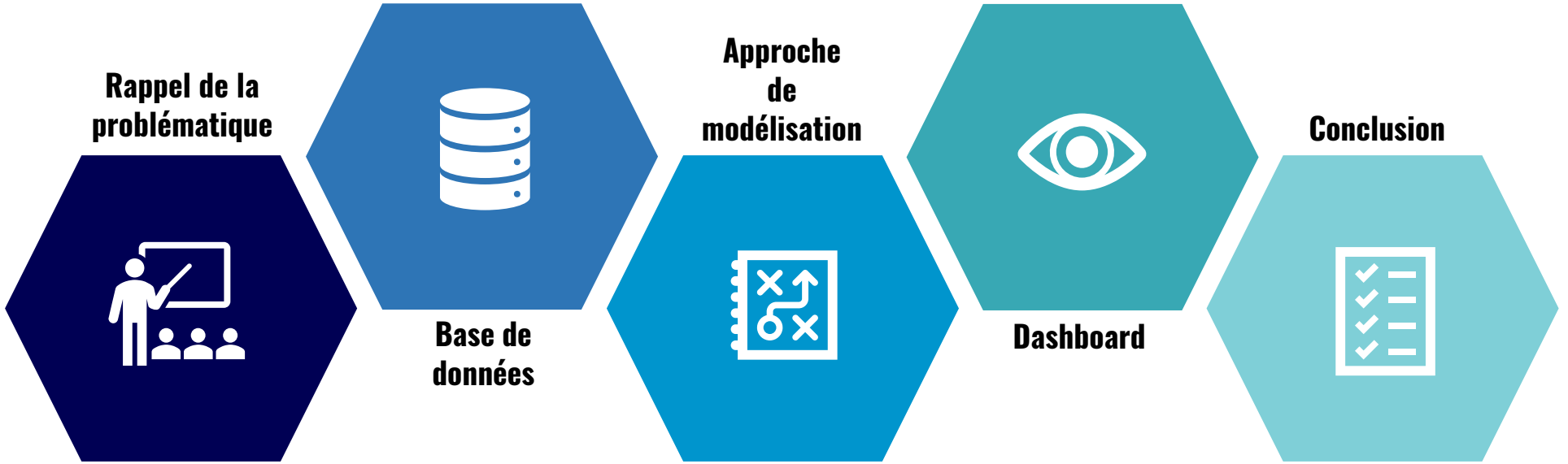
Parcours Data Scientist

Projet N°7 : Implémentez un modèle de scoring

Daniel CHASTANET

Sommaire


Rappel de la problématique
Base de données
Approche de modélisation
Dashboard
Conclusion

Rappel de la problématique

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

1. Mettre en œuvre un outil de « scoring crédit »
2. Classifier la demande en crédit accordé ou refusé

à partir de :

→ **données comportementales**
→ **données provenant d'autres institutions financières,**
→ **etc**

L'intention de ce projet est de réaliser un projet de bout en bout jusqu'à sa mise en production.

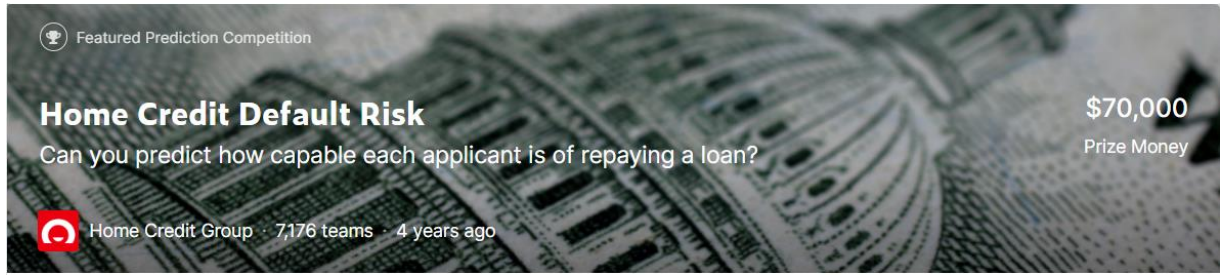
Points importants :

→ Transparence de la décision de crédit
→ Dashboard interactif avec informations personnelles du client

Partie I – Base de données et Analyse

Image : facilogi.com

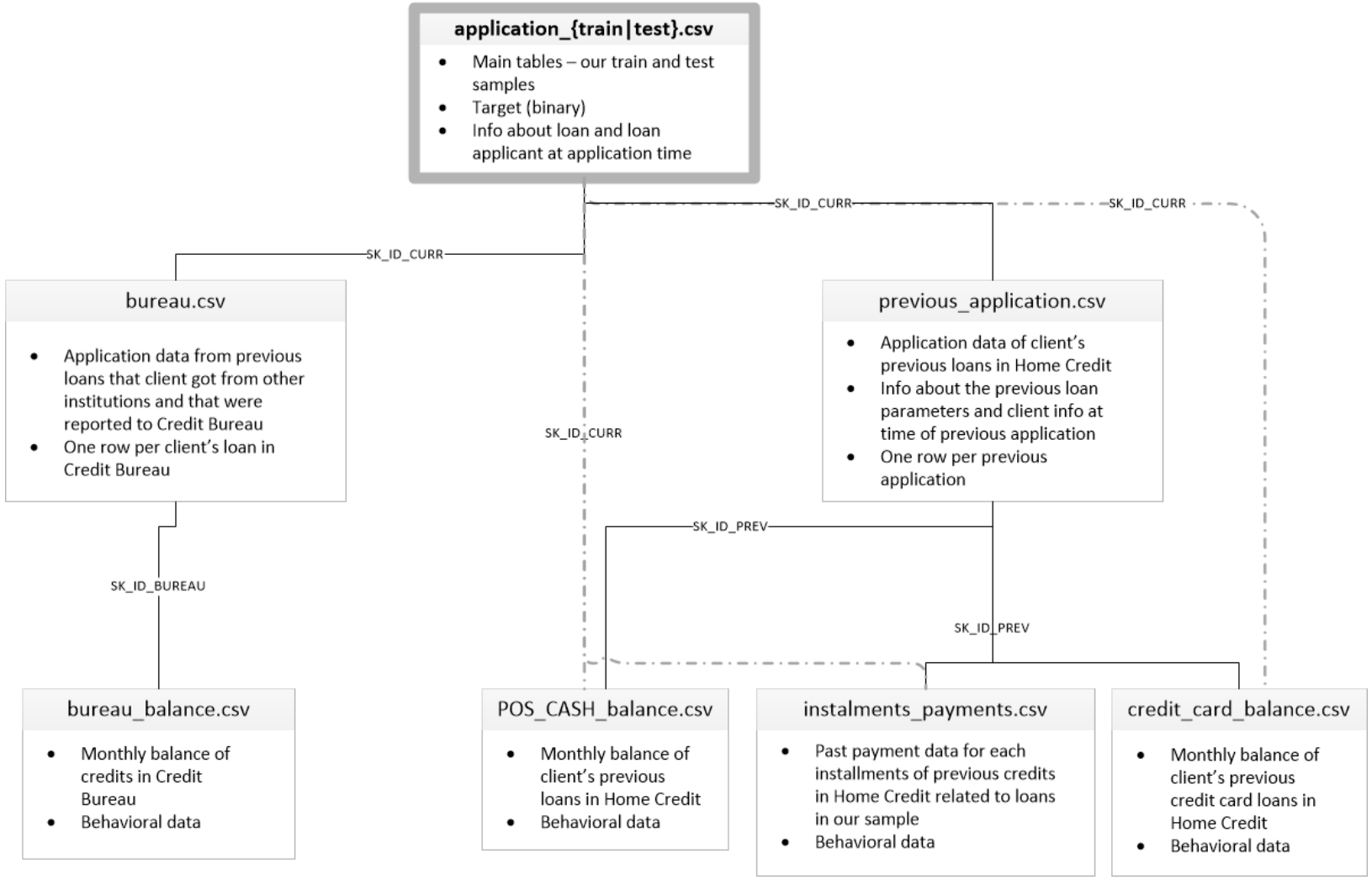
I – Les données



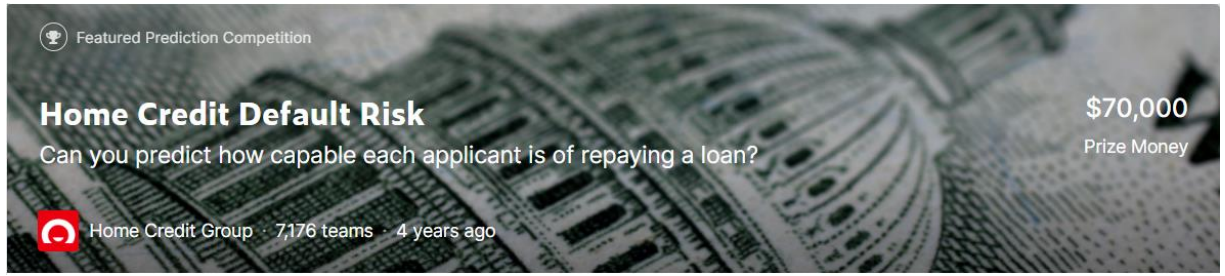
Home Credit Default Risk

7 bases de données

- application_train/test
- bureau
- bureau_balance
- Previous application
- POS_CASH_balance
- Instalments_payments
- credit_card_balance



I – Les données



Home Credit Default Risk

Focus sur application_train
+ (HomeCredit_columns_descriptions)

Information personnelles :
- Profession,
- Famille,
- Possessions immobilières
- etc

application_{train|test}.csv

- Main tables – our train and test samples
- Target (binary)
- Info about loan and loan applicant at application time

	attributs	valeurs_manquantes	type	Com
76	COMMONAREA_MEDI	69.872297	float64	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
48	COMMONAREA_AVG	69.872297	float64	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
62	COMMONAREA_MODE	69.872297	float64	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
70	NONLIVINGAPARTMENTS_MODE	69.432963	float64	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
56	NONLIVINGAPARTMENTS_AVG	69.432963	float64	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

I – Analyse de donnée

Restriction à la base de donnée application_train.csv
(307511, 122)

Unbalanced values (+ de 90% de la classe 0)

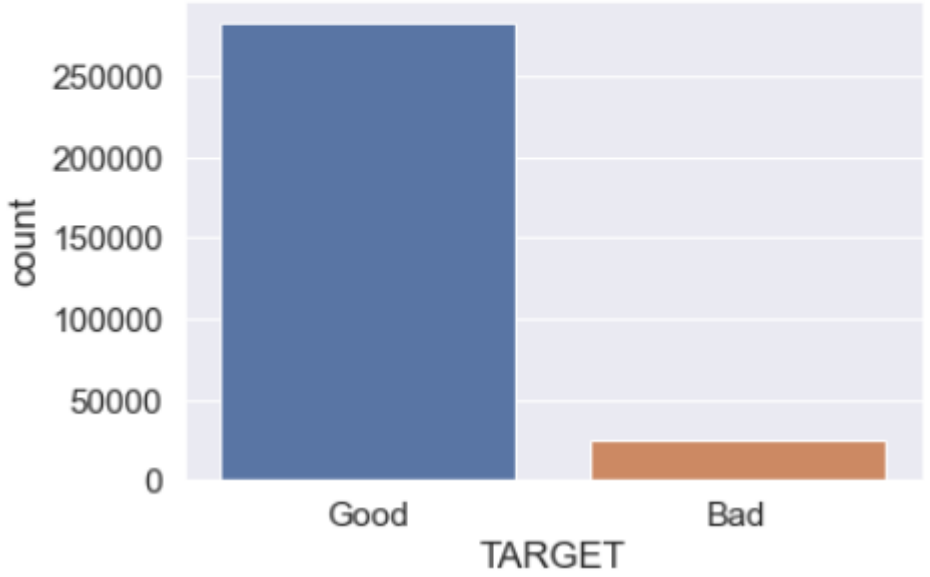
Analyse du jeu de données
(Inspiré du kaggle de Will Koehrsen)

Feature engeniering

Missing values (itérative Imputer)

Train / Test dataframe
(307507, 94)

```
----------
TARGET repartition in the dataframe :
0    282682
1     24825
Name: TARGET, dtype: int64
----------
pourcentage target 1/0 : 8.781952865764357 %
```



Partie II – Choix du modèle et du score d'évaluation

Image : canstockphoto

II – Score d'évaluation

ROC AUC > f1
(receiver operating characteristic > accuracy)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Score :

100 % de 0

→ ***accuracy*** **globale : 91,2 %**

→ ***accuracy*** **classe majoritaire : 100 %**

→ **accuracy classe minoritaire : 0 %**

II – Score d'évaluation

ROC AUC > f1
(receiver operating characteristic > accuracy)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

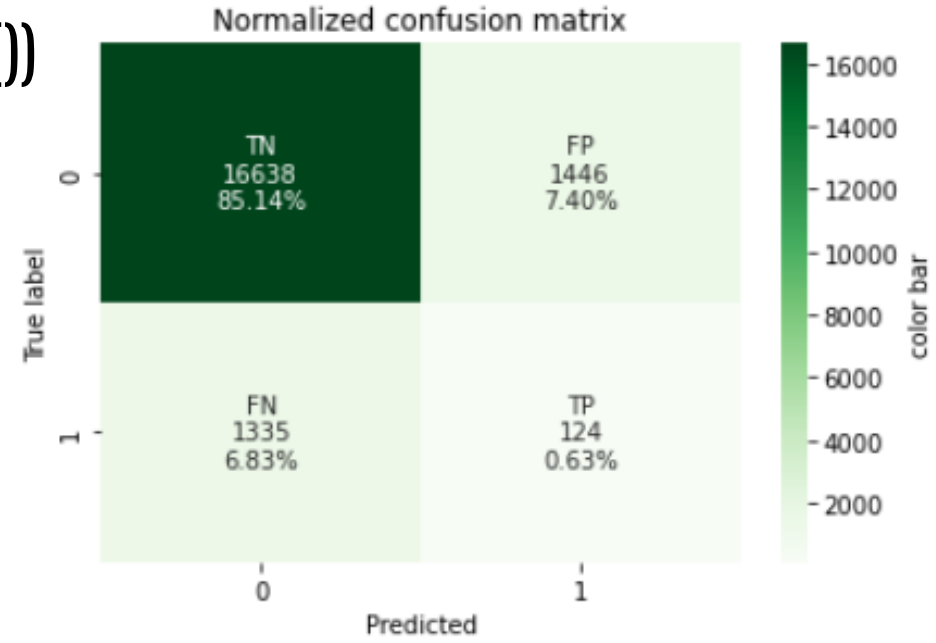
$$FPR = \frac{FP}{FP + TN} \qquad precision\ (TPR) = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN} \qquad F_\beta = \frac{(1 + \beta^2)\ * precision\ * recall}{(\beta^2 * precision) + recall}$$

II – Score d'évaluation

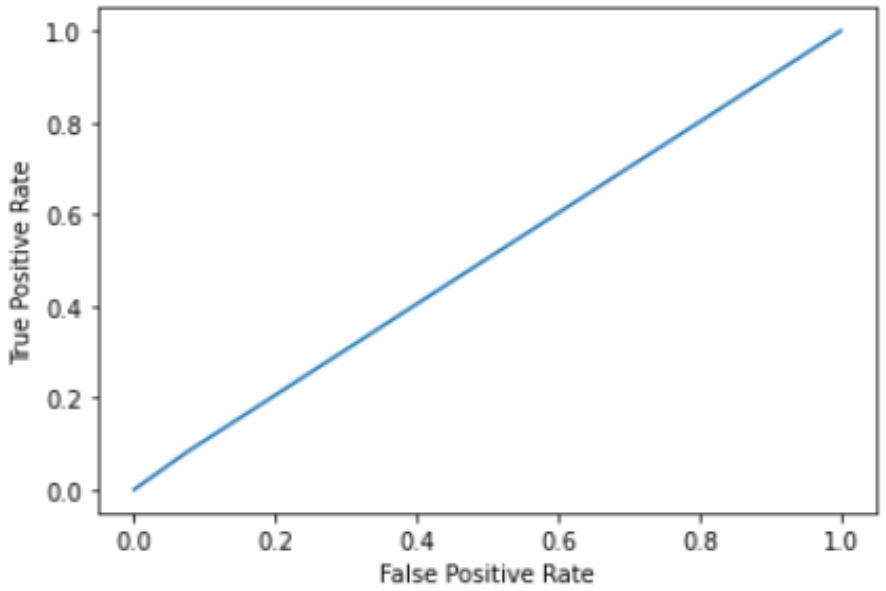
ROC AUC > f1
(receiver operating characteristic > accuracy)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Baseline
(DummyClassifier())



ROC curve



II – Modèles et traitement du déséquilibre

Data prepation
Pipeline (StandardScaler(), MinMaxScaler(), {SMOTE}, model())

Type of models
- Algorithmes linéaires
- Algorithmes non-linéaires
- Algorithmes ensemblistes

Gestion du déséquilibre
- Rien
- class_weight (« cost sensitive »)
- SMOTE (Synthetic Minority Oversampling Technique)

II – LogisticRegression unbalanced treatment

Evaluation des modèles

cross_val_score(RepeatedStratifiedKFold(n_splits=10, n_repeats=3)

	model	scoring	score_trains	std_trains	fitting_time	std_fit	predict_time	score_test	TPR	FNR
0	LR	roc_auc	0.500798	0.038979	16.081664	0.404100	(14.007, sec)	0.918865	0.999383	0.080618
1	LR_weight	roc_auc	0.672930	0.018072	14.453520	0.186147	(12.42, sec)	0.596369	0.590039	0.046950
2	LR_SMOTE	roc_auc	0.685561	0.002308	714.738109	8.851074	(1823.383, sec)	0.577305	0.570291	0.050062
3	LR	make_scorer(fbeta_score, beta=2)	0.001620	0.000882	15.905251	0.384689	(14.018, sec)	0.918865	0.999383	0.080618
4	LR_weight	make_scorer(fbeta_score, beta=2)	0.352641	0.017613	13.732434	0.181817	(12.523, sec)	0.596369	0.590039	0.046950
5	LR_SMOTE	make_scorer(fbeta_score, beta=2)	0.365627	0.005887	710.028121	10.947881	(1807.287, sec)	0.598241	0.590780	0.044897
6	LR	make_scorer(fonction_metier)	-0.998679	0.000728	15.653084	0.390384	(15.175, sec)	0.918865	0.999383	0.080618
7	LR_weight	make_scorer(fonction_metier)	-0.372375	0.032243	13.468927	0.186016	(12.817, sec)	0.596369	0.590039	0.046950
8	LR_SMOTE	make_scorer(fonction_metier)	-0.340377	0.001971	685.754411	15.263532	(1905.344, sec)	0.577929	0.567761	0.045148

II – Présélection

Evaluation des modèles

cross_val_score(RepeatedStratifiedKFold(n_splits=10, n_repeats=3)

Score : roc_auc

LR_B : Basic Logistic Regression

All the other models include a pipeline with SMOTE

LR : Logistic Regression

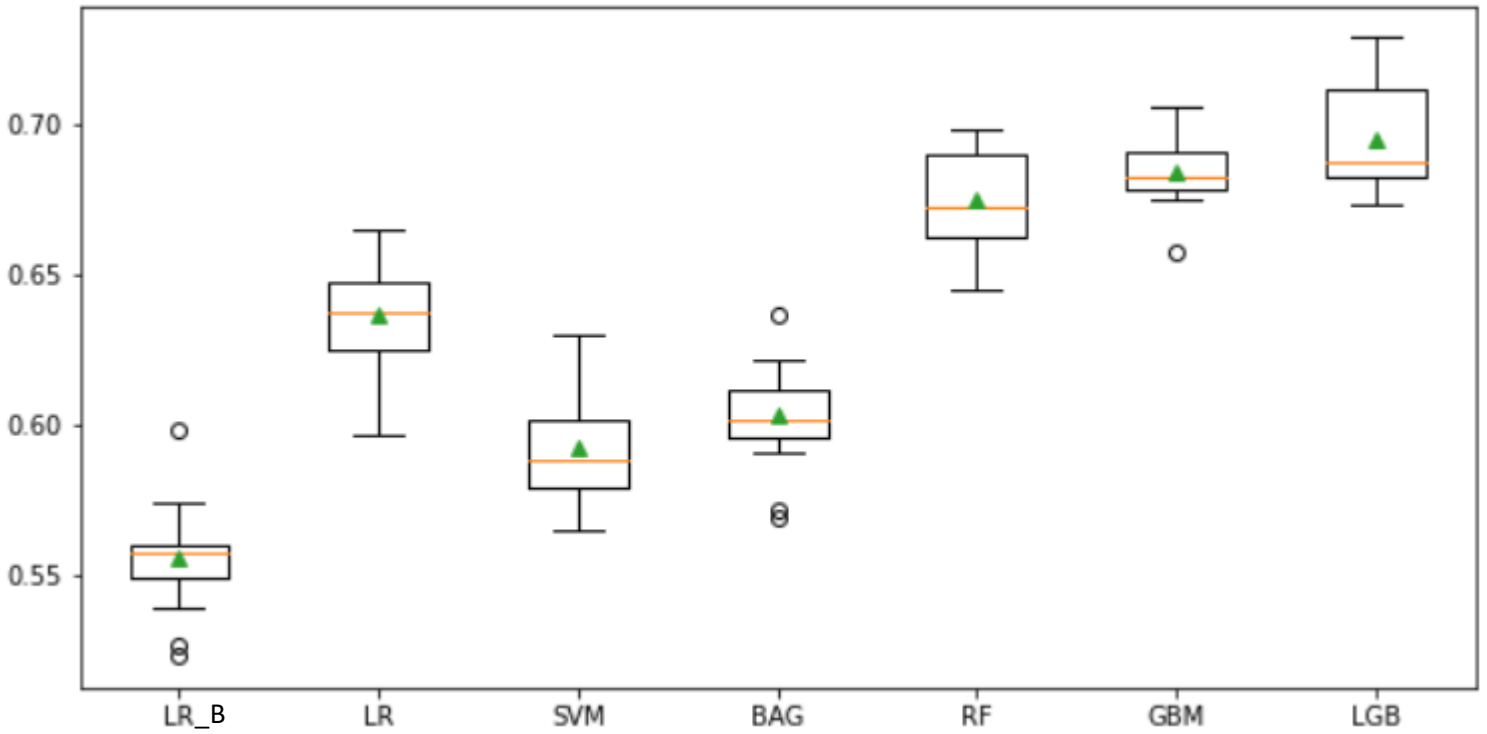
SVM : support vecteur machine cost sensitive

BAG : bagging

RF : Random Forest

GBM : Gradient boosting

LGB_B : Light gradient boosting



II – Présélection

Evaluation des modèles

cross_val_score(RepeatedStratifiedKFold(n_splits=10, n_repeats=3)

Score : custom

LR_B : Basic Logistic Regression

All the other models include a pipeline with SMOTE

LR : Logistic Regression

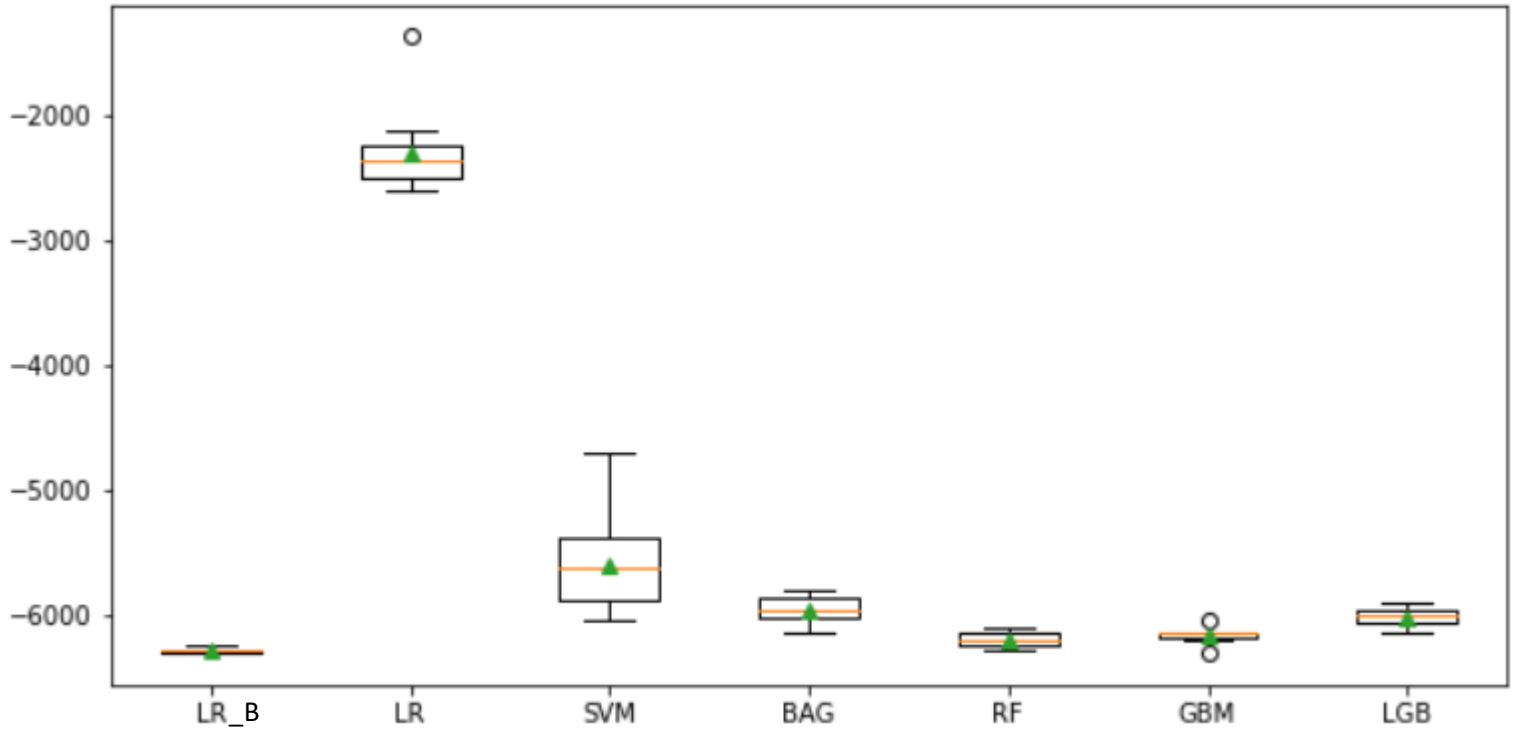
SVM : support vecteur machine cost sensitive

BAG : bagging

RF : Random Forest

GBM : Gradient boosting

LGB_B : Light gradient boosting



II – Présélection

Evaluation des modèles

cross_val_score(RepeatedStratifiedKFold(n_splits=10, n_repeats=3)

Score : fbeta

LR_B : Basic Logistic Regression

All the other models include a pipeline with SMOTE

LR : Logistic Regression

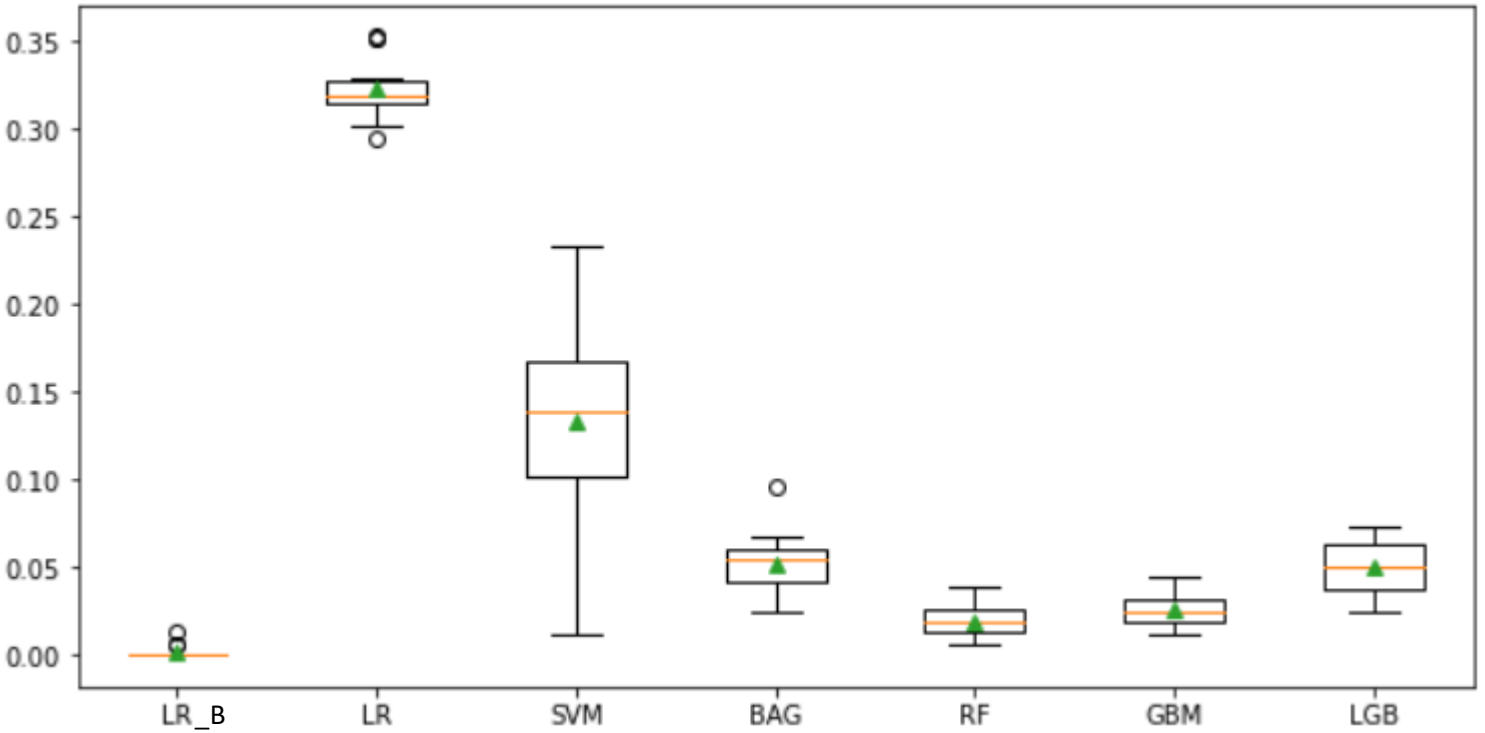
SVM : support vecteur machine cost sensitive

BAG : bagging

RF : Random Forest

GBM : Gradient boosting

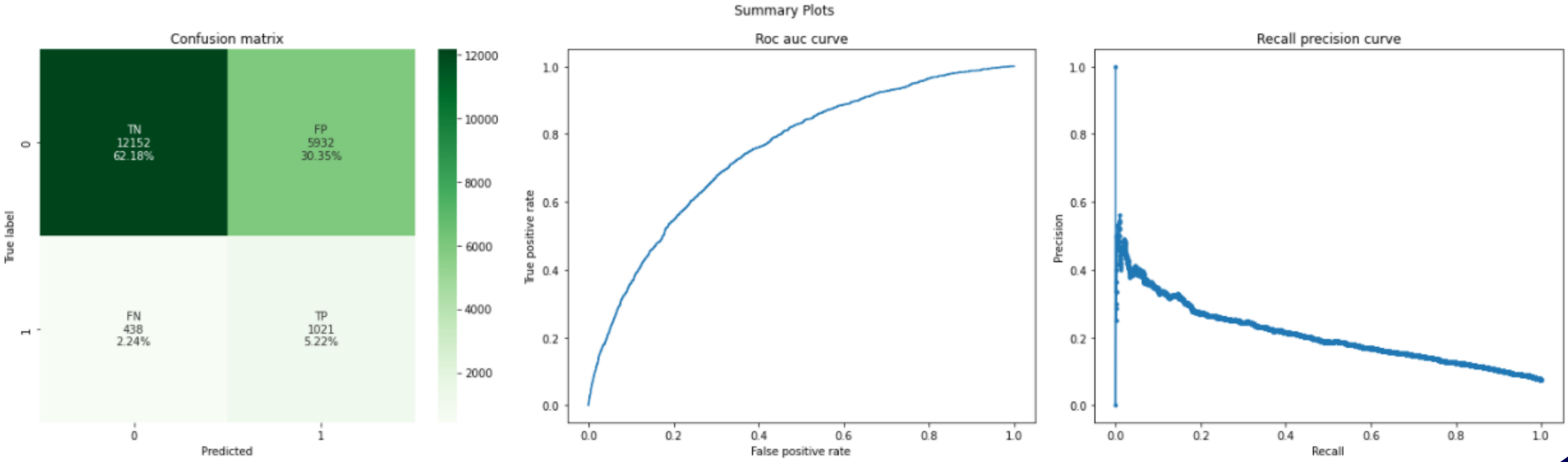
LGB_B : Light gradient boosting



II – Réglages des Hyperparamètres

Evaluation des modèles

GridSearchCV(model(), parameters)

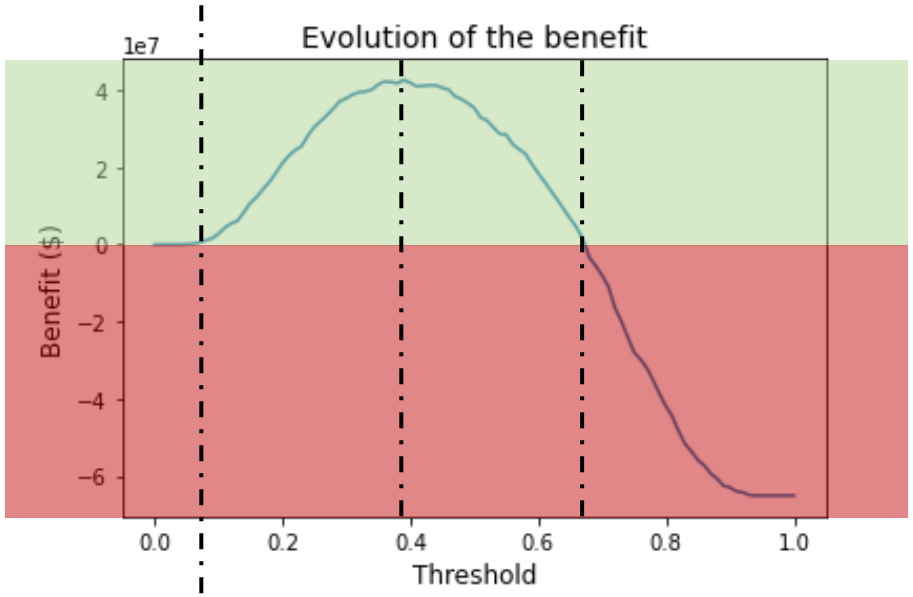


II – Fonction coût et retours métier

1. Fonction coût métier : $(\frac{TP}{FN+TP} - 33 * \frac{FN}{FN+TP})/32$

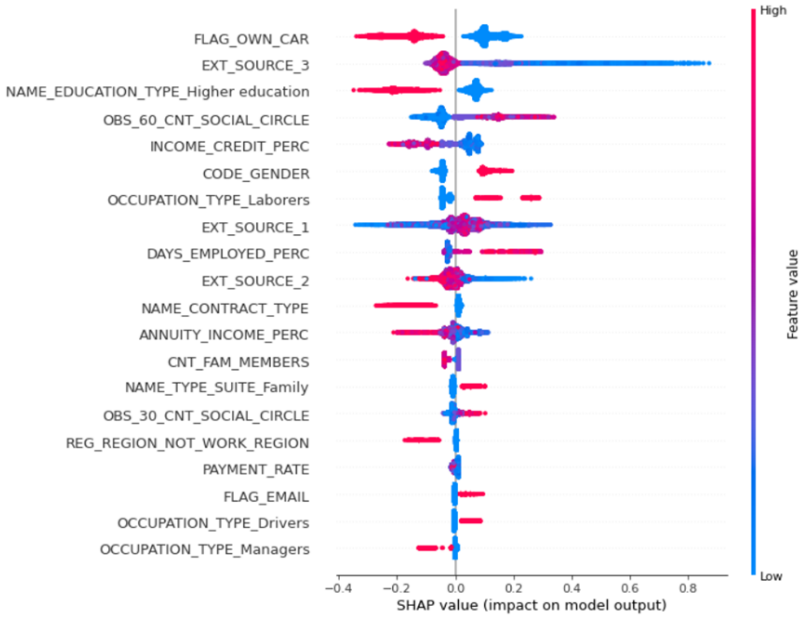
2. Optimisation des hyperparamètres

3. Choix du seuil



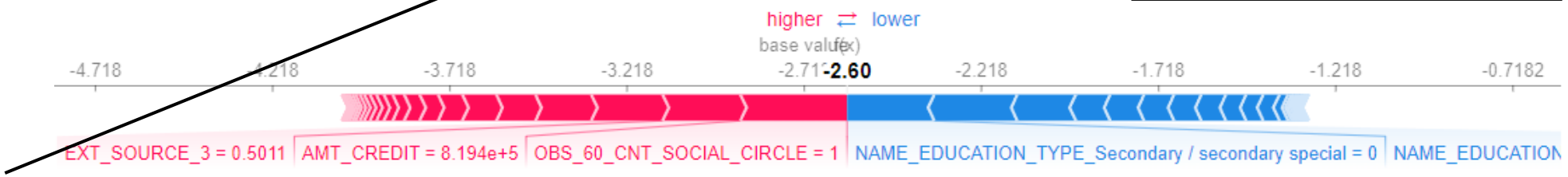
II – Feature importance

SHAP (Shapley Additive exPlanations)



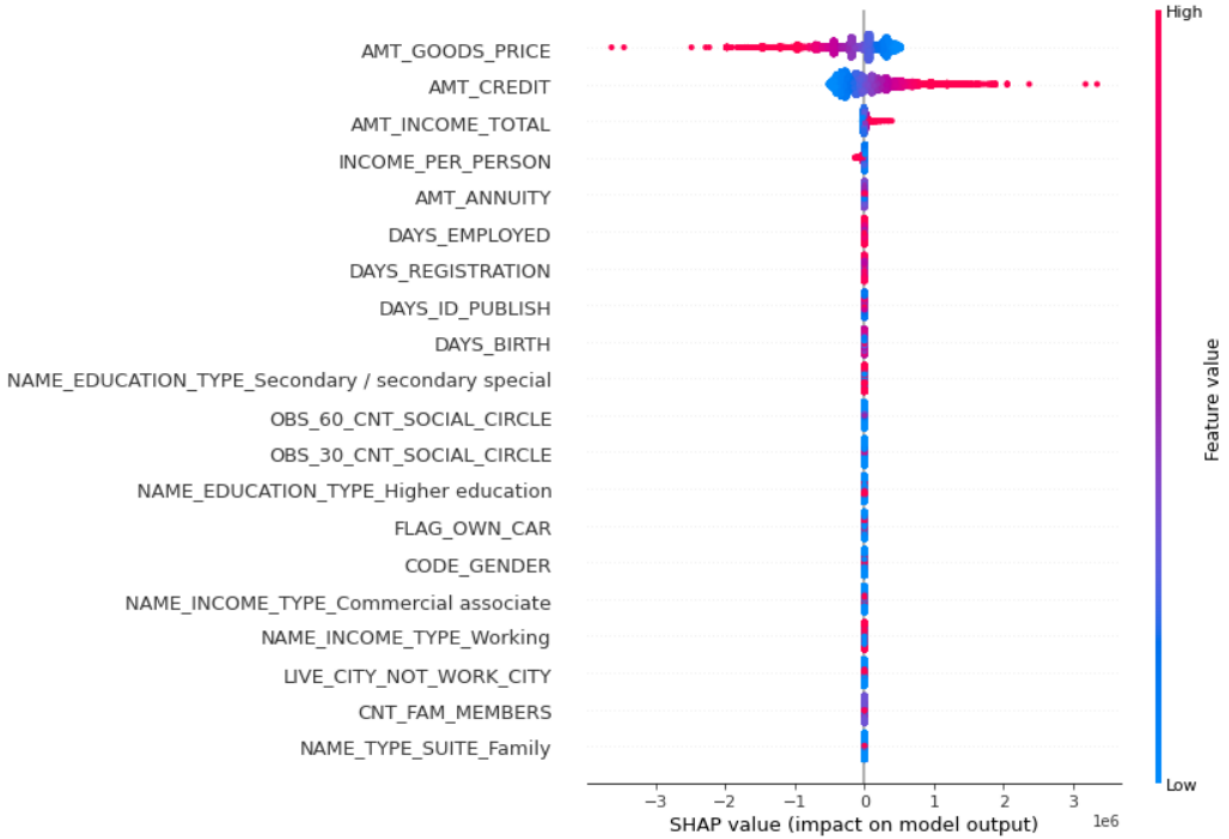
Approche globale

Approche locale

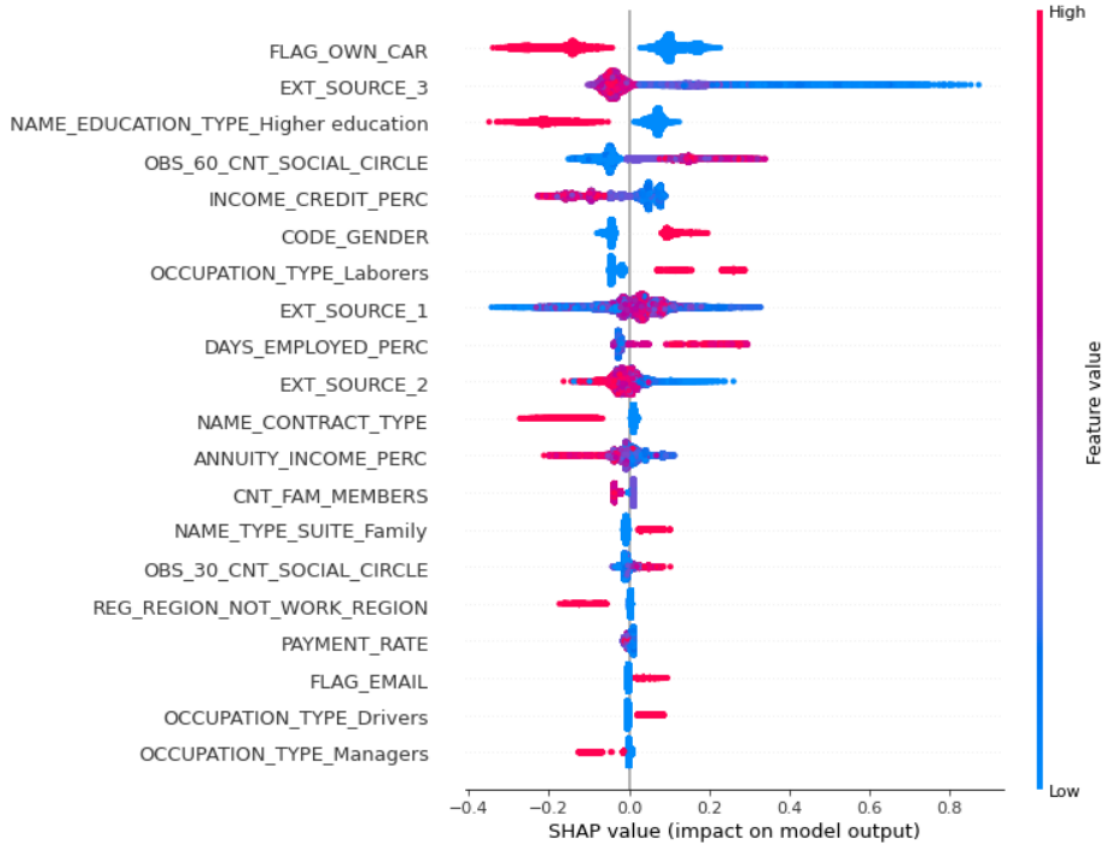


II – Interprétation du modèle

Logistic Reg



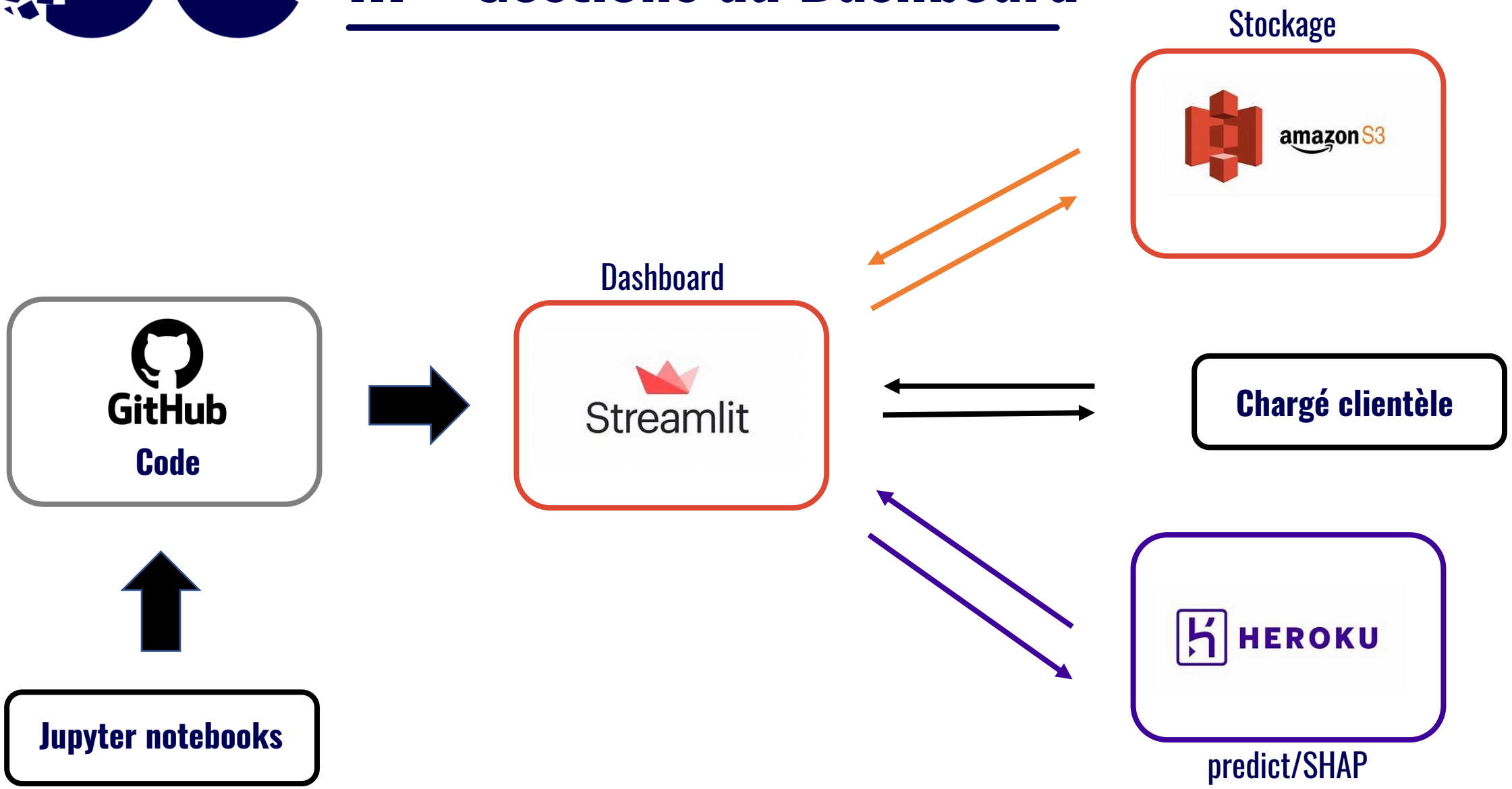
LGBM



Partie III – Présentation du Dashboard

Image : managersenmission

III – Gestions du Dashboard



III – Présentation du Dashboard

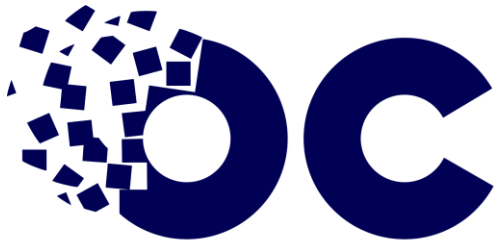
III – Note méthodologique

- La méthodologie d'entraînement du modèle
- La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation
- L’interprétabilité globale et locale du modèle
- Les limites et les améliorations possibles

Conclusion

- Réalisation d'un dashbord interactif et d'une API de prédiction

- Avec De multiples pistes d'amélioration
  - Inclusion de toutes les BDD
  - Plus de modèles de ML à tester
  - Prétraitement spécifiques à chaque modèle
  - Approfondissement des techniques SMOTE
  - Retours métiers

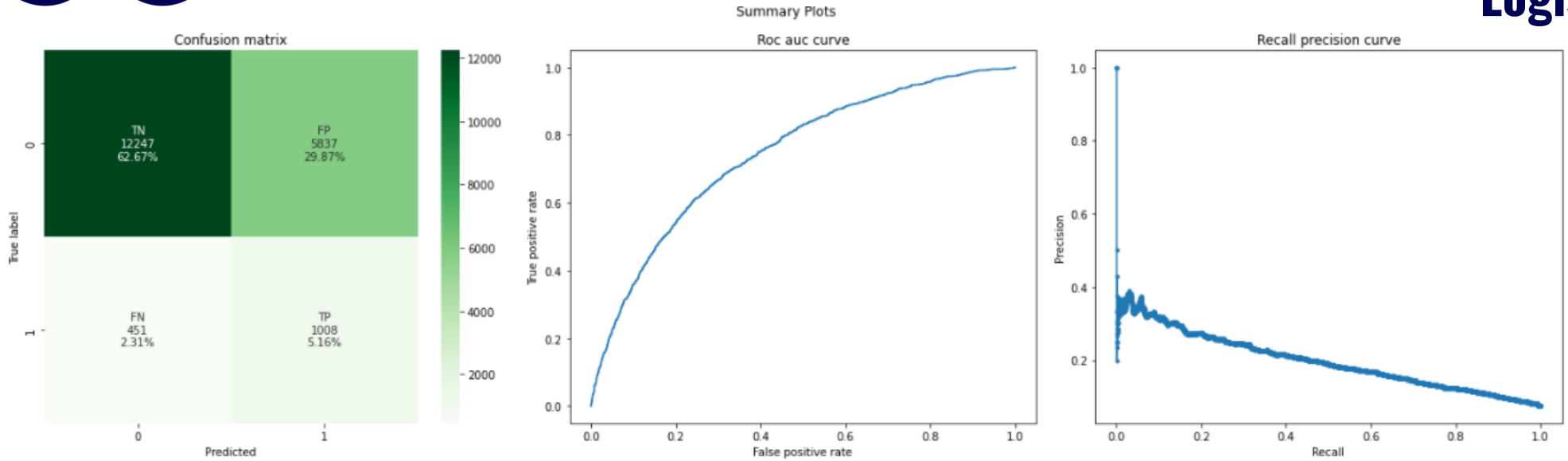


Fin

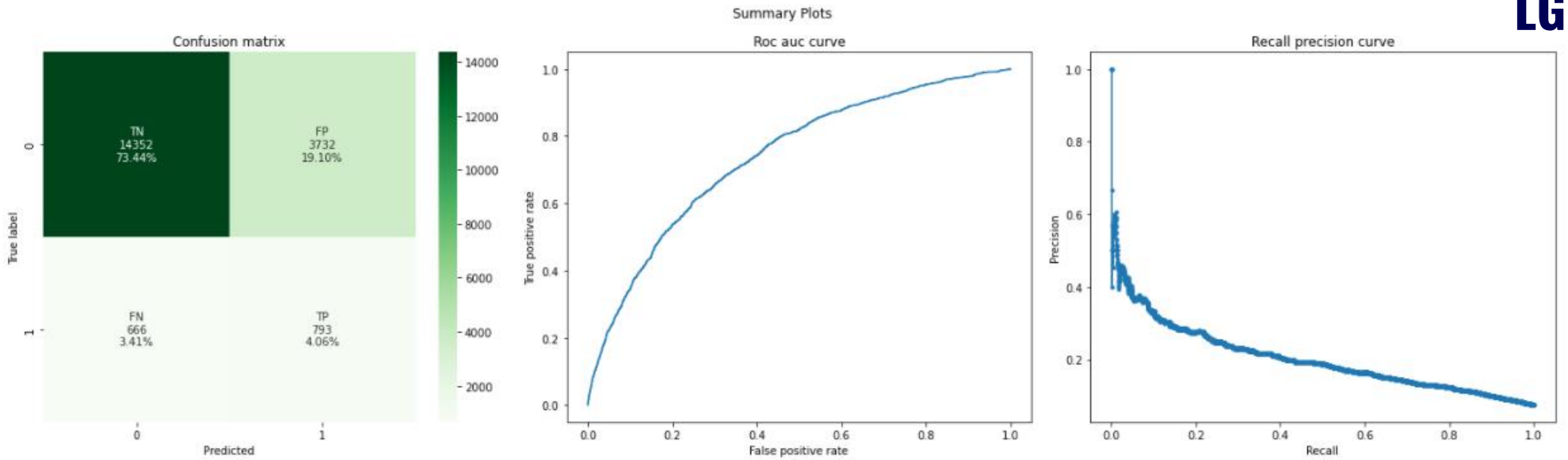
Merci de votre attention !

II – Interprétation du modèle

Logistic Reg



LGBM



II – Réglages des Hyperparamètres