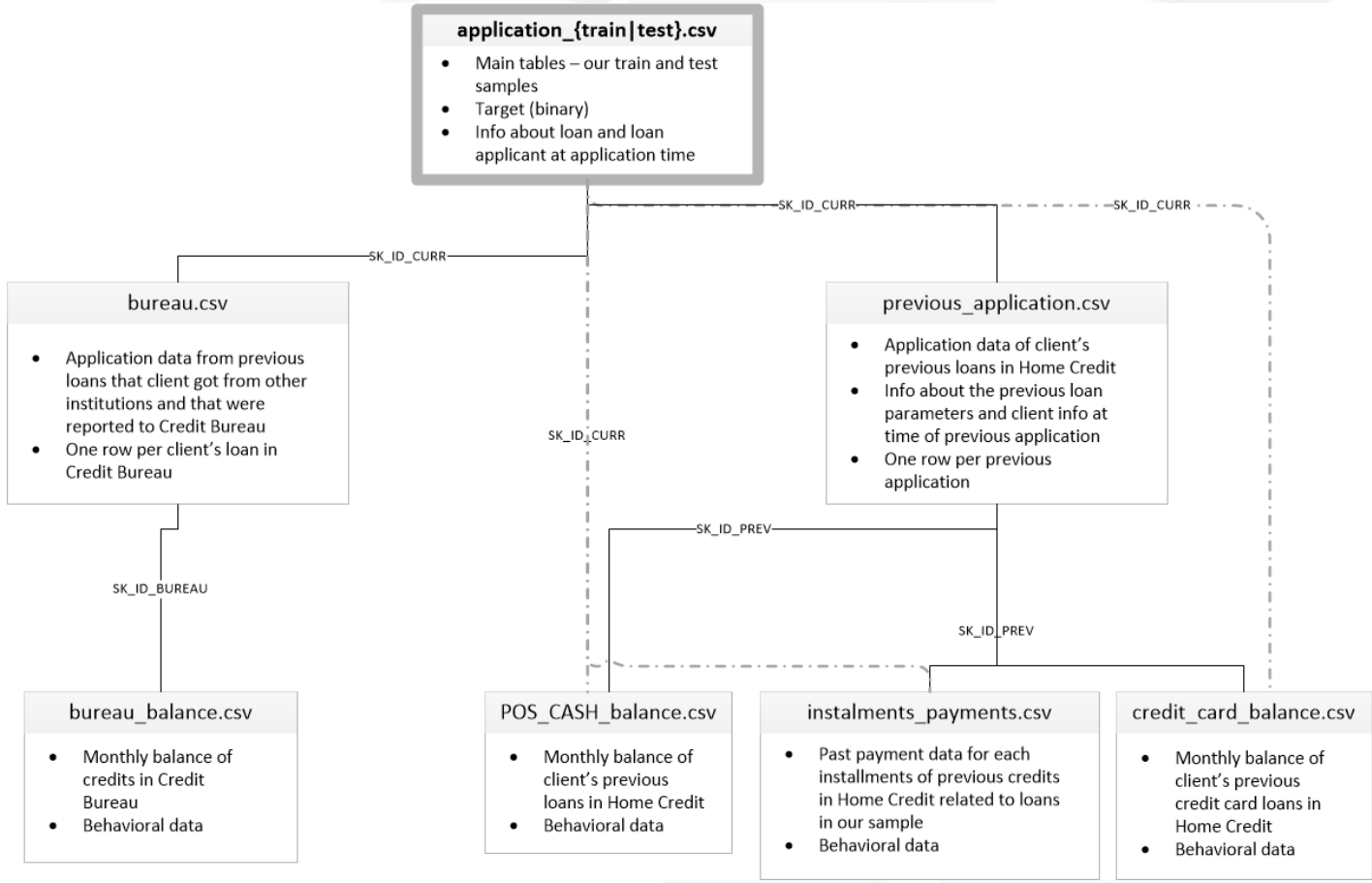
- Base de données composée de 7 sources différentes

  - Base de données principale :

120 Features : âge, sexe, nombre de jours travaillés, revenus, sources extérieures , informations relatives au crédit..

- Labels cible :

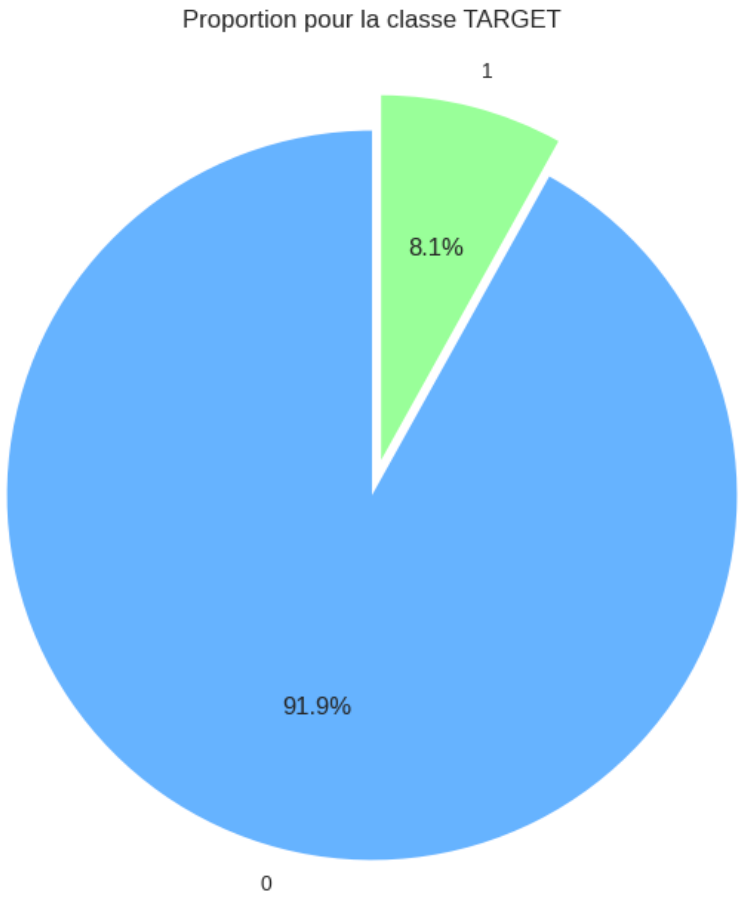
Défaut de crédit (1) **Pas de défaut de crédit (0)**



Présentation du Prétraitement des Données - Notebook Kaggle

Encodage variables	• Encodage variables: -Label Encoding pour les variables à 2 catégories. -One Hot Encoding pour les variables > 2 catégories.
Traitement d'outliers et des valeurs aberrantes	• Supprimer valeurs aberrantes : DAYS_EMPLOYED > 366 GENRES = XNA
Création de variables spécifiques à la problématique	• Ratio du montant du crédit par rapport au revenu du client • Ratio des annuités par rapport au revenu du client • Durée du crédit • Pourcentage de jours salariés par rapport à l'âge du client
Traitement des Valeurs manquantes	• Imputation par la médiane pour les valeurs manquantes
Résultats du Prétraitement	• Jeu de données final avec 244 variables

Données déséquilibrées



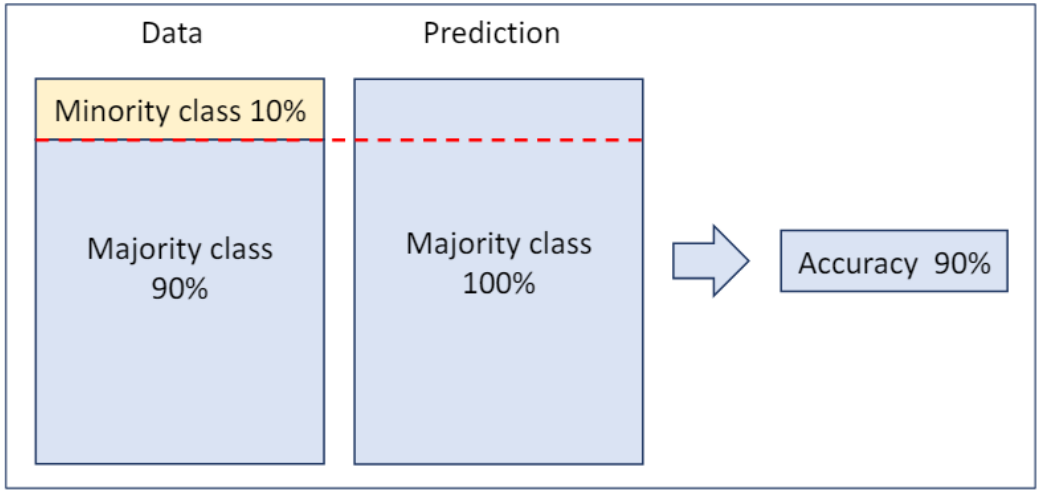
Problématique principale des données

Très large déséquilibre entre les classes

- 91.9 % des clients réguliers
- 8.1 % des clients avec des défauts de paiement

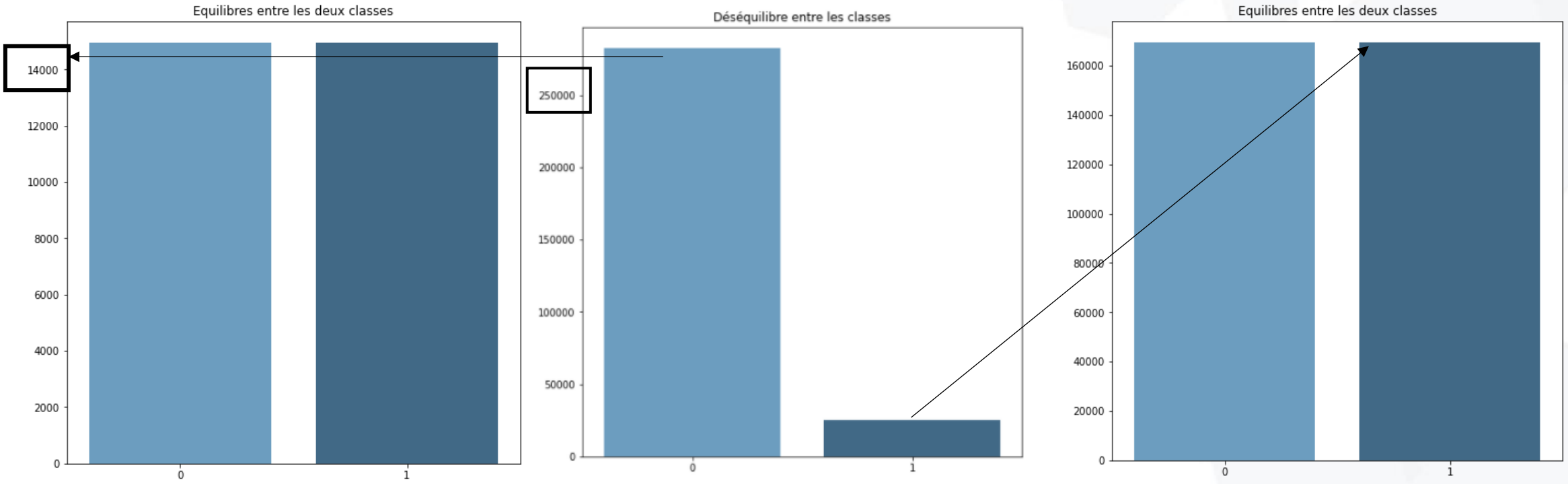
Problématique pour l'apprentissage du modèle

Risque de prédire uniquement la classe Majoritaire



Méthodes de gestion de données déséquilibre

Utilisation du **Undersampling**

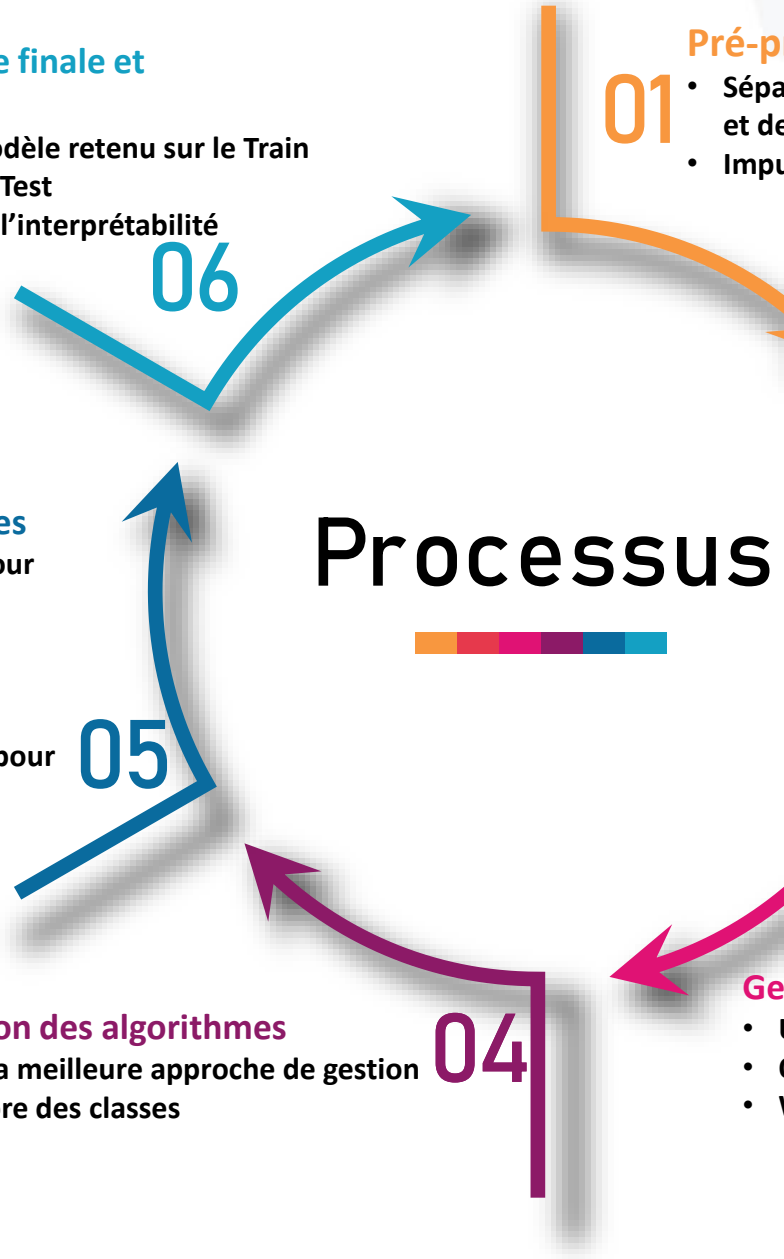


Utilisation du **Oversampling**

Utilisation de **class_weight**

Gestion du déséquilibre pendant l'entraînement du modèle en association un poids différent à chaque classe

Étapes du Processus de Modélisation



01 Pré-processing

- **Séparation en jeu d'entraînement, validation et de test ;**
- **Imputation et Standardisation.**

02 Mise en œuvre des algorithmes

- **Modele de classification : Random Forest RF et Light Gradien Boosting LGBM**
- **Choix des métriques:**
  - ROC-AUC
  - F1 score
  - Matrice de confusion

03 Gestion du déséquilibre des classes

- **Undersampling**
- **Oversampling**
- **Weight-balanced (balanced)**

04 Comparaison des algorithmes

- **Choix de la meilleure approche de gestion déséquilibre des classes**

05 Optimisation des hyperparamètres

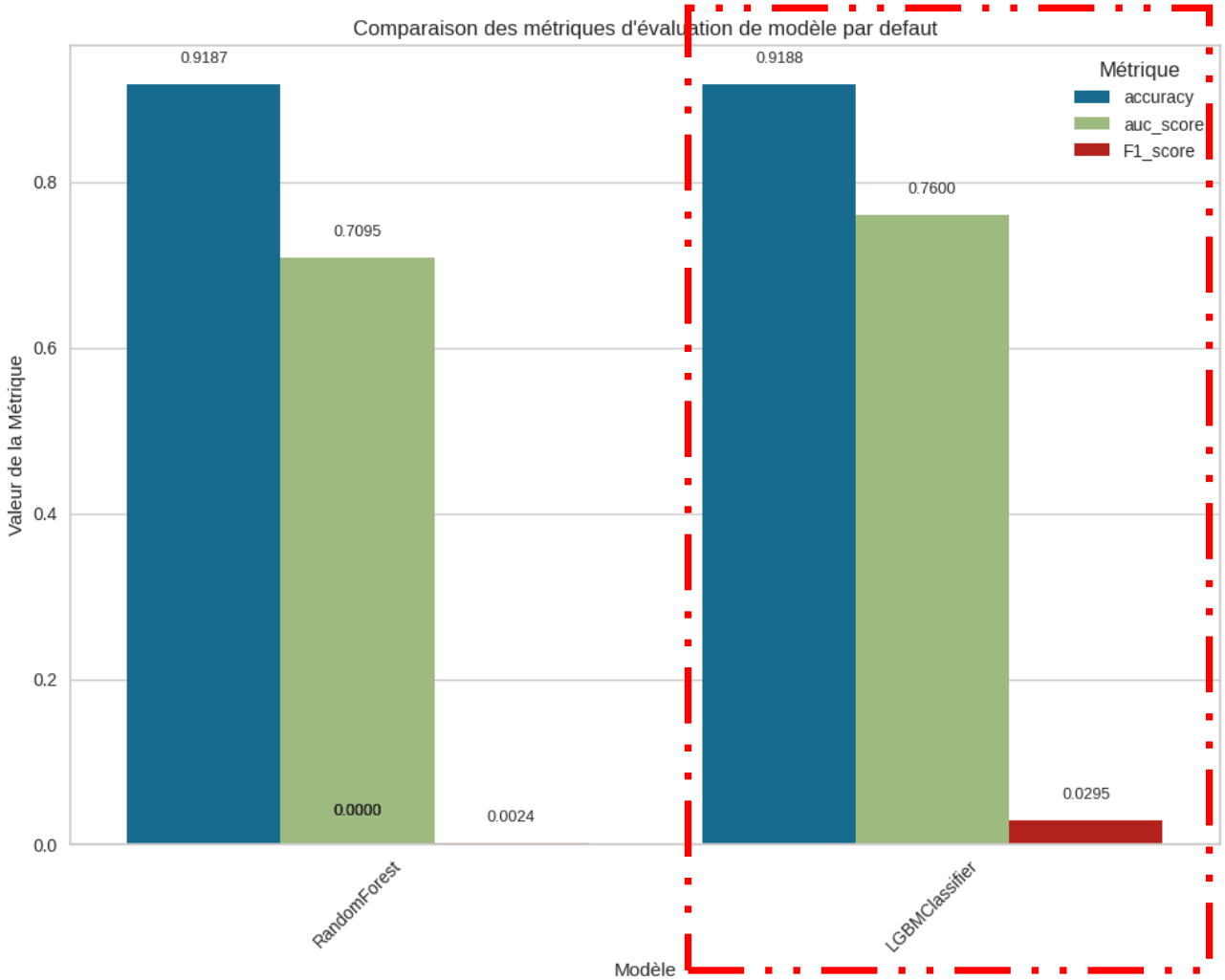
- **Définition d'une métrique métier pour optimiser les résultats du modèle:** Custom Score (fonction de cout)
- **Optimisation des hyperparamètres pour chaque modèle**

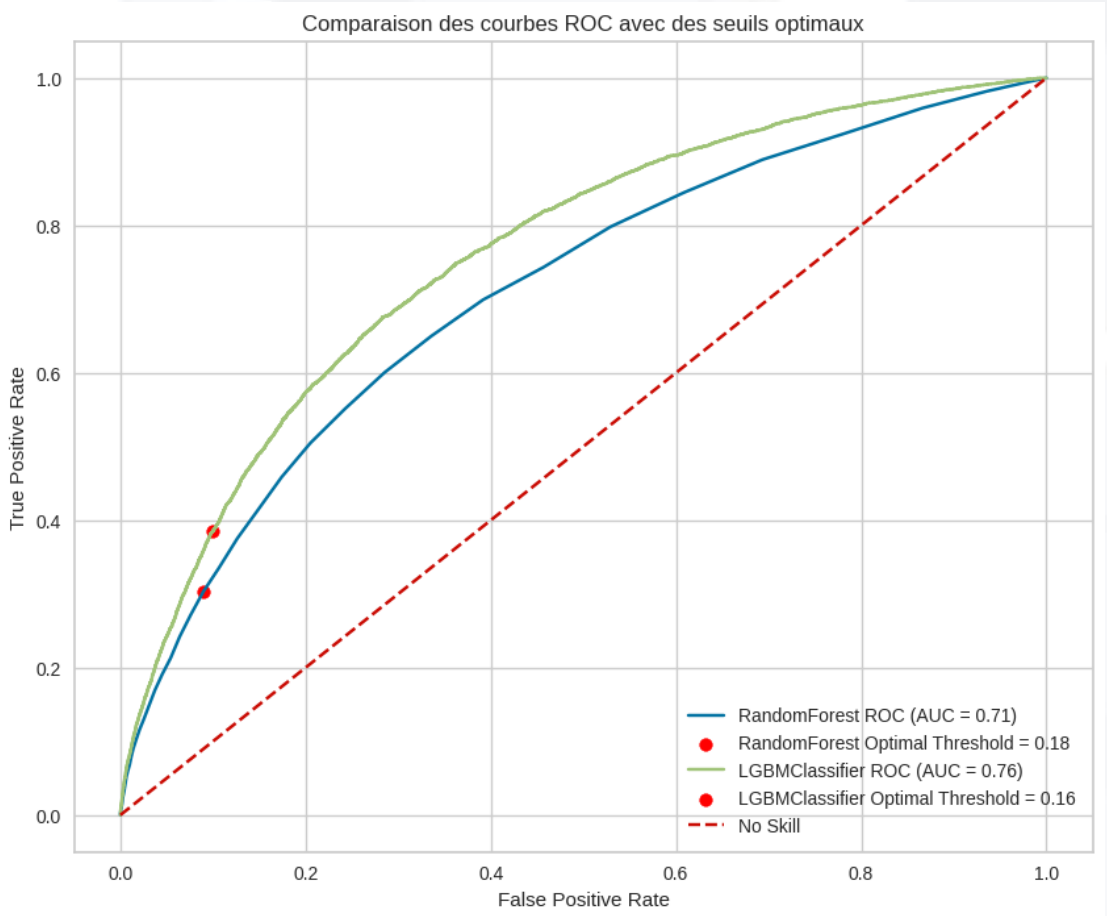
06 Sélection du modèle finale et Interprétabilité

- **Entrainement du modèle retenu sur le Train et Prédictions sur le Test**
- **Méthode shap pour l'interprétabilité**

Comparaison des Performances des Modèles par Défaut

- **LightGBM** a le meilleur performance en ce qui concerne le AUC et F1

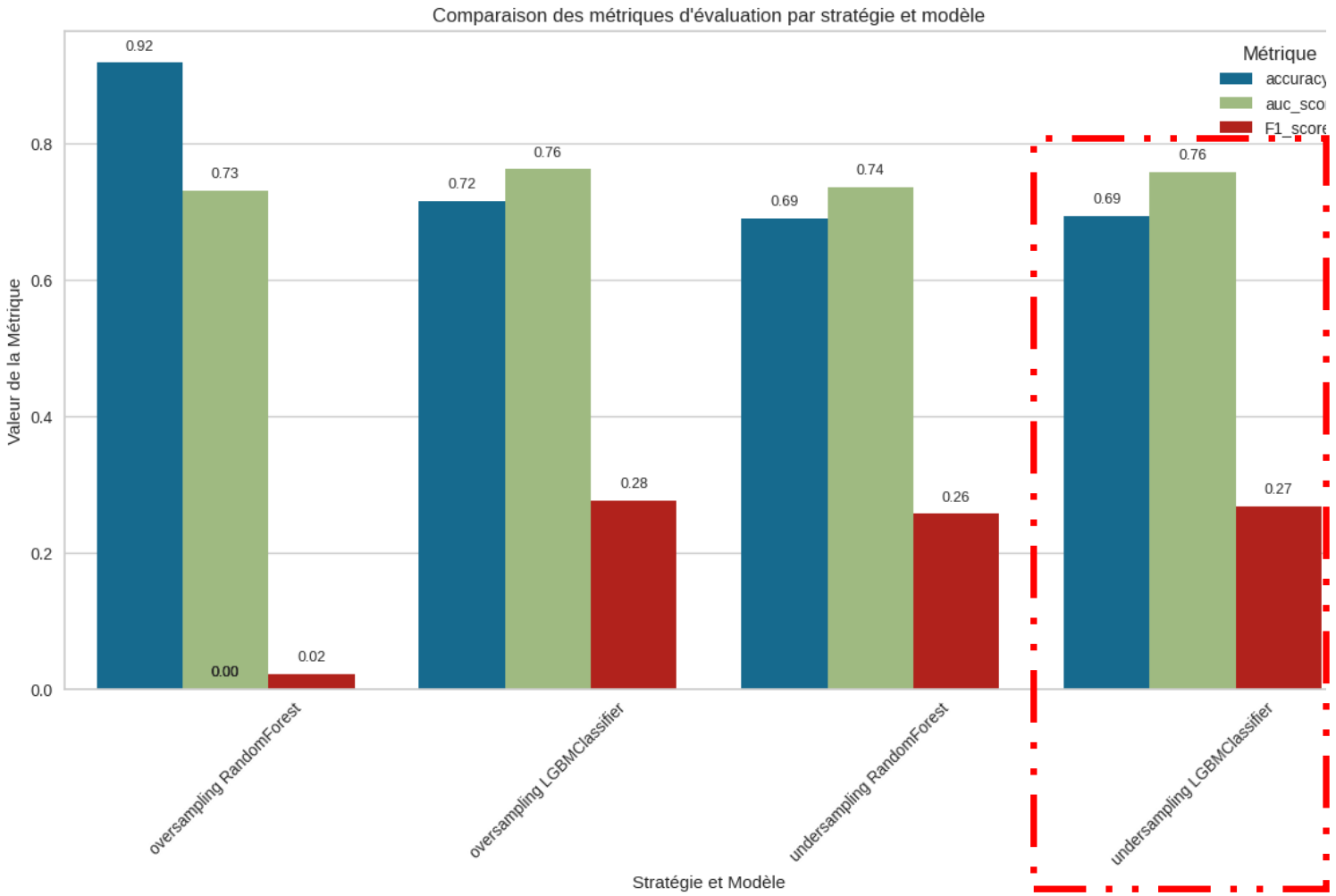


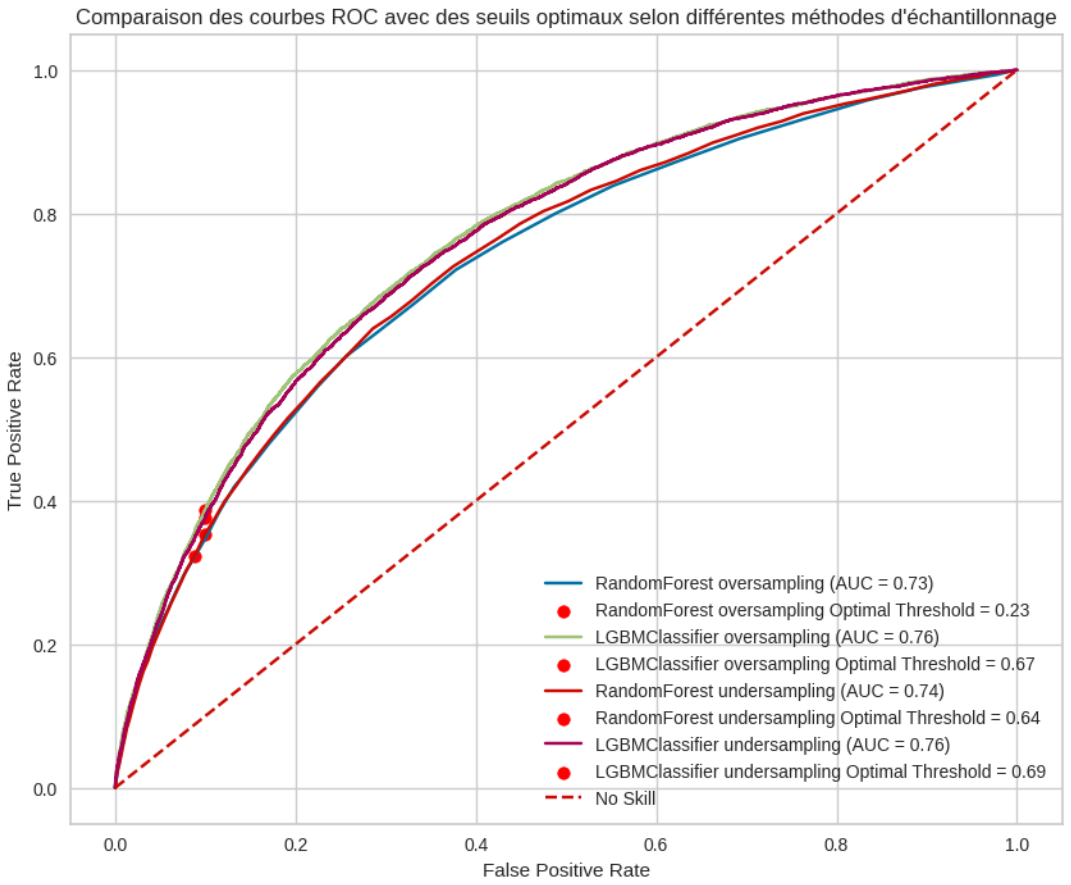


- **LightGBM** surpasse en AUC, distinguant mieux les clients à risque comparé à Random Forest et au modèle No Skill

Impact des Stratégies d'Échantillonnage sur la Performance des Modèles

- **LightGBM avec undersampling** présente les scores les plus élevés en AUC et F1.

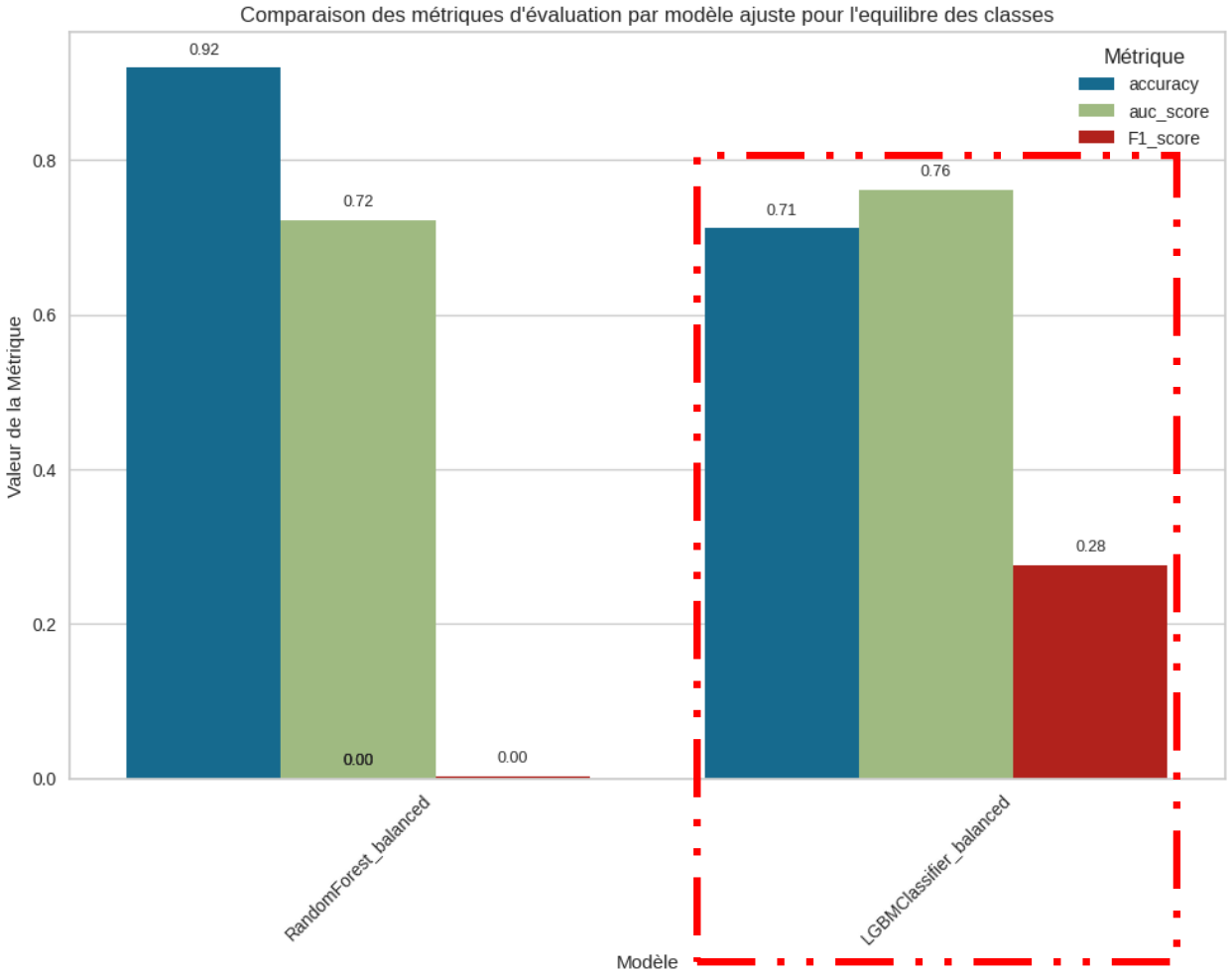


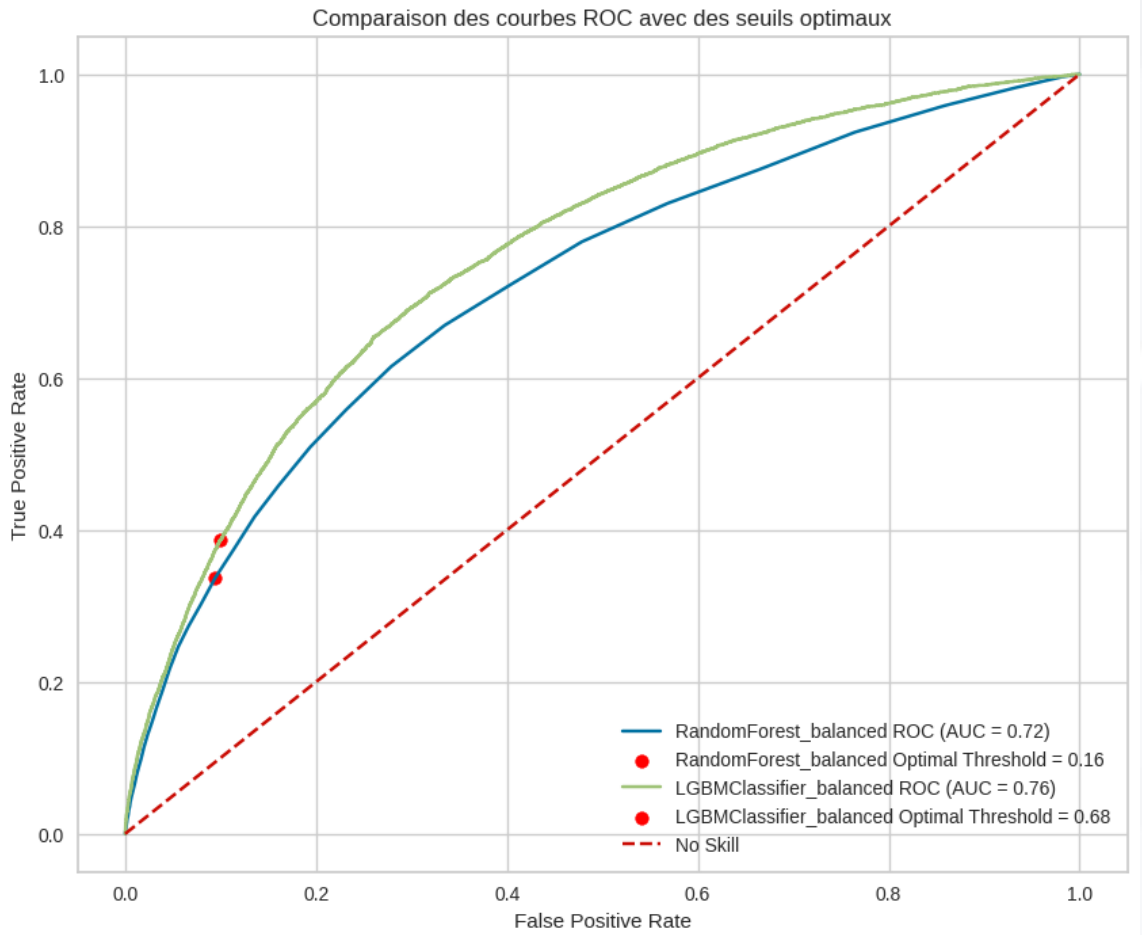


- **LightGBM avec undersampling** atteint la meilleure séparation de classes, indiquée par l'AUC la plus élevée, surtout à des seuils supérieurs."

Impact de l'Application de class_weight='balanced' sur les Performances des Modèles

- LightGBM affiche une amélioration notable en F1, indiquant une meilleure gestion des classes minoritaires





- LightGBM atteint une AUC plus élevée, indiquant une meilleure distinction entre les classes.

- RF bien que moins performant en AUC, montre néanmoins une amélioration notable par rapport à un modèle sans rééquilibrage des poids

DÉFINITION D'UN SCORING ADAPTÉ À LA PROBLÉMATIQUE MÉTIER

- True Negatives (TN) : Prêts accordés avec succès, léger bénéfice
- False Negatives (FN) : Prêts non remboursés, grande perte
- True Positives (TP): Risques bien identifiés, bénéfice marginal
- False Positives (FP): Opportunités manquées, perte faible

	Predicted: 0	**Predicted: 1**
Actual: 0	True Negatives **(TN)**	False Positives **(FP)**
Actual: 1	False Negatives **(FN)**	True Positives **(TP)**

-10 **+1** +2 **-1**

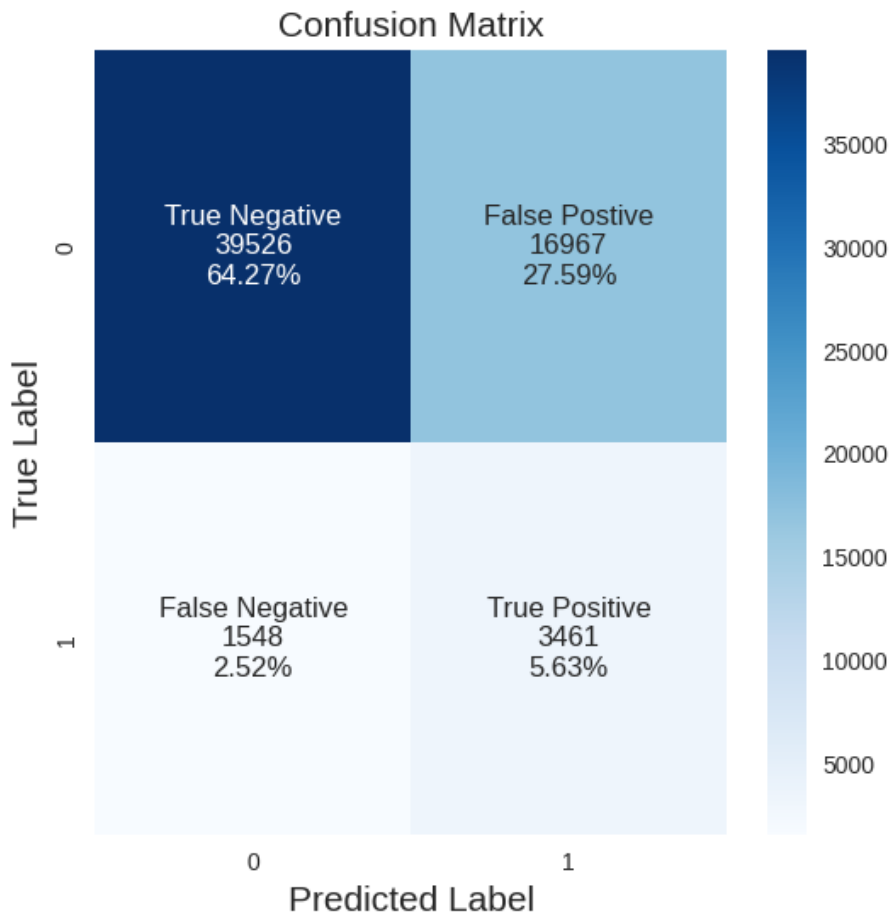
Score = (1 x TN) - (1 x FP) - (10 x FN) + (2 x TP)

- Score.Max pour prédiction parfaite $= \sum(\mathrm{TN} \times 1 + \mathrm{TP} \times 2)$
- Score.Min pour pire scénario $= \sum(\mathrm{FN} \times -10 + \mathrm{FP} \times -1)$

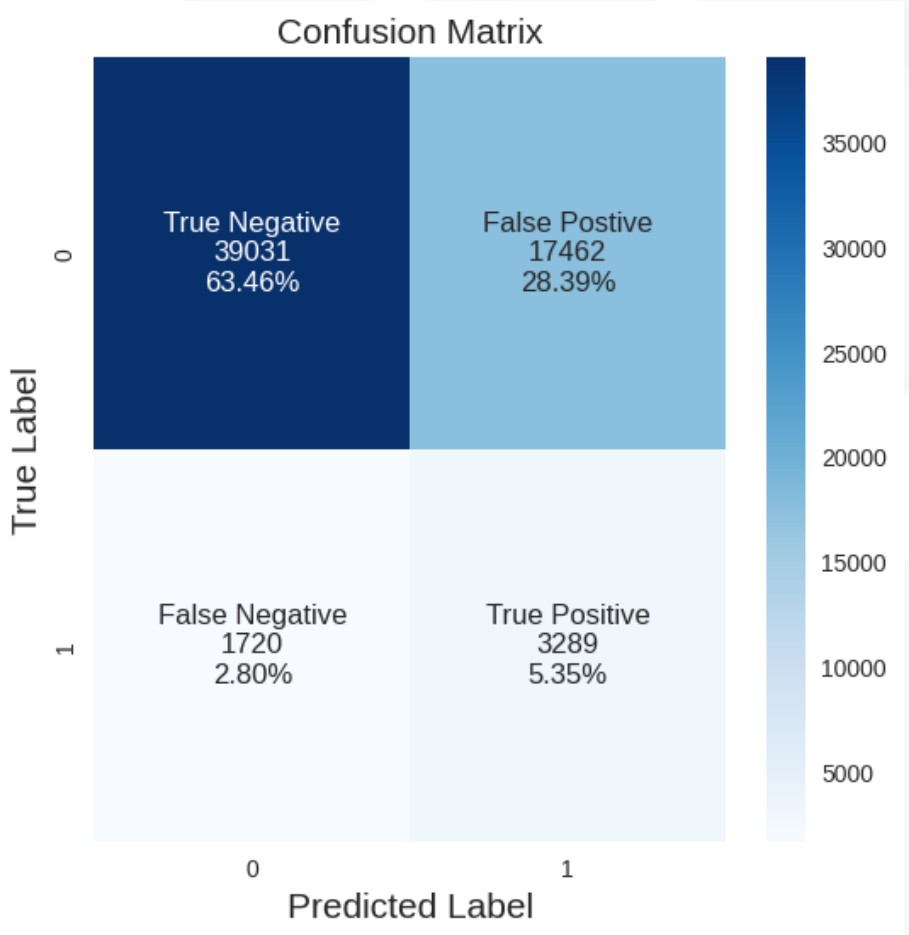
$$\text{Score normalise} = \frac{Score - \text{Score.Min}}{\text{Score.max} - Score.min} \quad \epsilon\,[0, 1]$$

Comparaison des Modèles Optimisés

LGBM Optimise

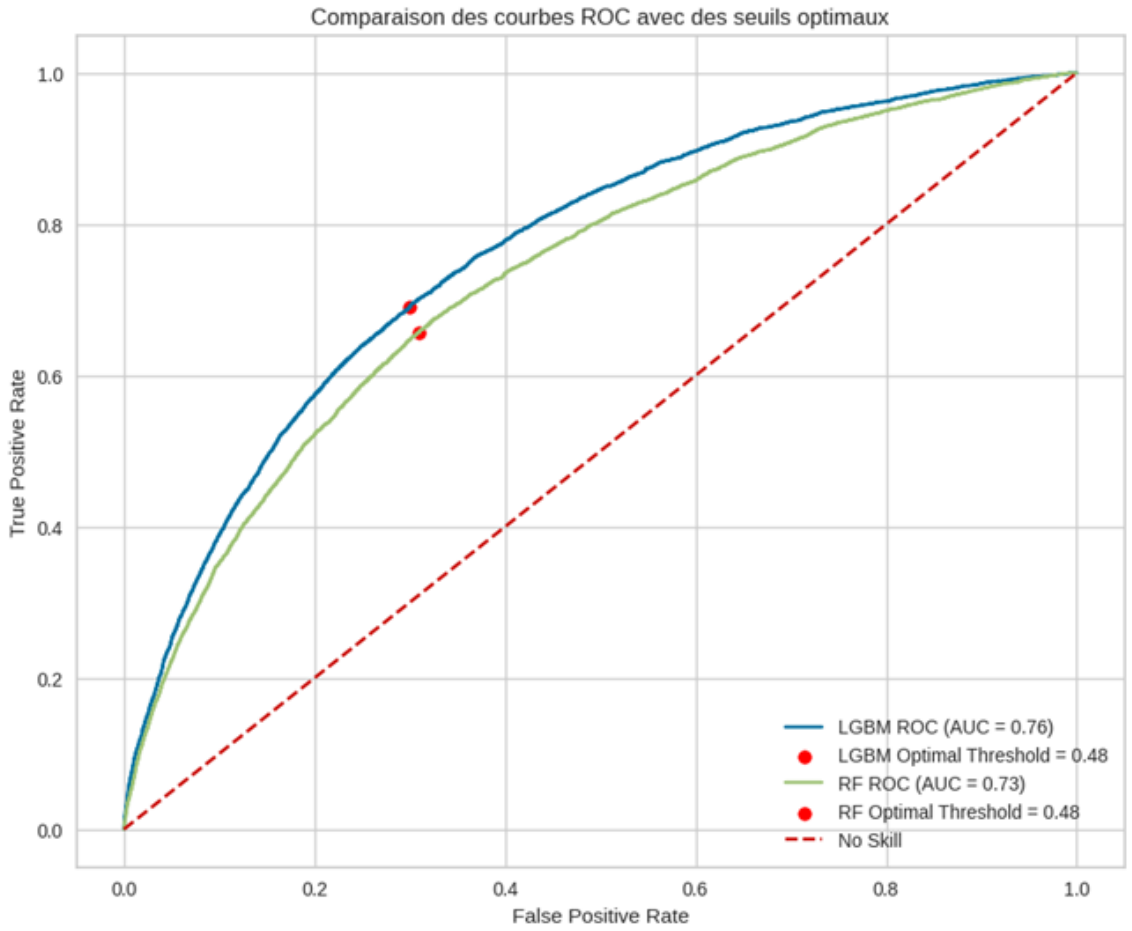


RF Optimise



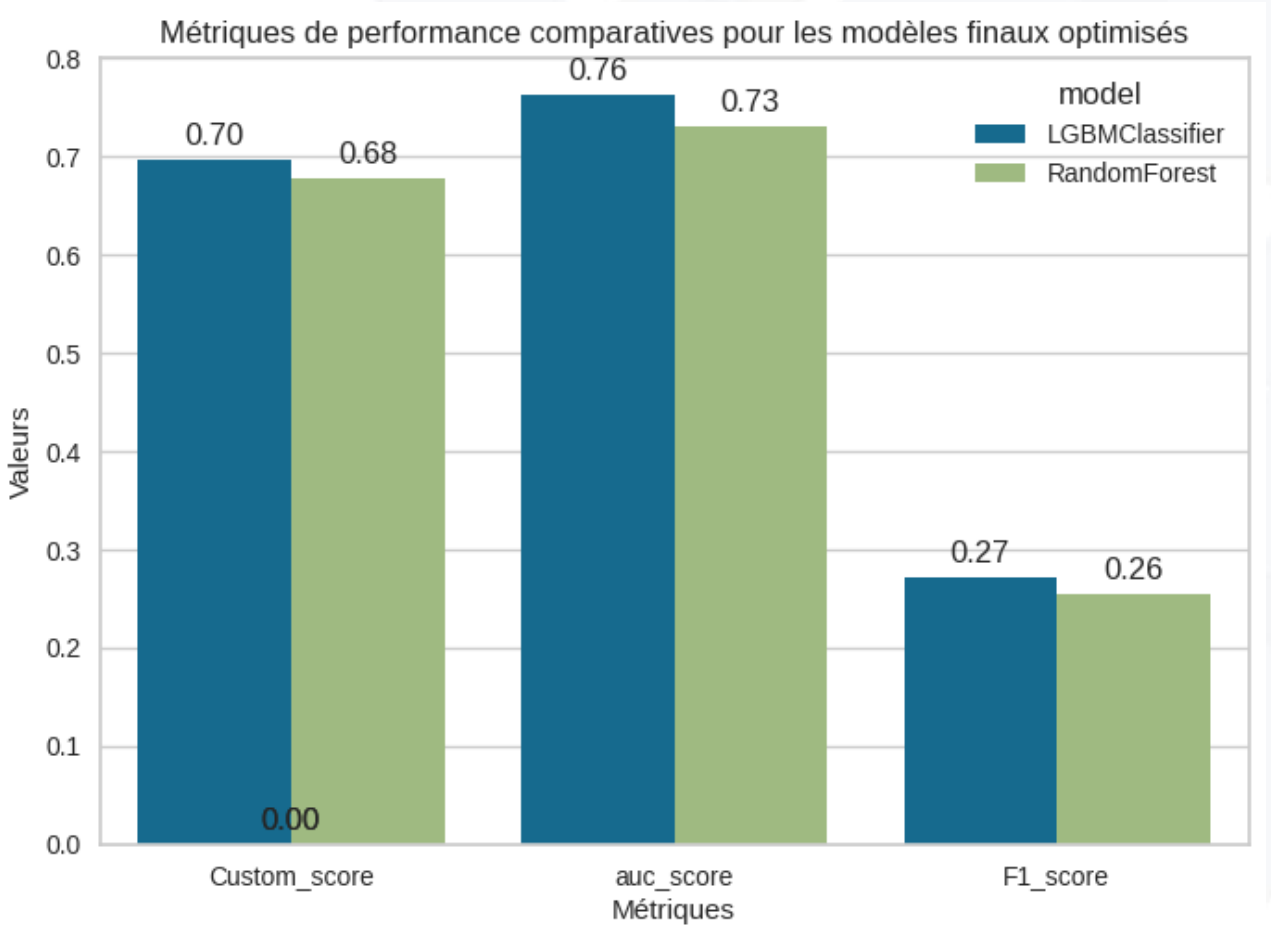
- LightGBM montre une meilleure précision avec un taux de vrais positifs plus élevé que le modèle RF
- RF présente une légère augmentation du nombre de faux positifs mais reste robuste

Comparaison des Modèles Optimisés



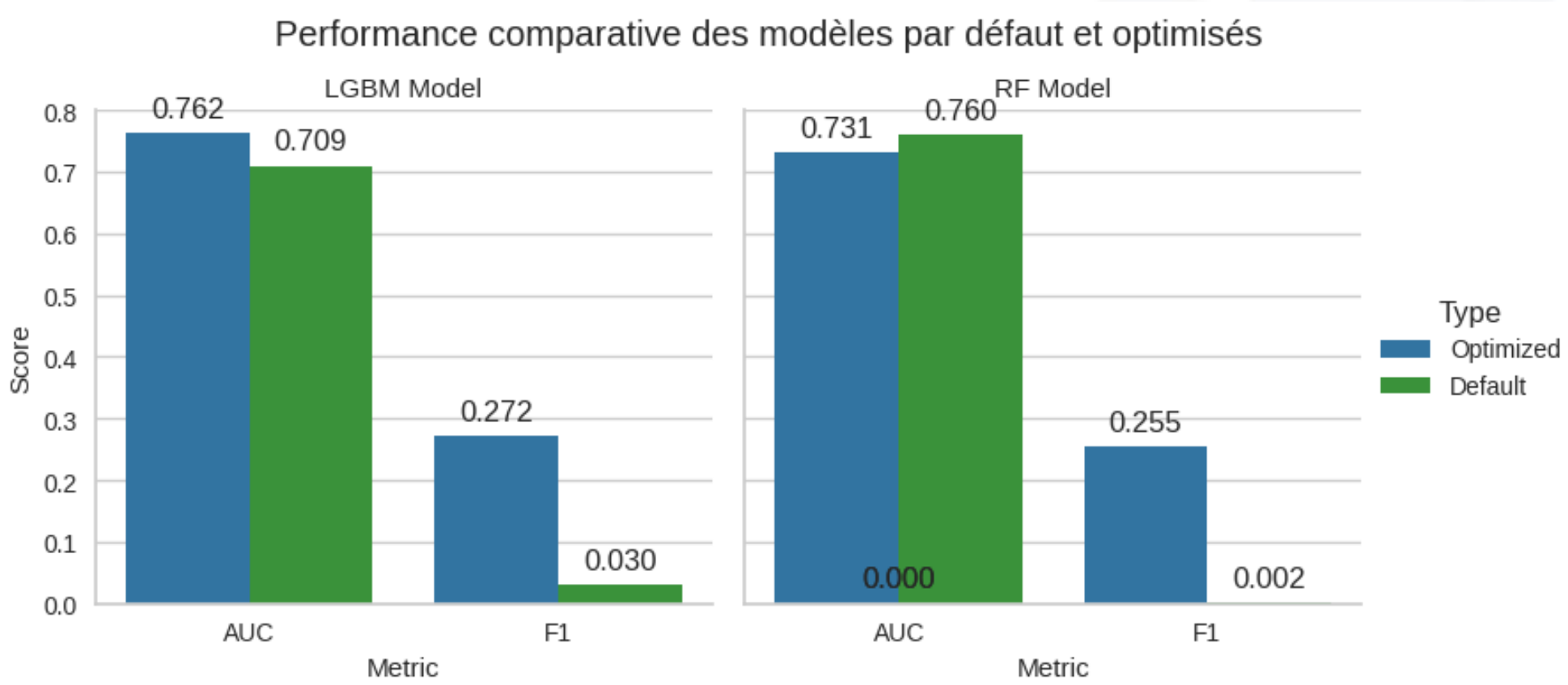
- Les courbes ROC montrent que le modèle LGBM a une meilleure performance avec une plus grande surface sous la courbe (AUC)

- Les scores AUC et F1 sont plus élevés pour LGBM, indiquant une meilleure performance globale



- LGBM a été sélectionné comme modèle final en raison de ses performances supérieures selon les trois métriques évaluées.

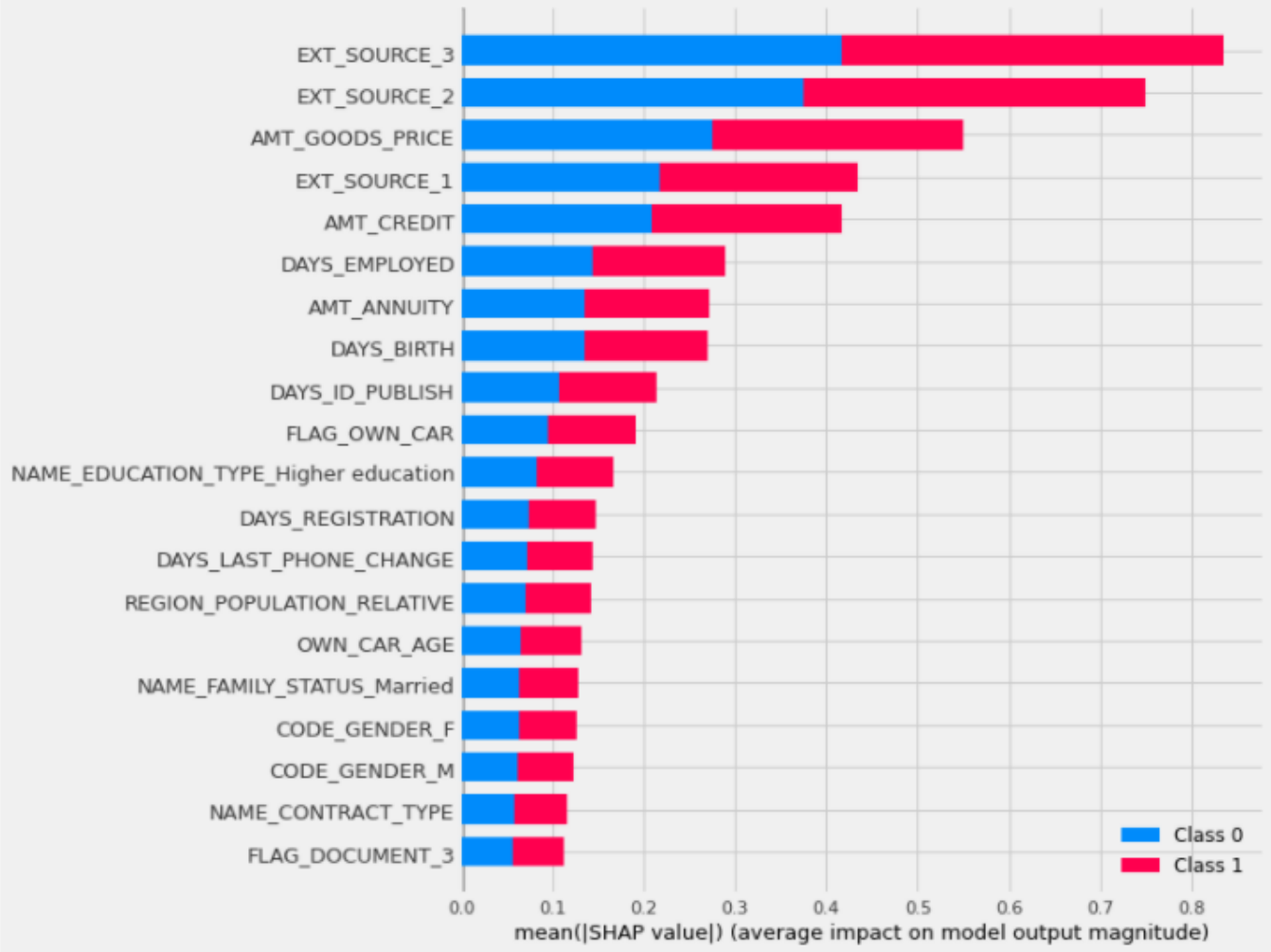
Comparaison des Performances des Modèles Optimisés et par Défaut



- La gestion de l'équilibre par la méthode « **Class weight = Balanced** » et l'optimisation des hyperparamètres ont permis un gain de7.48 % pour le modèle LGBM et de 3.97% pour le modèle RF

Analyse de l'Interprétabilité du Modèle via SHAP

shap.summary_plot → Importances des Features en global sur la prédiction

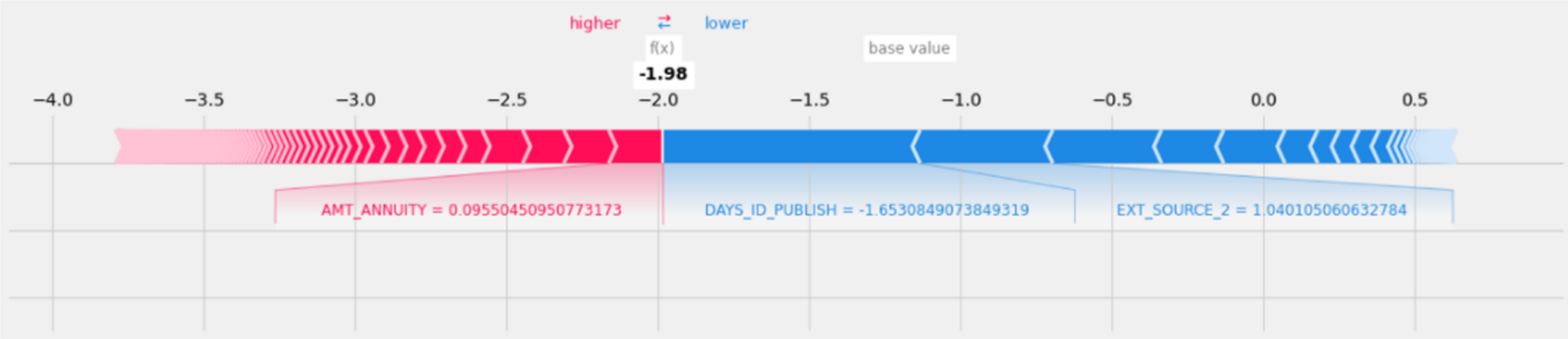


L'axe horizontal représente l'impact moyen de chaque variable sur les prédictions du modèle

- Les sources externes, telles que EXT_SOURCE_3 et EXT_SOURCE_2, jouent un rôle crucial dans les prédictions du modèle

Interprétabilité du Modèle avec SHAP - Analyse d'une Prédiction Particulière

shape.force_plot → Importance des Caractéristiques pour une Prédiction Spécifique



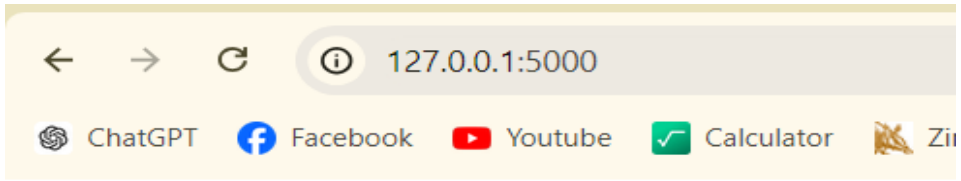
Création d'une API via FLASK

- Flask : API permettant une prédiction à partir de l'ID du client

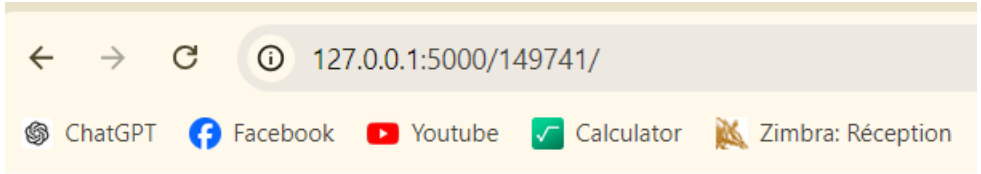
```
abbou@Marwa-PC MINGW64 /c/Users/hojei/env (master)
$ python api3.py
C:\Users\hojei\env\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator LabelEncoder from versi
 Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app 'api3'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
```

LGBMClassifier
au format PICKLE

URL LOCALE



Entrer une ID client dans la barre URL



Ce client est non solvable avec un taux de risque de 83.88%

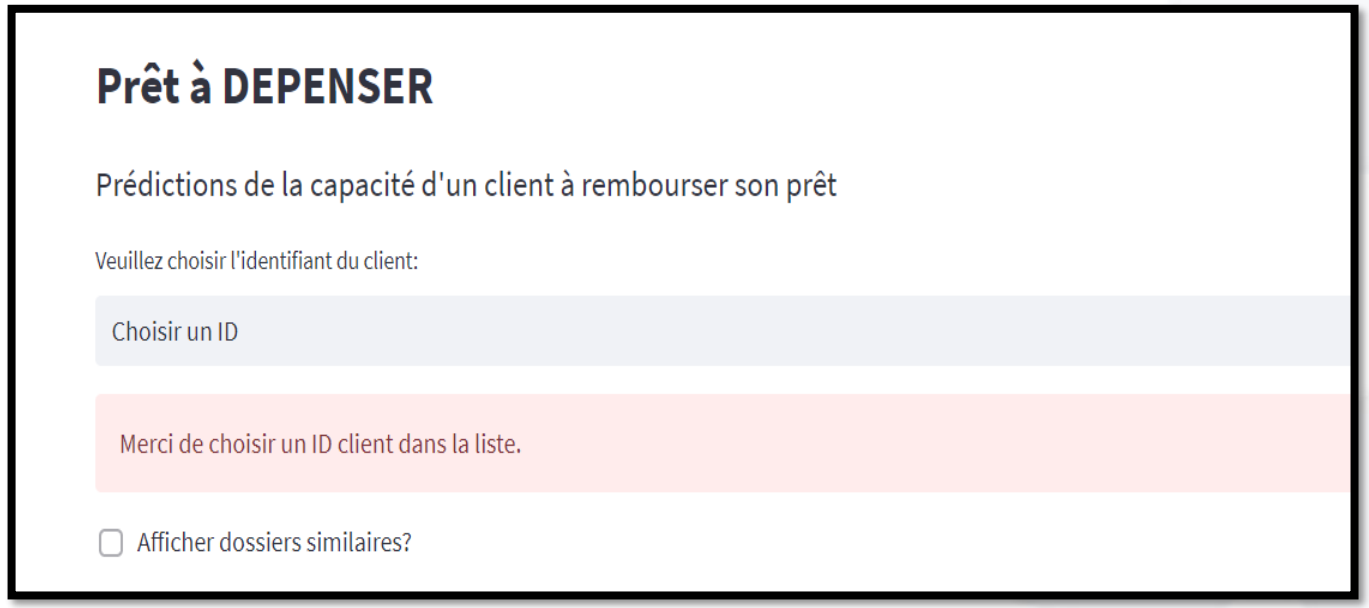
Creation du dashboard et hebergement sur Heroku

Le dashboard doit permettre de :

1. Visualiser le score pour chaque client
2. Visualiser des informations descriptives relatives à un client
3. Comparer les informations descriptives relatives à un client à l'ensemble des clients similaires

Présentation du Dashboard

Streamlit



- Déploiement du dashboard en passant par git et Heroku

- Lien vers le Dashboard https://limitless-escarpment-04117-235f6d042a6d.herokuapp.com/

CONCLUSION

- Utilisation d'un notebook issu de Kaggle
- Une population fortement asymétrique (92% - 8%)
- Evaluation du modèles avec une métrique métier
- Sélectionner la bonne approche pour le déséquilibre
- Amélioration des hyperparamètres pour le modelé LGBMClassifier
- Interprétation grâce à la bibliothèque SHAP
- Création d'une API web avec Flask.
- Mise en place d'un Dashboard interactif grâce à Streamlit

PERSPECTIVES

- Amélioration du preprocessing : Supprimer la plupart des variables présentant plus de 60% de valeurs manquantes
- Tester plus de modèles : - Decision Tree
  - XGBClassifier
  - CatBoost

- Optimiser d'autres hyperparamètres

PROFIL GitHub

- L'ensemble des fichiers de ce projet ont été stockés sur mon compte GitHub :

https://github.com/ABOUD43/OC-Projet6-Implementez-modele-scoring

Merci pour votre attention

Résultats de l'Optimisation des Hyperparamètres pour LightGBM et RandomForest

Modèle	Hyperparamètre	Valeur
LGBMClassifier	class_weight	balanced
	learning_rate	0.0112
	n_estimators	740.63
	num_leaves	55
	Best Threshold	0.4788
	AUC Score	0.7518
	F1 Score	0.2671
RandomForest	class_weight	balanced
	criterion	entropy
	max_depth	11
	n_estimators	660.20
	Best Threshold	0.4761
	AUC Score	0.7228
	F1 Score	0.2511