Implémentez un modèle de scoring

Ilaria Mereu

Mission

Prêt à dépenser

Société financière opérant sur le marché des prêts personnels

Nos objectifs:

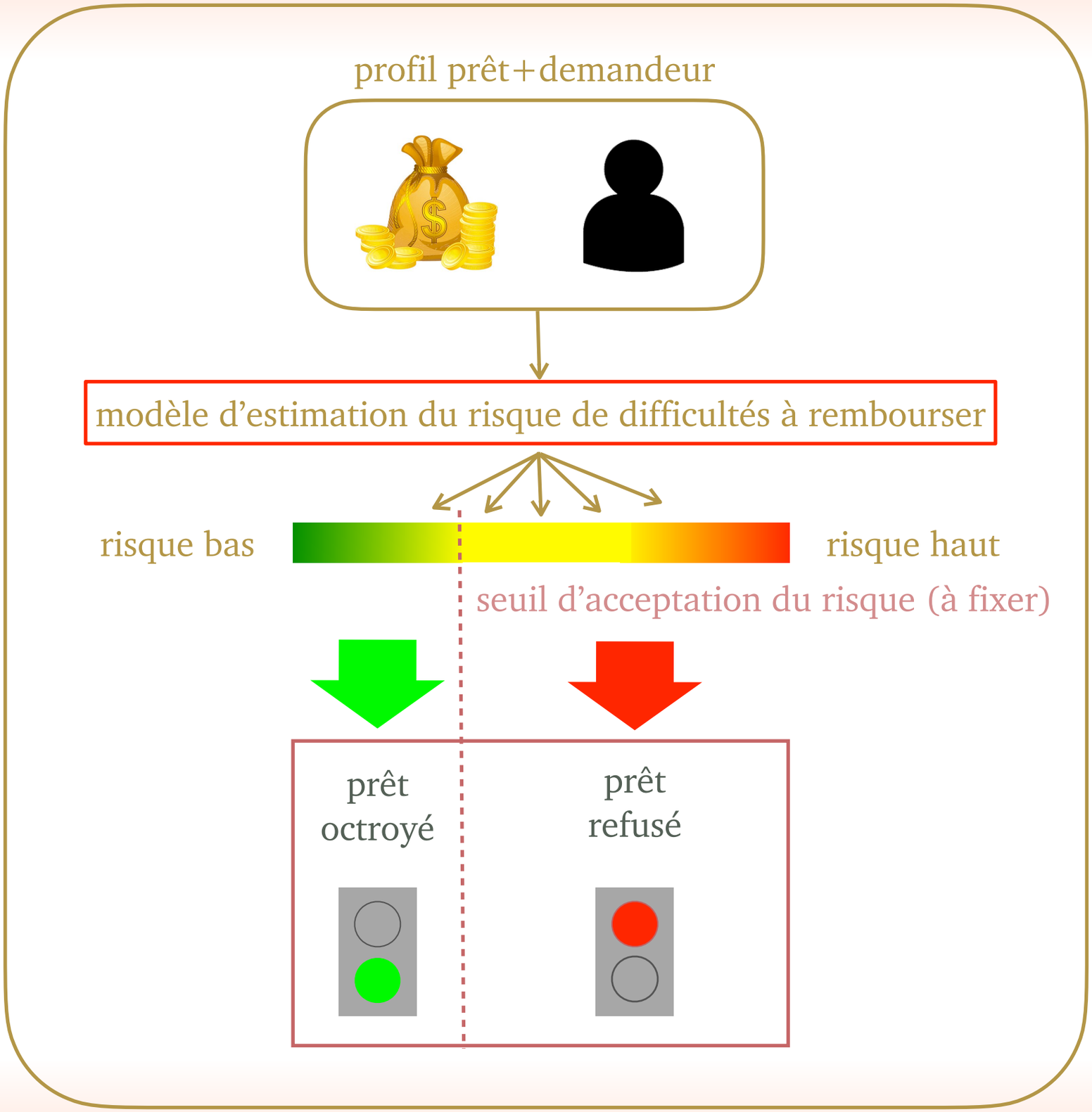
À partir d'un jeu de données contenant un grand nombre d'informations personnelles sur les demandeurs d'un prêt:

- **Implémenter un modèle capable de predire la capacité d'un credit d'être remboursé.**
- **Déployer une dashboard qui rende transparents les critères sous-jacent l'approbation ou le refus d'un prêt.**

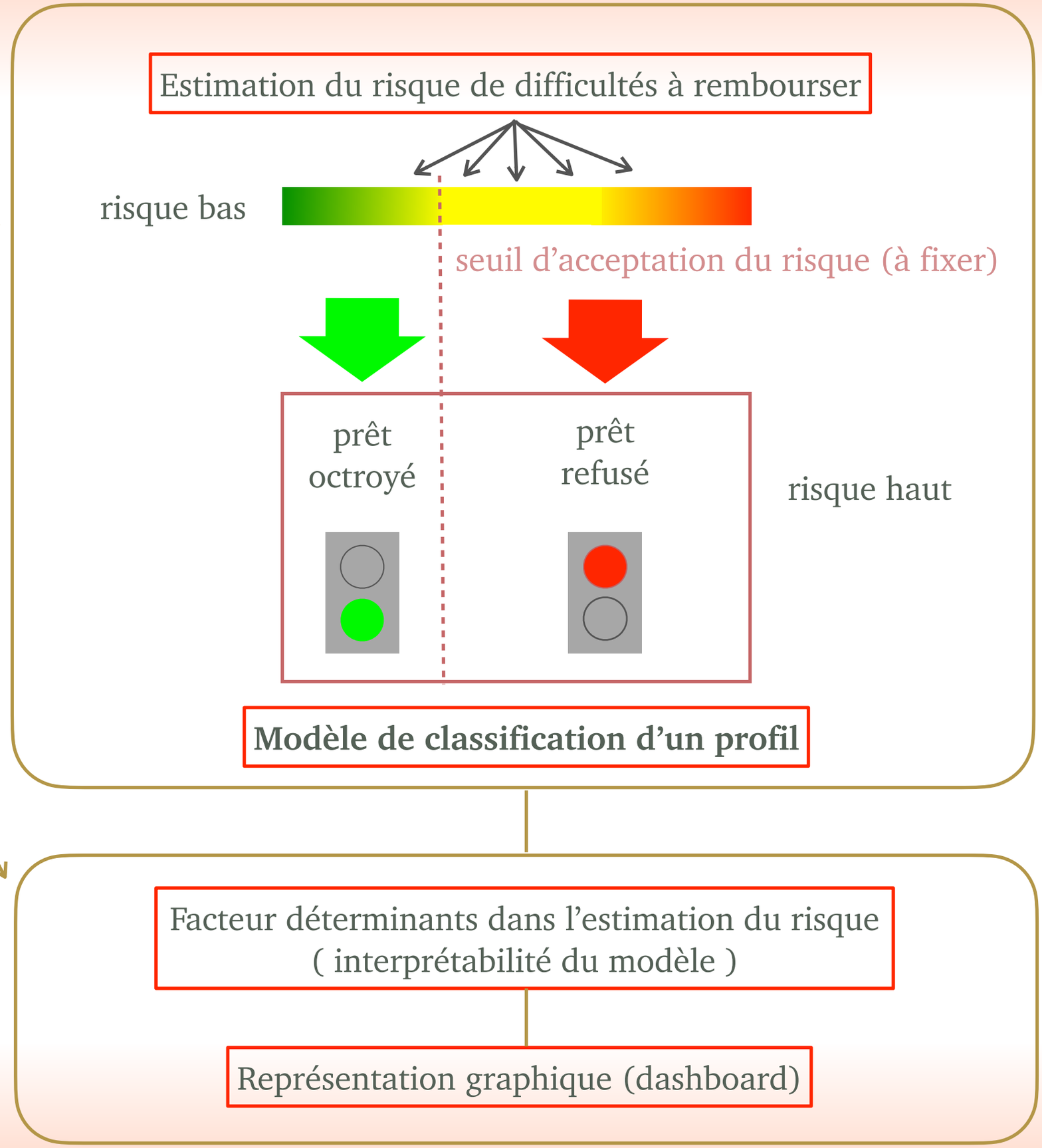
Sommaire

1. Description de la problématique

Mission

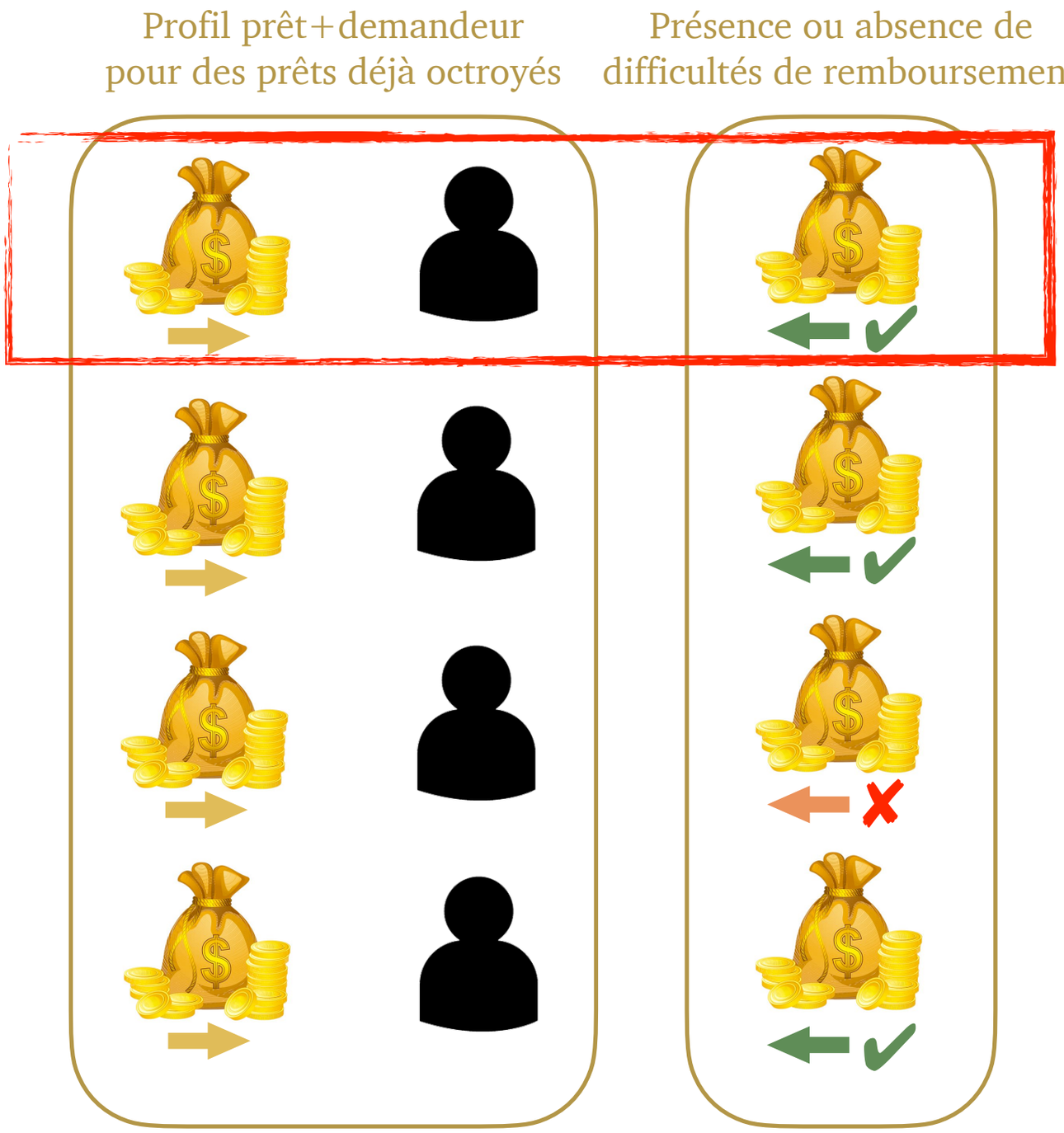

profil prêt+demandeur
modèle d'estimation du risque de difficultés à rembourser
risque bas
risque haut
seuil d'acceptation du risque (à fixer)
prêt octroyé
prêt refusé

Mission


Estimation du risque de difficultés à rembourser
risque bas
seuil d'acceptation du risque (à fixer)
prêt octroyé
prêt refusé
risque haut
Modèle de classification d'un profil
Facteur déterminants dans l'estimation du risque (interprétabilité du modèle)
Représentation graphique (dashboard)

2. Préparation du dataset

Phase 1: entraînement du modèle de classification



Chaque ligne passée au modele pour l'entraînement devra contenir:

- **le profil prêt+demandeur, soit:**
  - le montant du prêt
  - des nombreuses informations personnelles sur le demandeur:
    - activité,
    - revenus
    - âge
    - zone de résidence
    - …
- **<u>Les éventuelles difficultés à rembourser le prêt en question</u>**
  (notre variable target)

Phase 1: entraînement du modèle de classification

Le profil prêt+demandeur est contenu en 6 fichiers:

POS_CASH_balance.csv
bureau.csv
bureau_balance.csv
credit_card_balance.csv
installments_payments.csv
previous_application.csv

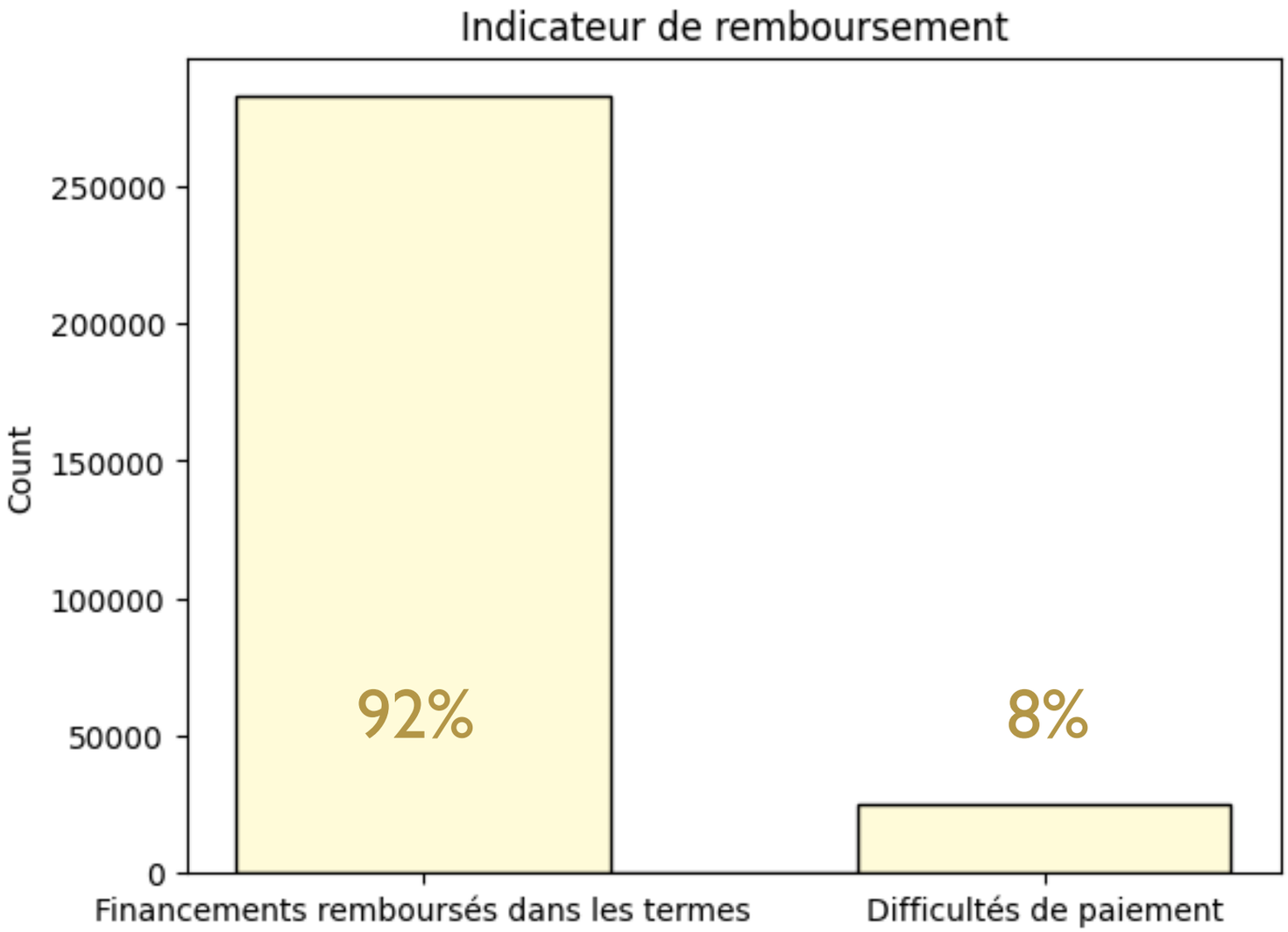
Testo

Adaptation d'un kernel kaggle pour le pretraitement des données

On obtient un seul fichier de 307507 lignes et ~500 colonnes. Ce fichier présente deux difficultés techniques à traiter:

1) **Des valeurs manquantes: imputation avec la mediane de la colonne**
2) **Fort déséquilibre de la variable target**

Déséquilibre de la variable target



Total: 307507; valeurs absolues: 282682 et 24825.

Phase 1: traitement du déséquilibre de la variable target

Trois stratégies:

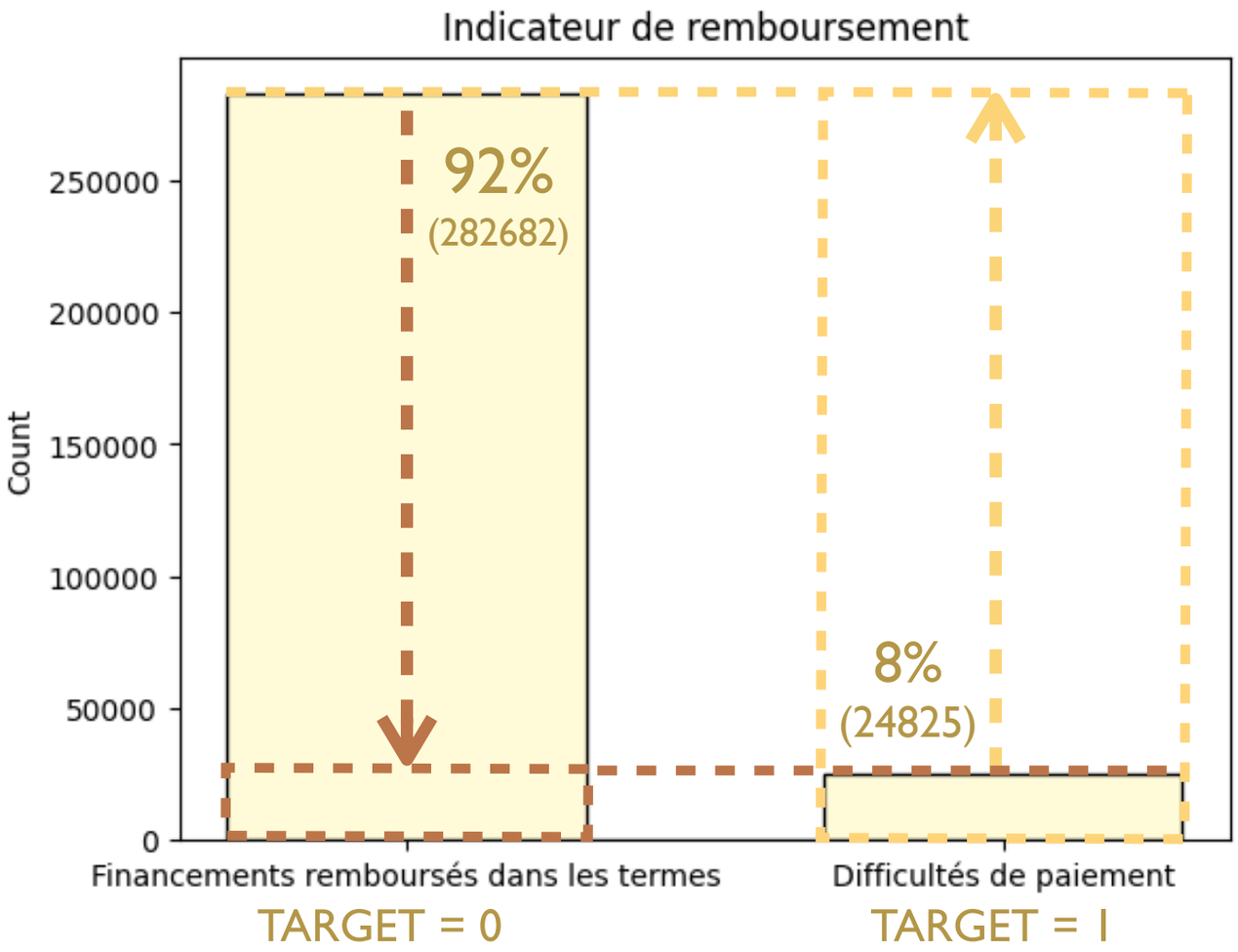
1) Choix d'un classificateur capable de reconnaitre et pondérer ce déséquilibre (HistGBC)

2) Élimination du déséquilibre:
   1) sur-echantillonnage (SMOTE)
   2) sous-echantillonnage: random undersampling ('rus') et cluster-centroids undersampling ('ccus')

Phase 1: traitement du déséquilibre de la variable target

1

Sous-échantillonnage:
Reéchantillonnage
et réduction
du groupe majoritaire
(Cluster Centroids)



Total profils originaux: 307507

2

Sur-échantillonnage
Reéchantillonnage
et expansion
du groupe minoritaire
(SMOTE)

Total après réechantillonnage:
49650 profils

Total après réechantillonnage:
565364 profils

3. Choix du modèle

Phase 1: entraînement du modele de classification

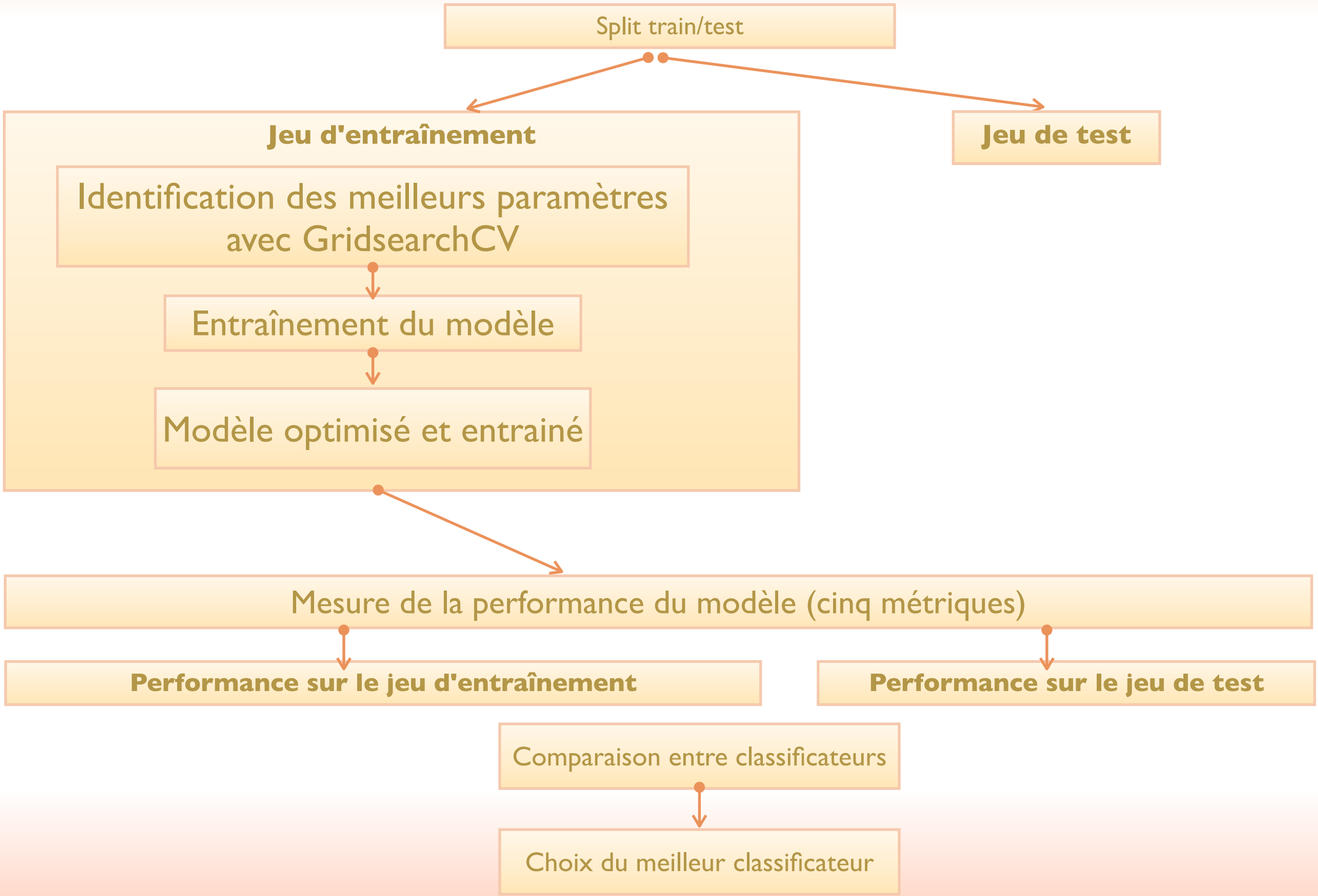
On sélectionne les 100 features le plus déterminantes pour la classification avec kbest

Reéchantillonnage sur le jeu avec le nombre de features réduit.

Entraînement et test de plusieurs modèles de classification:

- Regression Logistique (‘LogReg’)
- Gaussian Naive Bayes (‘GauNB’)
- Linear Support Vector (‘SVM’)
- Random forest (‘RandFC’)
- Classificateur du type k-nearest neighbors (‘KNC’)
- Arbre de classification du type Histogram-based Gradient Boosting (‘HistGBC’)
- Classificateur du type Gradient Boosting du package XGBoost (‘XGBClass’)

Schéma du choix du classificateur



Déterminer la qualité de la classification

		Valeur cible	
		0 (solvable)	1 (non solvable)
Prévision	0: prévision de client solvable	00 vrai négatif TN	01 faux négatif FN octroi erroné
	1: prévision de client non solvable	10 faux positif FP refus erroné	11 vrai positif TP

Nos objectifs:

1) réaliser la classification

2) quantifier l'impact des différents erreurs

Déterminer la qualité de la classification

- Accuracy: fraction des prévisions correctes.
- Recall: fractions de VP sur le total des positifs
- $F_{\beta(=3)}$: combinaison de précision et recall

$$\text{precision} = \frac{tp}{tp + fp},$$

$$\text{recall} = \frac{tp}{tp + fn},$$

$$F_\beta = (1 + \beta^2)\frac{\text{precision} \times \text{recall}}{\beta^2\text{precision} + \text{recall}}$$

- Roc auc: aire au dessous de la courbe

- R^2 : représente la proportion of variance expliquée par le modèle. Fournit une autre mesure du pouvoir prédictif du modèle.

Mesures de la performance

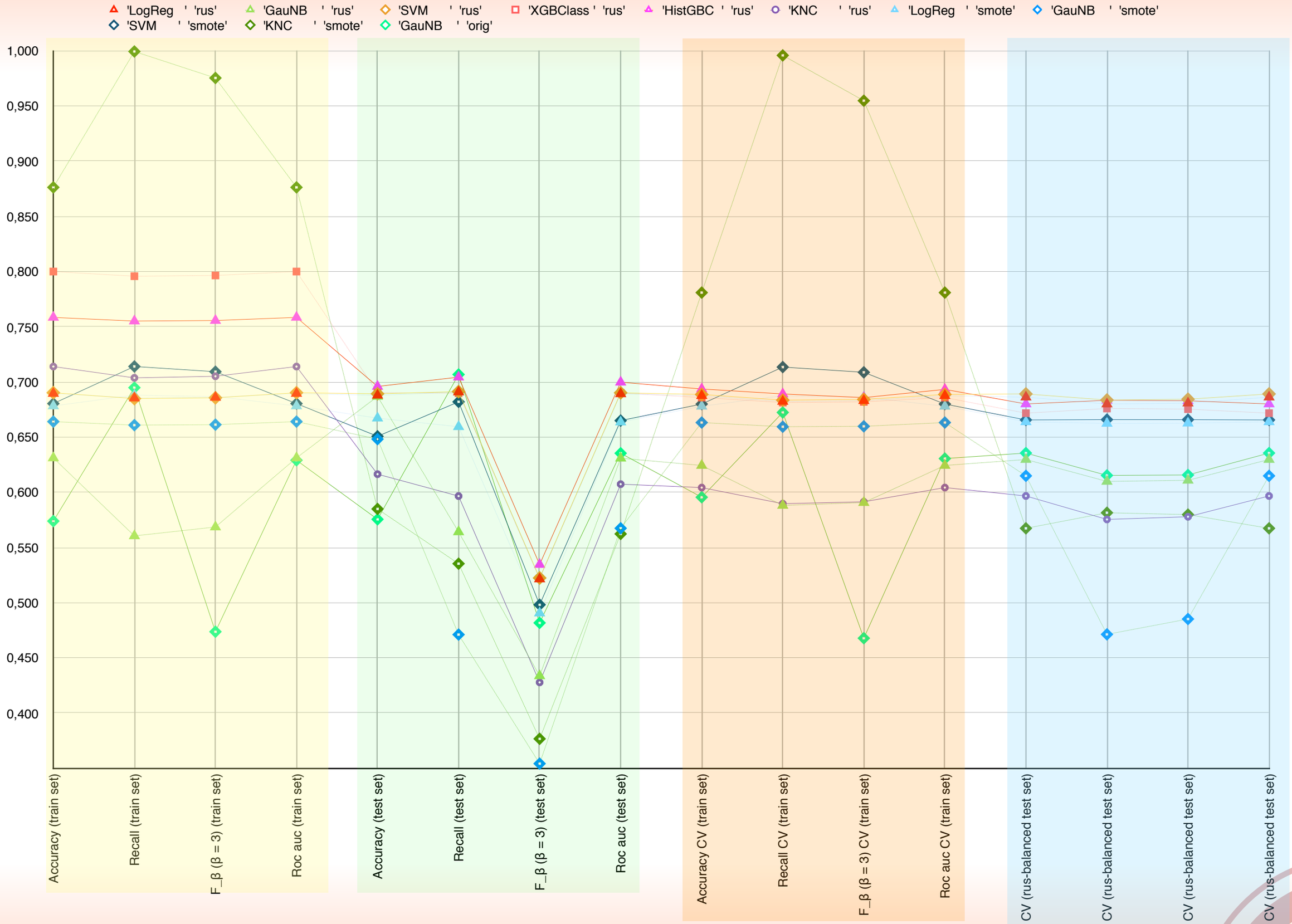
Performance directe du modèle sur le jeu d'entraînement

Performance directe du modèle sur le jeu de test

Performance mesurée par validation croisée sur le jeu d'entrainement (cross_val_score)

Performance mesurée par validation croisée sur le jeu de test (cross_val_score)

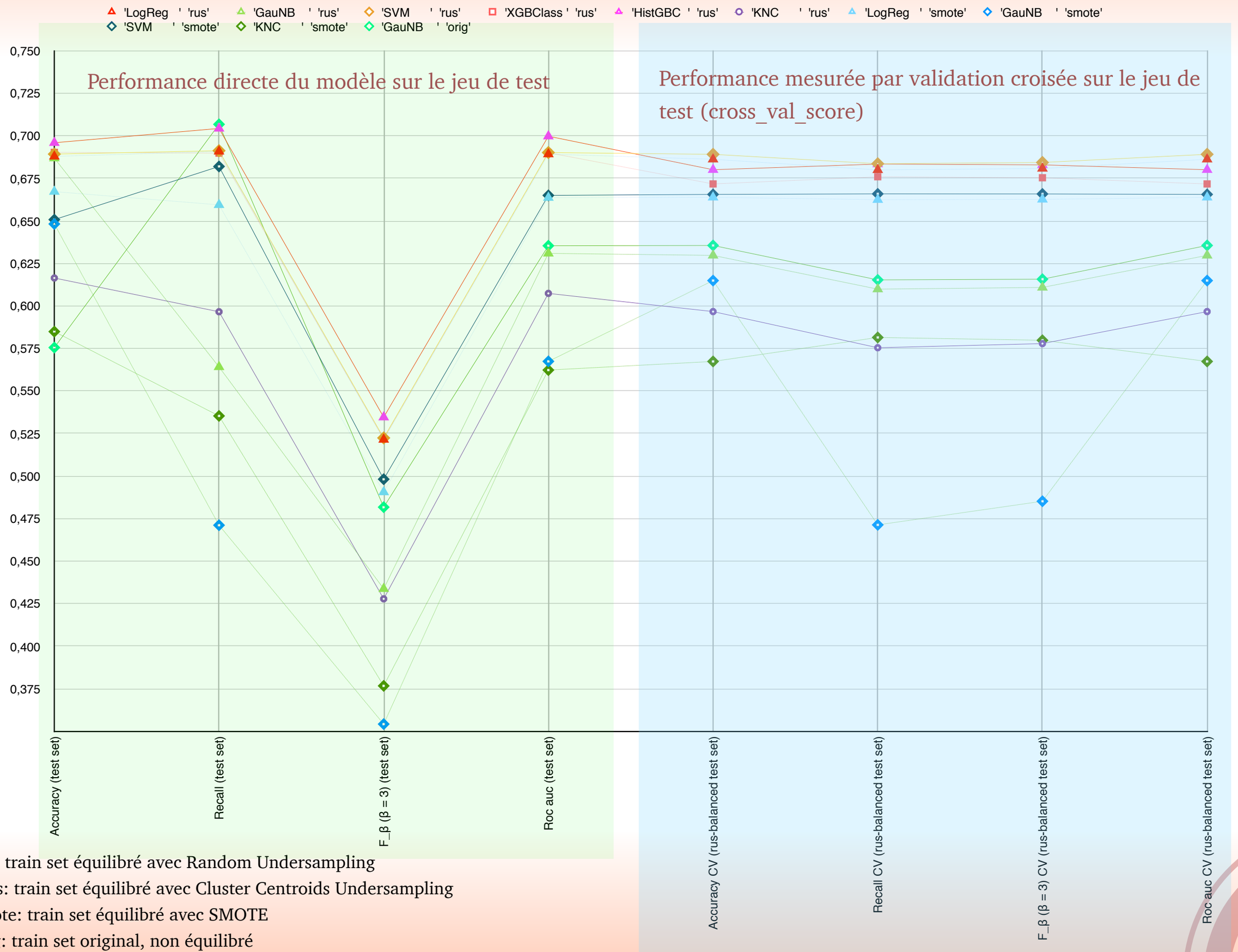
Meilleures performances (jeux d'entraînement et de test)



rus: train set équilibré avec Random Undersampling
ccus: train set équilibré avec Cluster Centroids Undersampling

smote: train set équilibré avec SMOTE
orig: train set original, non équilibré

Meilleures performances (jeu de test)



rus: train set équilibré avec Random Undersampling
ccus: train set équilibré avec Cluster Centroids Undersampling
smote: train set équilibré avec SMOTE
orig: train set original, non équilibré

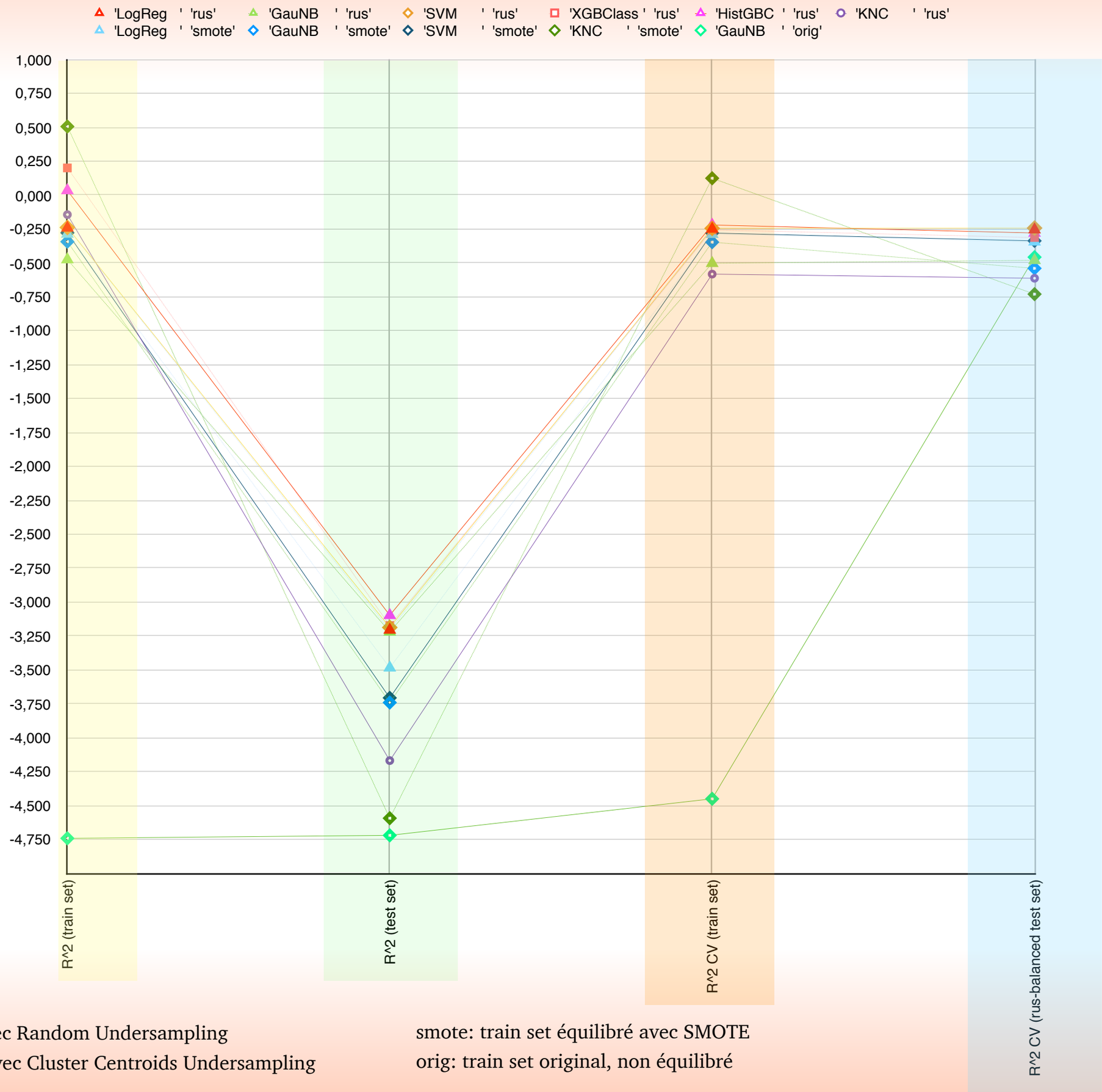
R^2 (jeux d'entraînement et de test)

Performance directe du modèle sur le jeu d'entraînement

Performance directe du modèle sur le jeu de test

Performance mesurée par validation croisée sur le jeu d'entraînement (cross_val_score)

Performance mesurée par validation croisée sur le jeu de test (cross_val_score)



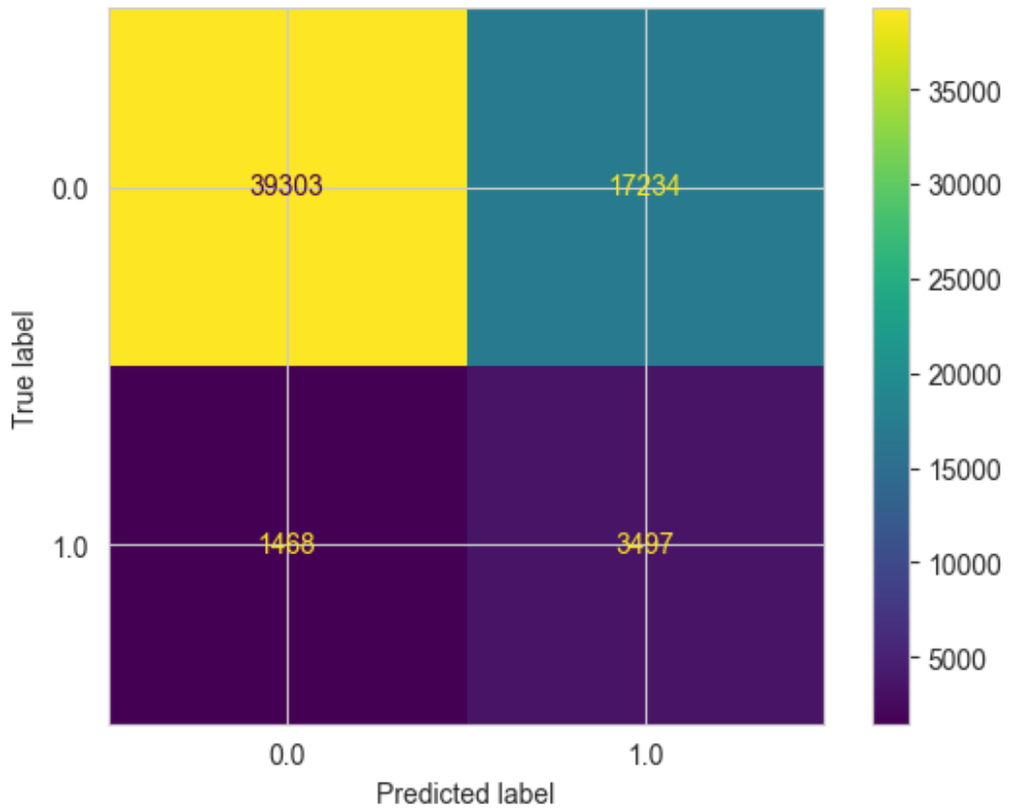
rus: train set équilibré avec Random Undersampling
ccus: train set équilibré avec Cluster Centroids Undersampling

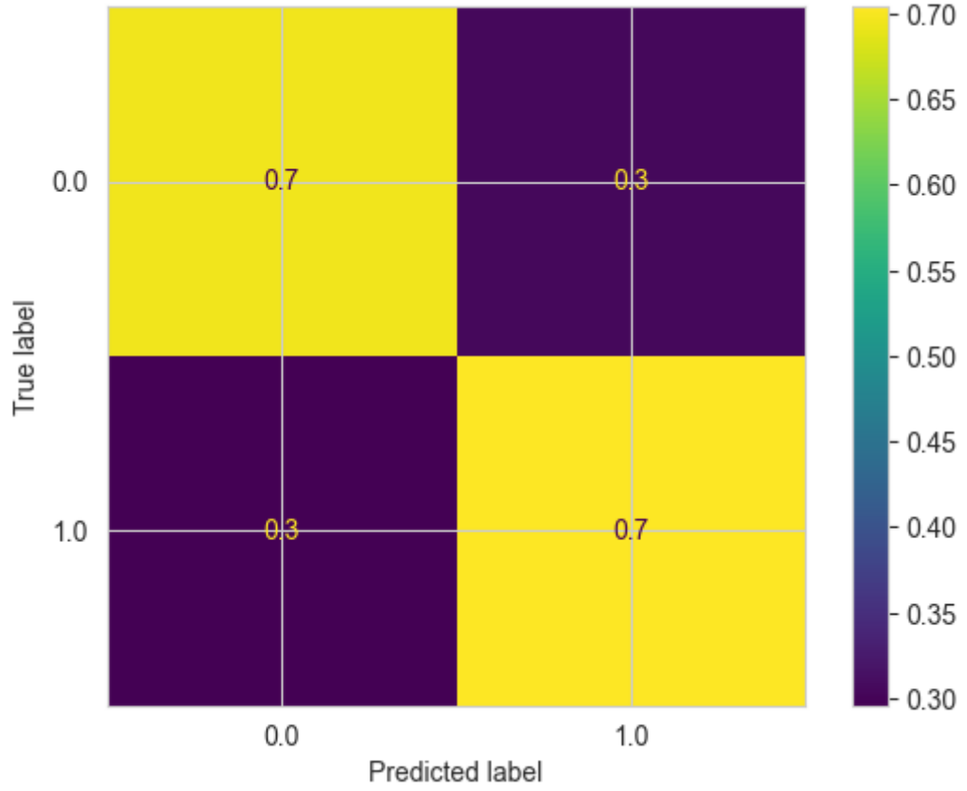
smote: train set équilibré avec SMOTE
orig: train set original, non équilibré

Modèle retenu

HistGradientBoosting Classifier:

- entraîné sur le jeu équilibré avec Random Undersampling
- critère de scoring pour GridSearchCV : recall
- meilleurs paramètres issus de GridSearchCV : {'max_iter': 200, 'max_leaf_nodes': 40}

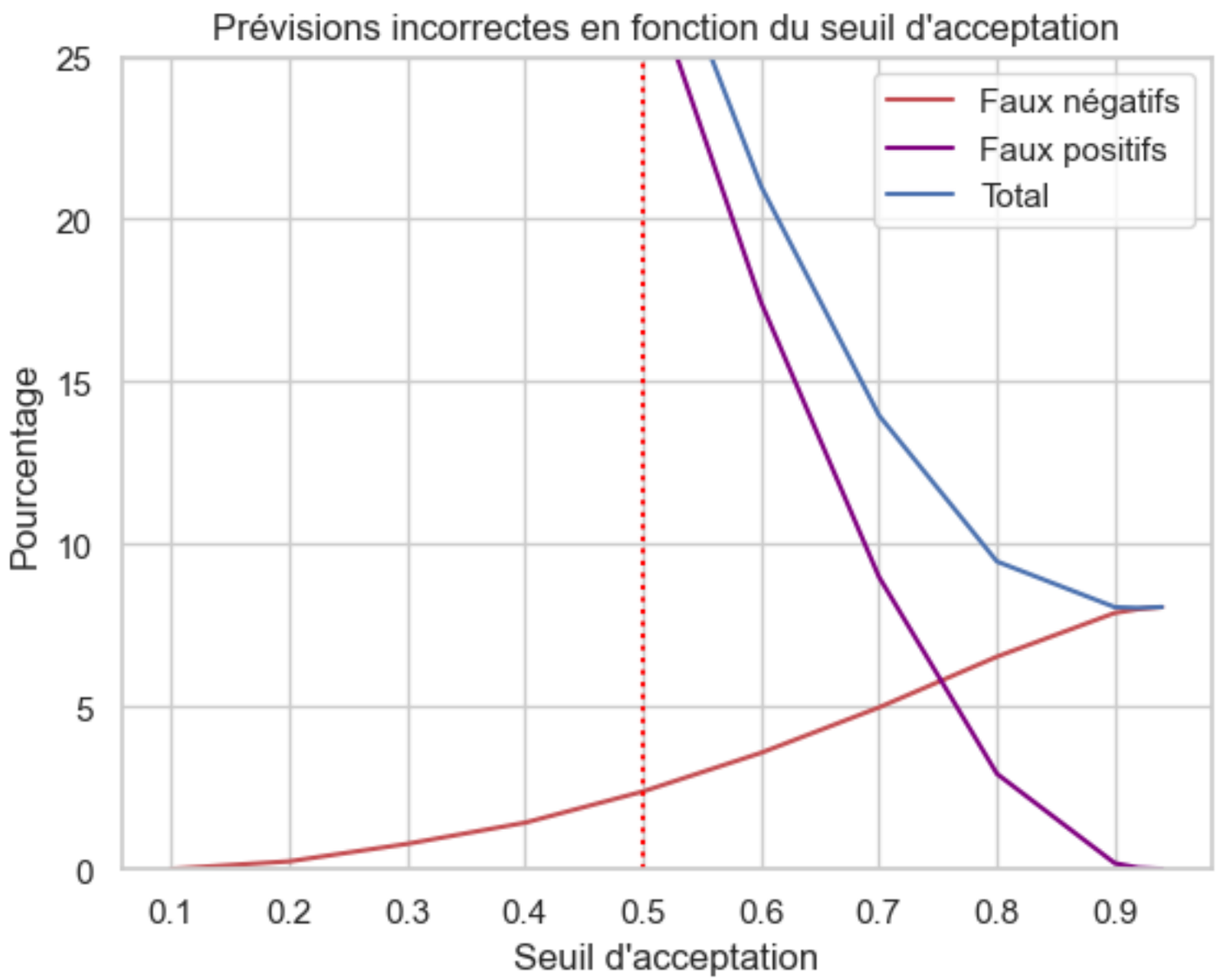




Le modèle identifie correctement le 70% de chaque classe

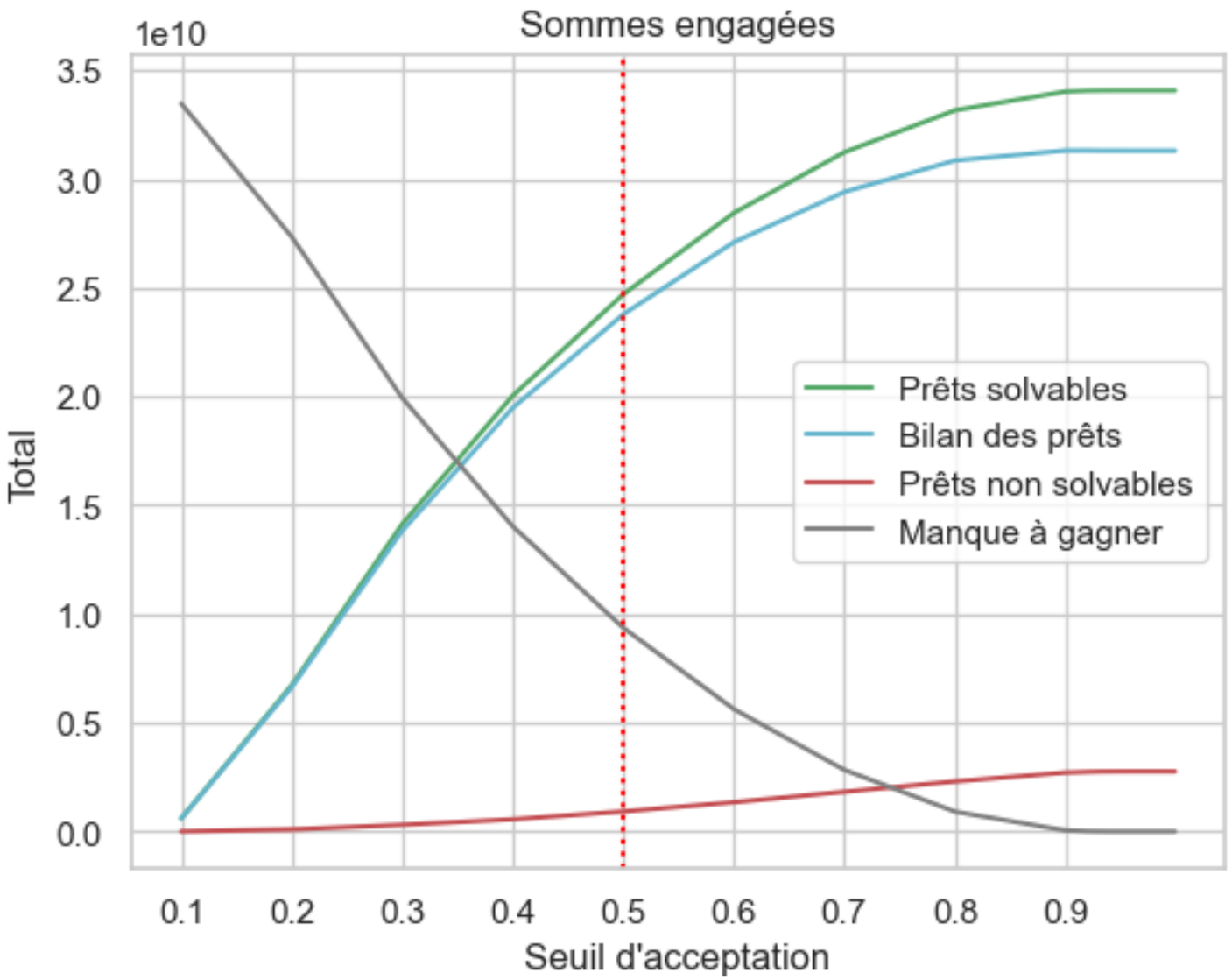
4. Coût des erreurs de classification

Bilan des prévisions



Le FP tendent à zéro avec le seuil croissant, alors que le FN (les plus problématiques) rejoignent un plateau

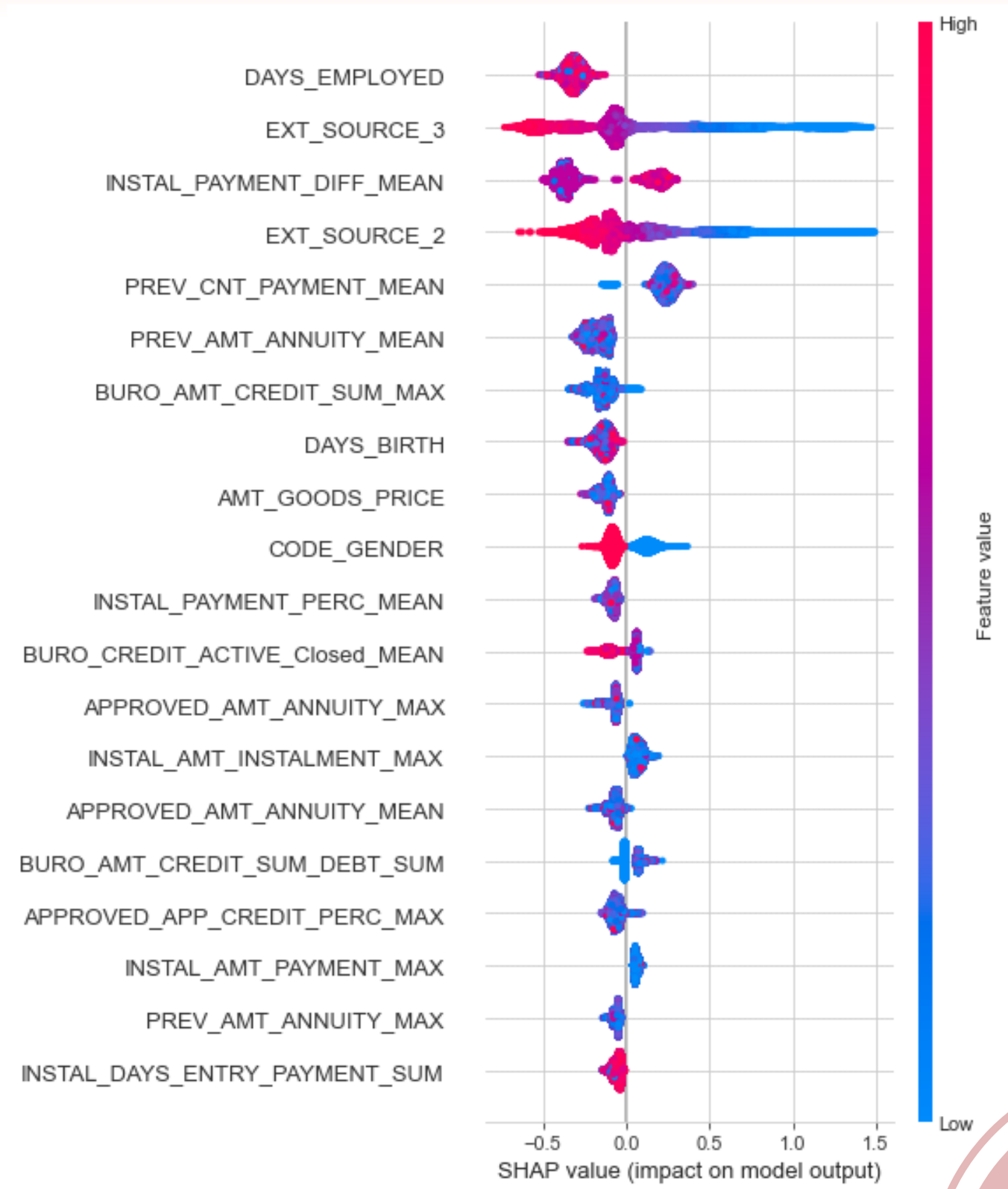
Bilan des sommes engagées



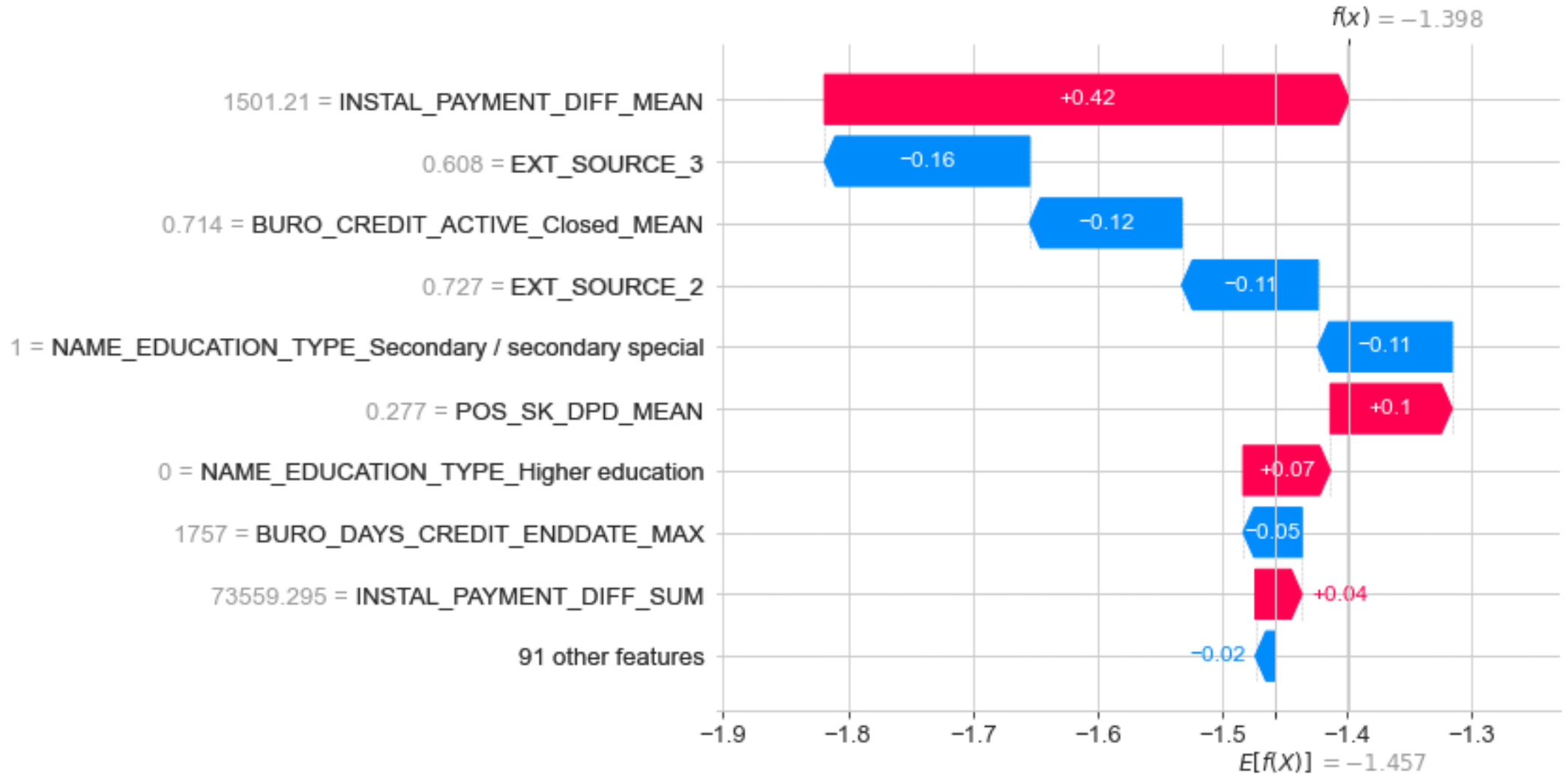
Du point de vue des sommes engagées, le bilan des prévisions du modèle indique qu'un seuil > 0.9 maximise les FN mais aussi le bilan total.

5. Interprétabilité du modèle

Interprétabilité globale

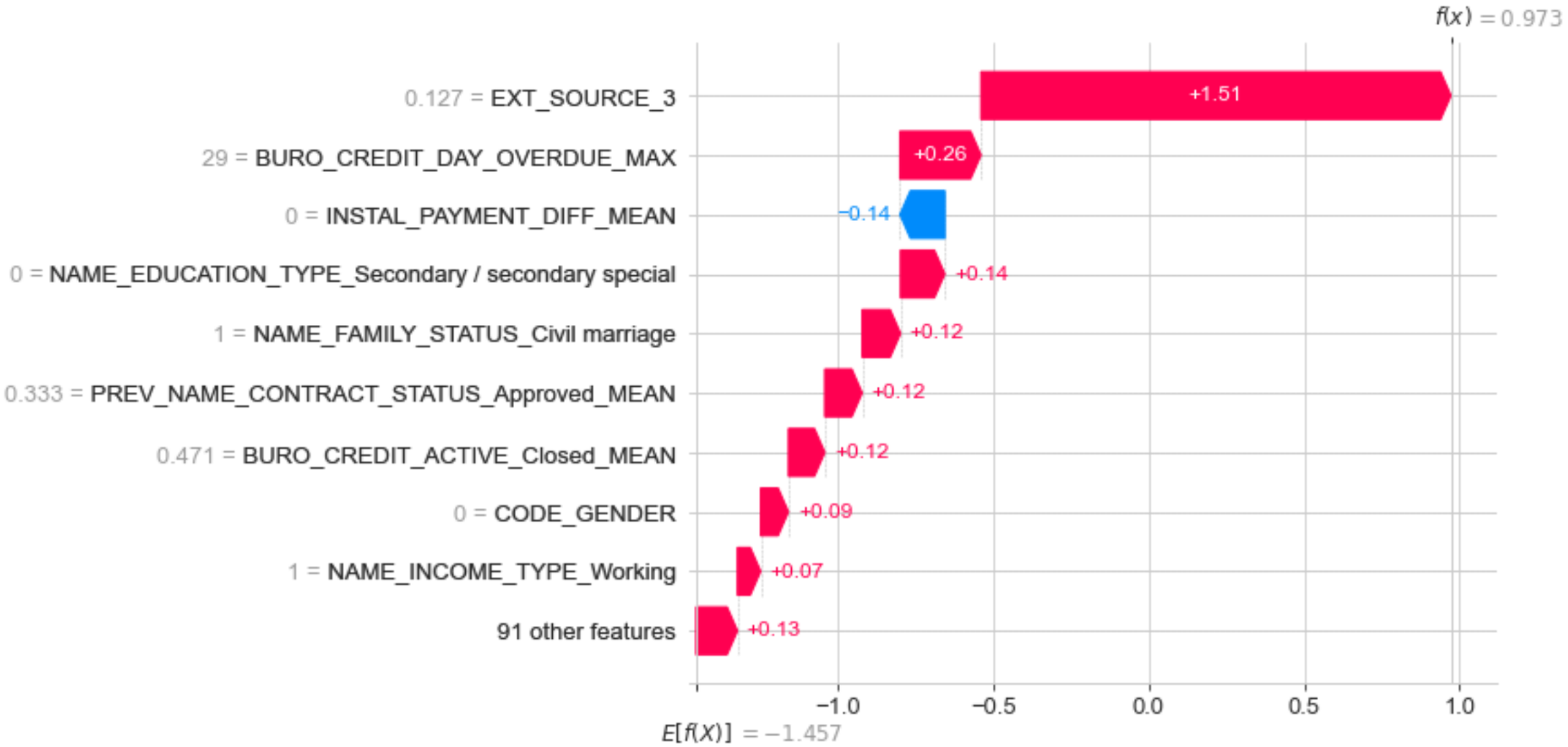


Interprétabilité locale



Au niveau des profils individuels, on peut voir quels facteurs ont l'impact le plus fort sur la classification. Ici un exemple pour un profil fort.

Interprétabilité locale



… et ici un profil plus faible.

6. Tableau de bord

Tableau de bord

Le tableau de bord est disponible à ce lien.

Il a est composé de:

- Un frontend réalisé avec Streamlit et déployé sur Streamlit Community Cloud.
- Cette page est branchée au repository GitHub repérable à ce lien (fichier *main.py*).
- Le serveur est déployé sur Heroku et il est branché au même repository (fichier *main_backend.py*).

Tableau de bord

https://im612-p7-deploy-main-yo2f8r.streamlit.app

50%

RUNNING... Stop

Prêt à dépenser

Tableau de bord

Détail des crédits sollicités

Nombre de clients: 1000

Saisir le code client :

338564

Code client: 338564

Code client
338564

Prévision
Solvable

Probabilité de non solvabilité
0.22
↑ +0.68

Le crédit est accordé 👍

Le crédit est refusé si la probabilité de non solvabilité dépasse 0.90

Tableau de bord

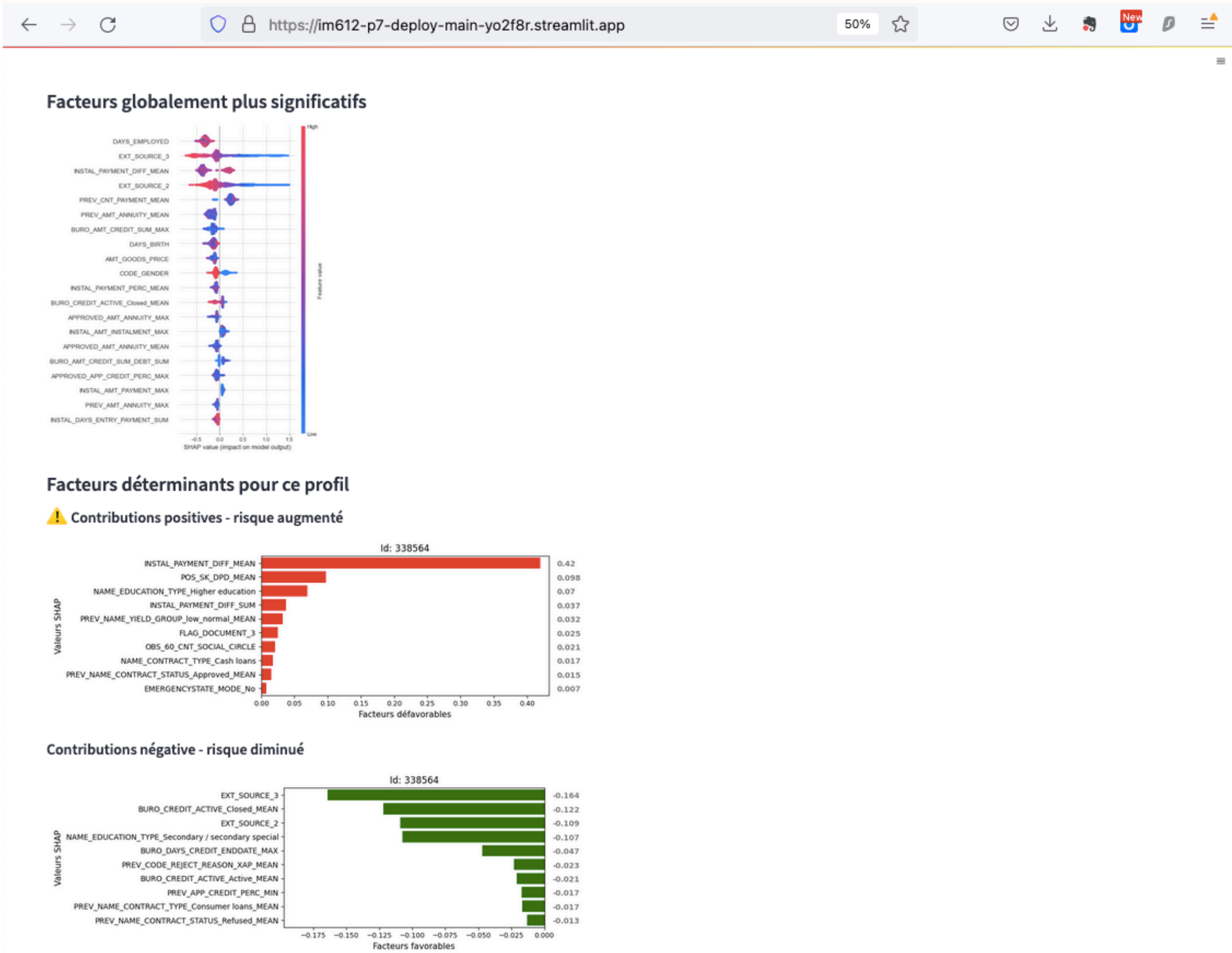
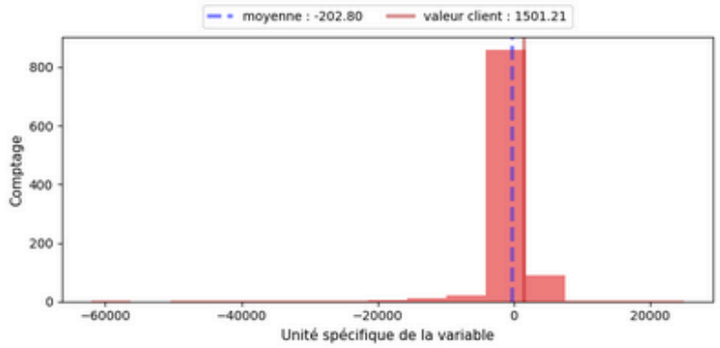


Tableau de bord

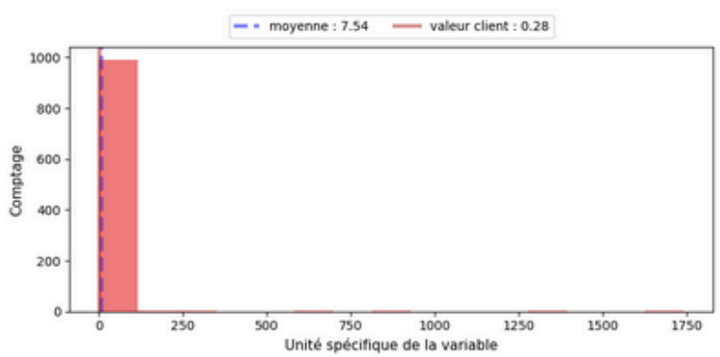
https://im612-p7-deploy-main-yo2f8r.streamlit.app

Distributions des facteurs défavorables pour le client

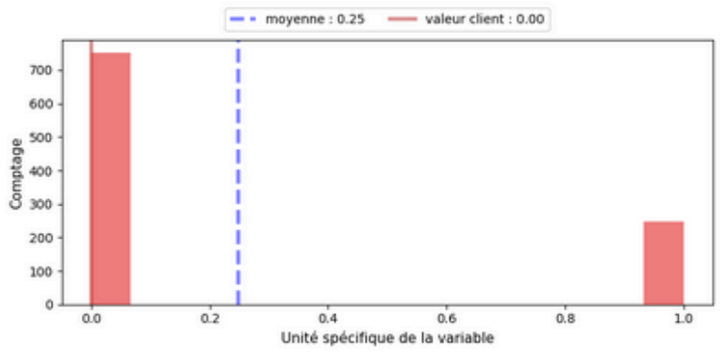
1 - variable INSTAL_PAYMENT_DIFF_MEAN



2 - variable POS_SK_DPD_MEAN



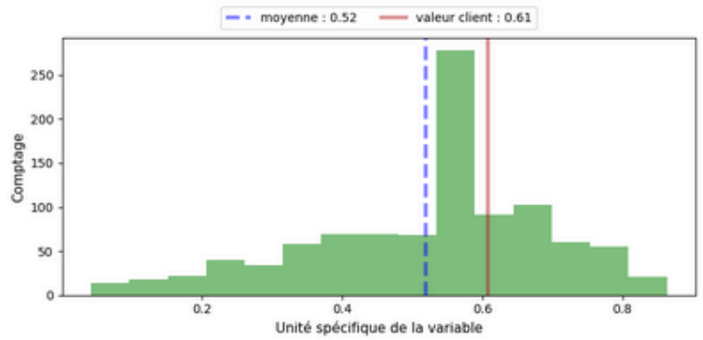
3 - variable NAME_EDUCATION_TYPE_Higher education



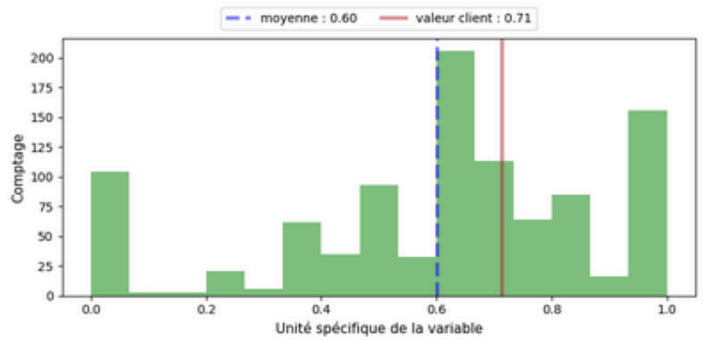
https://im612-p7-deploy-main-yo2f8r.streamlit.app

Distributions des facteurs favorables au client

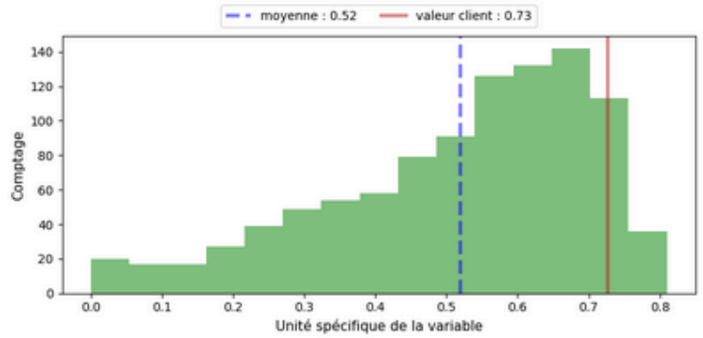
1 - variable EXT_SOURCE_3



2 - variable BURO_CREDIT_ACTIVE_Closed_MEAN



3 - variable EXT_SOURCE_2



7. Perspectives

Perspectives

- L’analyse du risque associé aux erreurs peut être rendue plus complexe (ex.: tranches de risque).
- Sur le tableau de bord, les noms des variables déterminantes ne sont pas clair.

Merci

Comparaison des modèles

Classificateur	Dataset	Performance sur le jeu d'entraînement					Performance sur le jeu de test					Performance calculée par validation croisée sur le jeu d'entraînement					Performance calculée par validation croisée sur le jeu de test équilibré par random undersampling				
		Accuracy	Recall	F_β ($\beta = 3$)	Roc auc	R^2	Accuracy	Recall	F_β ($\beta = 3$)	Roc auc	R^2	Accuracy	Recall	F_β ($\beta = 3$)	Roc auc	R^2	Accuracy	Recall	F_β ($\beta = 3$)	Roc auc	R^2
'Dummy '	'rus'	0,500	**0,000**	**0,000**	0,500	-1,000	0,919	**0,000**	**0,000**	0,500	-0,088	0,500	**0,000**	**0,000**	0,500	-1,000	0,500	**0,000**	**0,000**	0,500	-1,000
'LogReg '	'rus'	0,690	0,685	0,686	0,690	-0,242	0,688	0,691	0,521	0,689	-3,206	0,688	0,682	0,683	0,688	-0,250	0,686	0,680	0,681	0,686	-0,255
'GauNB '	'rus'	0,631	0,561	0,569	0,631	-0,475	0,687	0,564	0,434	0,631	-3,219	0,624	0,588	0,591	0,624	-0,503	0,630	0,610	0,611	0,630	-0,481
'SVM '	'rus'	0,690	0,685	0,686	0,690	-0,239	0,689	0,691	0,522	0,690	-3,187	0,689	0,683	0,684	0,689	-0,245	0,689	0,684	0,684	0,689	-0,244
'XGBClass '	'rus'	0,800	0,796	0,796	0,800	0,200	0,690	0,690	0,522	0,690	-3,174	0,686	0,681	0,682	0,686	-0,257	0,672	0,676	0,675	0,672	-0,313
'RandFC '	'rus'	1,000	1,000	1,000	1,000	1,000	0,682	0,683	0,513	0,683	-3,282	0,678	0,678	0,681	0,683	-0,277	0,672	0,683	0,685	0,678	-0,293
'RandFC '	'rus'	1,000	1,000	1,000	1,000	1,000	0,691	0,680	0,515	0,686	-3,168	0,686	0,672	0,675	0,685	-0,274	0,683	0,672	0,671	0,684	-0,280
'KNC '	'rus'	0,714	0,704	0,705	0,714	-0,144	0,616	0,597	0,428	0,607	-4,169	0,604	0,590	0,591	0,604	-0,583	0,597	0,575	0,578	0,597	-0,613
'HistGBC '	'rus'	0,758	0,755	0,756	0,758	0,034	0,696	0,704	0,535	0,700	-3,098	0,694	0,689	0,686	0,693	-0,221	0,680	0,683	0,683	0,680	-0,280
'Dummy '	'ccus'	0,500	**0,000**	**0,000**	0,500	-1,000	0,919	**0,000**	**0,000**	0,500	-0,088	0,500	**0,000**	**0,000**	0,500	-1,000	0,500	**0,000**	**0,000**	0,500	-1,000
'LogReg '	'ccus'	0,826	0,838	0,836	0,826	0,303	0,369	0,832	0,476	0,580	-7,497	0,822	0,836	0,834	0,822	0,289	0,666	0,662	0,663	0,666	-0,337
'GauNB '	'ccus'	0,824	0,862	0,856	0,824	0,297	**0,221**	0,853	0,441	0,509	-9,498	0,819	0,860	0,853	0,819	0,277	0,526	**0,087**	**0,094**	0,526	-0,894
'SVM '	'ccus'	0,832	0,847	0,845	0,832	0,327	0,353	0,839	0,474	0,574	-7,724	0,827	0,844	0,841	0,827	0,309	0,666	0,665	0,665	0,666	-0,335
'XGBClass '	'ccus'	1,000	1,000	1,000	1,000	1,000	**0,150**	0,938	0,460	0,509	-10,452	0,961	0,945	0,948	0,961	0,843	0,642	0,643	0,643	0,642	-0,431
'RandFC '	'ccus'	1,000	1,000	1,000	1,000	1,000	**0,163**	0,921	0,456	0,509	-10,275	0,952	0,929	0,933	0,952	0,805	0,646	0,651	0,651	0,651	-0,382
'RandFC '	'ccus'	1,000	1,000	1,000	1,000	1,000	**0,163**	0,923	0,457	0,510	-10,277	0,952	0,930	0,934	0,951	0,805	0,646	0,663	0,662	0,652	-0,409
'KNC '	'ccus'	0,921	0,913	0,915	0,921	0,684	**0,289**	0,823	0,446	0,532	-8,586	0,848	0,823	0,827	0,848	0,391	0,568	0,550	0,552	0,568	-0,728
'HistGBC '	'ccus'	0,980	0,965	0,968	0,980	0,919	**0,150**	0,940	0,461	0,511	-10,451	0,961	0,944	0,947	0,961	0,847	0,660	0,657	0,658	0,660	-0,362
'Dummy '	'smote'	0,500	**0,000**	**0,000**	0,500	-1,000	0,919	**0,000**	**0,000**	0,500	-0,088	0,500	**0,000**	**0,000**	0,500	-1,000	0,500	**0,000**	**0,000**	0,500	-1,000
'LogReg '	'smote'	0,678	0,688	0,687	0,678	-0,287	0,667	0,659	0,491	0,664	-3,484	0,678	0,688	0,686	0,678	-0,288	0,664	0,662	0,663	0,664	-0,345
'GauNB '	'smote'	0,664	0,661	0,661	0,664	-0,344	0,648	0,471	0,354	0,567	-3,741	0,663	0,659	0,660	0,663	-0,347	0,615	0,471	0,485	0,615	-0,540
'SVM '	'smote'	0,680	0,714	0,709	0,680	-0,278	0,651	0,682	0,498	0,665	-3,706	0,680	0,713	0,709	0,680	-0,280	0,666	0,666	0,666	0,666	-0,338
'XGBClass '	'smote'	0,962	0,925	0,932	0,962	0,847	0,916	**0,042**	**0,047**	0,518	-0,128	0,952	0,912	0,916	0,952	0,807	0,642	0,646	0,645	0,642	-0,430
'RandFC '	'smote'	1,000	1,000	1,000	1,000	1,000	0,913	**0,041**	**0,044**	0,515	-0,171	0,949	0,911	0,915	0,949	0,794	0,655	0,658	0,653	0,651	-0,388
'KNC '	'smote'	0,876	0,999	0,975	0,876	0,505	0,585	0,535	0,377	0,562	-4,593	0,781	0,996	0,955	0,781	0,124	0,567	0,581	0,580	0,567	-0,731
'HistGBC '	'smote'	0,958	0,918	0,925	0,958	0,833	0,918	**0,027**	**0,030**	0,512	-0,104	0,953	0,911	0,916	0,953	0,814	0,652	0,650	0,651	0,652	-0,391
'Dummy '	'orig'	0,919	**0,000**	**0,000**	0,500	-0,088	0,919	**0,000**	**0,000**	0,500	-0,088	0,919	**0,000**	**0,000**	0,500	-0,088	0,500	**0,000**	**0,000**	0,500	-1,000
'LogReg '	'orig'	0,919	**0,016**	**0,017**	0,507	-0,086	0,920	**0,014**	**0,016**	0,507	-0,084	0,919	**0,015**	**0,017**	0,507	-0,086	0,690	0,684	0,685	0,690	-0,239
'GauNB '	'orig'	0,574	0,695	0,474	0,629	-4,742	0,576	0,707	0,482	0,635	-4,720	0,596	0,672	0,468	0,631	-4,450	0,636	0,615	0,616	0,636	-0,458
'SVM '	'orig'	0,919	**0,000**	**0,000**	0,500	-0,088	0,919	**0,000**	**0,000**	0,500	-0,088	0,919	**0,000**	**0,000**	0,500	-0,088	0,691	0,686	0,687	0,691	-0,236
'XGBClass '	'orig'	0,934	**0,196**	**0,213**	0,597	0,109	0,918	**0,056**	**0,062**	0,525	-0,102	0,918	**0,055**	**0,060**	0,524	-0,110	0,674	0,675	0,675	0,674	-0,306
'RandFC '	'orig'	0,986	0,823	0,838	0,911	0,807	0,919	**0,015**	**0,017**	0,507	-0,098	0,918	**0,015**	**0,016**	0,507	-0,101	0,626	0,553	0,560	0,630	-0,484
'KNC '	'orig'	0,930	**0,217**	**0,233**	0,605	0,057	0,904	**0,047**	**0,051**	0,513	-0,288	0,905	**0,046**	**0,050**	0,513	-0,283	0,576	0,561	0,563	0,576	-0,696
'HistGBC '	'orig'	0,921	**0,043**	**0,047**	0,521	-0,058	0,920	**0,031**	**0,034**	0,514	-0,078	0,919	**0,028**	**0,031**	0,513	-0,084	0,682	0,684	0,684	0,682	-0,274

Classificateurs testés

Nom du classificateur abrégé	Classificateur	Paramètres explorés
Dummy	DummyClassifier()	{'dummyclassifier__strategy' : ['most_frequent']}))
LogReg	LogisticRegression(random_state=0, max_iter=10000)	{'logisticregression__solver' : ['newton-cg', 'lbfgs', 'liblinear']}))
GauNB	GaussianNB()	{'gaussiannb__var_smoothing' : [1e-9, 1e-10]}))
SVM	LinearSVC(max_iter=100000)	{'linearsvc__loss' : ['hinge', 'squared_hinge']}))
XGBClass	XGBClassifier()	{'xgbclassifier__booster' : ['gbtree', 'gblinear', ‘dart'], 'xgbclassifier__n_estimators': [200, 300, 400, 500]}))
RandFC	RandomForestClassifier(),	{'randomforestclassifier__n_estimators' : [10, 25, 30, 50, 75, 100], 'randomforestclassifier__criterion' : ['gini', 'entropy']}))
KNC	KNeighborsClassifier()	{'kneighborsclassifier__n_neighbors': [3, 5, 7, 10]}))
HistGBC	HistGradientBoostingClassifier()	{'histgradientboostingclassifier__max_iter': [100, 200, 500], 'histgradientboostingclassifier__max_leaf_nodes': [20, 40]}))

Plus de paramètres du modèle retenu

```
name: HistGBC  , model: HistGradientBoostingClassifier(), parameters:
{'histgradientboostingclassifier__max_iter': [100, 200, 500],
'histgradientboostingclassifier__max_leaf_nodes': [20, 40]}
Thu_Mar_23_00:35:11_2023
Fitting 5 folds for each of 6 candidates, totalling 30 fits
best_index_: 3,
best_score_: 0.690986908358509 6,
best_params_: {'histgradientboostingclassifier__max_iter': 200,
'histgradientboostingclassifier__max_leaf_nodes': 40}
Testing performance
finished balancing x_test
X_train (39720, 100)
y_train (39720,)
X_test (61502, 100)
y_test (61502,)
X_test_bal (9930, 100)
y_test_bal (9930,)
CF: [[15129  4731]
 [ 4863 14997]]
CF: [[39303 17234]
 [ 1468  3497]]
### ['HistGBC  ', 'rus', (0.7584592145015105, 0.7551359516616314, 0.7556381885240945,
0.7584592145015105, 0.033836858006042 3), (0.695912328054372 2, 0.704330312185297 1,
0.534578696343402 2, 0.699751692343245 5, -3.097559596162114 7), [0.693605236656596 1,
0.689123867069486 4, 0.685793937608695 4, 0.693227593152064 4, -0.220644511581067 5,
0.680060422960725, 0.683383685800604 3, 0.682914771631994 2, 0.680060422960725,
-0.279758308157099 7]]
```

Modèle retenu
Performances

Accuracy (train set)	0,758	
Recall (train set)	0,755	
F_β (β = 3) (train set)	0,756	
Roc auc (train set)	0,758	
R^2 (train set)	0,034	
Accuracy (test set)	0,696	1ère
Recall (test set)	0,704	2ème
F_β (β = 3) (test set)	0,535	1ère
Roc auc (test set)	0,700	1ère
R^2 (test set)	-3,098	1ère
Accuracy CV (train set)	0,694	
Recall CV (train set)	0,689	
F_β (β = 3) CV (train set)	0,686	
Roc auc CV (train set)	0,693	
R^2 CV (train set)	-0,221	
Accuracy CV (rus-balanced test set)	0,680	
Recall CV (rus-balanced test set)	0,683	
F_β (β = 3) CV (rus-balanced test set)	0,683	
Roc auc CV (rus-balanced test set)	0,680	
R^2 CV (rus-balanced test set)	-0,280	

Déterminer la qualité de la classification

- Accuracy: fraction des prévisions correctes.
- Recall: fractions de VP sur le total des prévisions incorrectes.
- $F_{\beta(3)}$: combinaison de précision et recall (voir formule)
- Roc auc
- R^2

https://scikit-learn.org/stable/modules/model_evaluation.html#binary-classification

Binary classification

In a binary classification task, the terms ''positive'' and ''negative'' refer to the classifier's prediction, and the terms ''true'' and ''false'' refer to whether that prediction corresponds to the external judgment (sometimes known as the ''observation''). Given these definitions, we can formulate the following table:

	Actual class (observation)	
Predicted class (expectation)	tp (true positive) Correct result	fp (false positive) Unexpected result
	fn (false negative) Missing result	tn (true negative) Correct absence of result

In this context, we can define the notions of precision, recall and F-measure:

Colonnes collineaires

Frank E. Harrell, Jr.

Regression Modeling Strategies

With Applications to Linear Models, Logistic Regression, and Survival Analysis

With 141 Figures

Springer

4.6 Collinearity

When at least one of the predictors can be predicted well from the other predictors, the standard errors of the regression coefficient estimates can be inflated and corresponding tests have reduced power.[149] In stepwise variable selection, collinearity can cause predictors to compete and make the selection of "important" variables arbitrary. Collinearity makes it difficult to estimate and interpret a particular regression coefficient because the data have little information about the effect of changing

one variable while holding another (highly correlated) variable constant [70, p. 173]. However, collinearity does not affect the joint influence of highly correlated variables when tested simultaneously. Therefore, once groups of highly correlated predictors are identified, the problem can be rectified by testing the contribution of an entire set with a multiple d.f. test rather than attempting to interpret the coefficient or one d.f. test for a single predictor.

Collinearity does not affect predictions made on the same dataset used to estimate the model parameters or on new data that have the same degree of collinearity as the original data [321, pp. 379–381] as long as extreme extrapolation is not attempted. Consider as two predictors the total and LDL cholesterols that are highly correlated. If predictions are made at the same combinations of total and LDL cholesterol that occurred in the training data, no problem will arise. However, if one makes a prediction at an inconsistent combination of these two variables, the predictions may be inaccurate and have high standard errors.

When the ordinary truncated power basis is used to derive component variables for fitting linear and cubic splines, as was described earlier, the component variables can be very collinear. It is very unlikely that this will result in any problems, however, as the component variables are connected algebraically. Thus it is not possible for a combination of, for example, x and $\max(x-10,0)$ to be inconsistent with each other. Collinearity problems are then more likely to result from partially redundant subsets of predictors as in the cholesterol example above.

Colonnes collineaires

Frank E. Harrell, Jr.

Regression Modeling Strategies

With Applications to
Linear Models,
Logistic Regression,
and Survival Analysis

With 141 Figures

Springer

One way to quantify collinearity is with *variance inflation factors* or *VIF*, which in ordinary least squares are diagonals of the inverse of the $X'X$ matrix scaled to have unit variance (except that a column of 1s is retained corresponding to the intercept). Note that some authors compute VIF from the correlation matrix form of the design matrix, omitting the intercept. VIF_i is $1/(1 - R_i^2)$ where R_i^2 is the squared multiple correlation coefficient between column i and the remaining columns of the design matrix. For models that are fitted with maximum likelihood estimation, the information matrix is scaled to correlation form, and VIF is the diagonal of the inverse of this scaled matrix.[106, 446] Then the VIF are similar to those from a weighted correlation matrix of the original columns in the design matrix. Note that indexes such as VIF are not very informative as some variables are algebraically connected to each other.

The SAS `VARCLUS` procedure[362] and S-PLUS `varclus` function can identify collinear predictors. Summarizing collinear variables using a summary score is more powerful and stable than arbitrary selection of one variable in a group of collinear variables (see the next section).