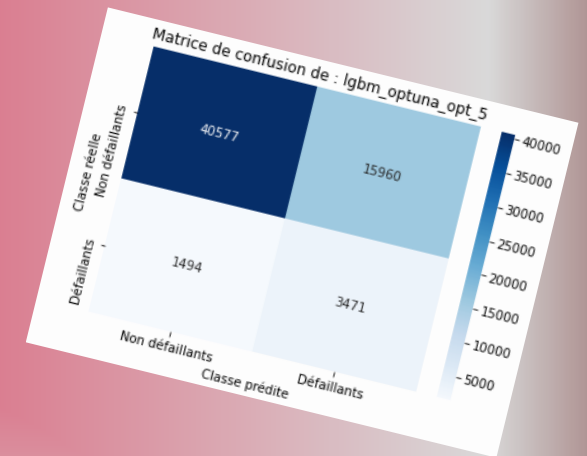
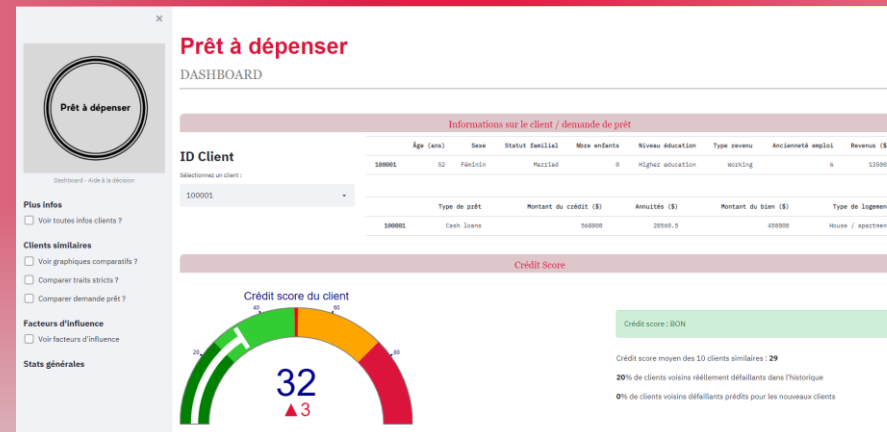
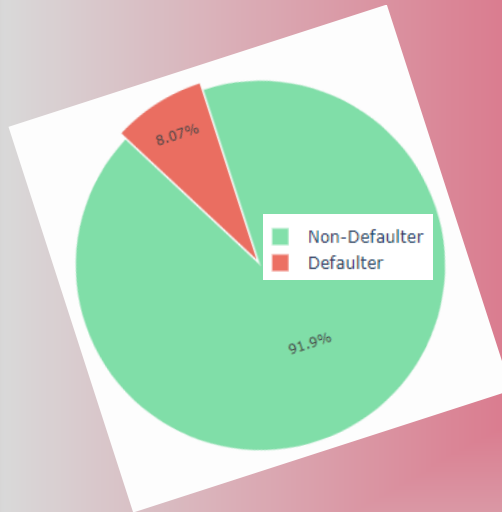
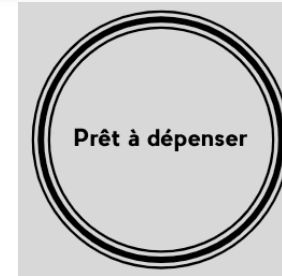


# Implémentez un modèle de scoring



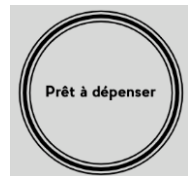
- ① Problématique
- ② Données
- ③ Modélisation
- ④ Dashboard
- ⑤ Conclusions

- ① **Problématique**
- ② Données
- ③ Modélisation
- ④ Dashboard
- ⑤ Conclusions

# 1 Problématique - présentation



## Prêt à dépenser :



Société financière d'offre de crédit à la consommation pour la clientèle ayant peu ou pas d'historique de prêt.

## Mission :

Construire un **modèle de scoring** prédisant automatiquement la **probabilité** de **défaut de paiement** d'un client.

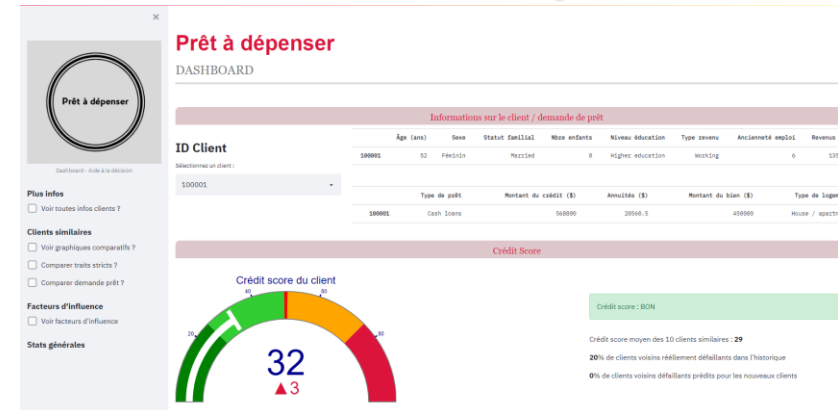
Développer un **dashboard interactif**.

## Objectifs :

**Étayer** la décision d'accorder ou non un prêt.

Améliorer la relation avec le client en faisant preuve de **transparence**.

Montrer au client les **informations** le concernant grâce à **l'interactivité**.

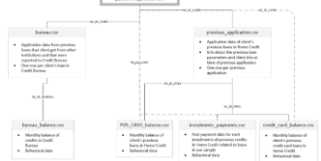


# 1 Problématique - processus

Prêt à dépenser

EDA

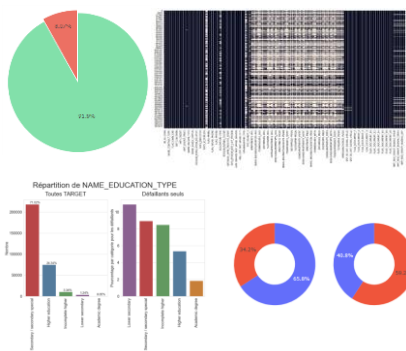
[Kaggle 8 fichiers](#)



[Kernel Rishabhrao](#)



Stats, types données, nan, unique



PRÉ  
PROCESSING

Nettoyage

- Type de données (bool, mémoire..)
- Val. aberrantes
- Imputation

Feature  
Engineering

- Création variables métiers
- Création variables automatiques (min, max, mean, sum, count..)
- Encodage
- Suppression des colinéarités fortes
- Assemblage (merge)



FEATURES  
SELECTION

LightGbm

- Plusieurs itérations
- Features importances

Boruta

BorutaShap

Permutation  
importance

- Sklearn, eli5

RFECV

→ Conserve les variables les plus répétées pour toutes ces méthodes

MODELISATION

FXCARET

- Première idée
- Choix du jeu de données
- Choix du modèle

LightGbm

- Split du jeu de données, rééquilibrage
- Choix des métriques
- Optimisation du modèles
- Seuil de probabilité optimal

Modèle Final

DASHBOARD

Dév. local



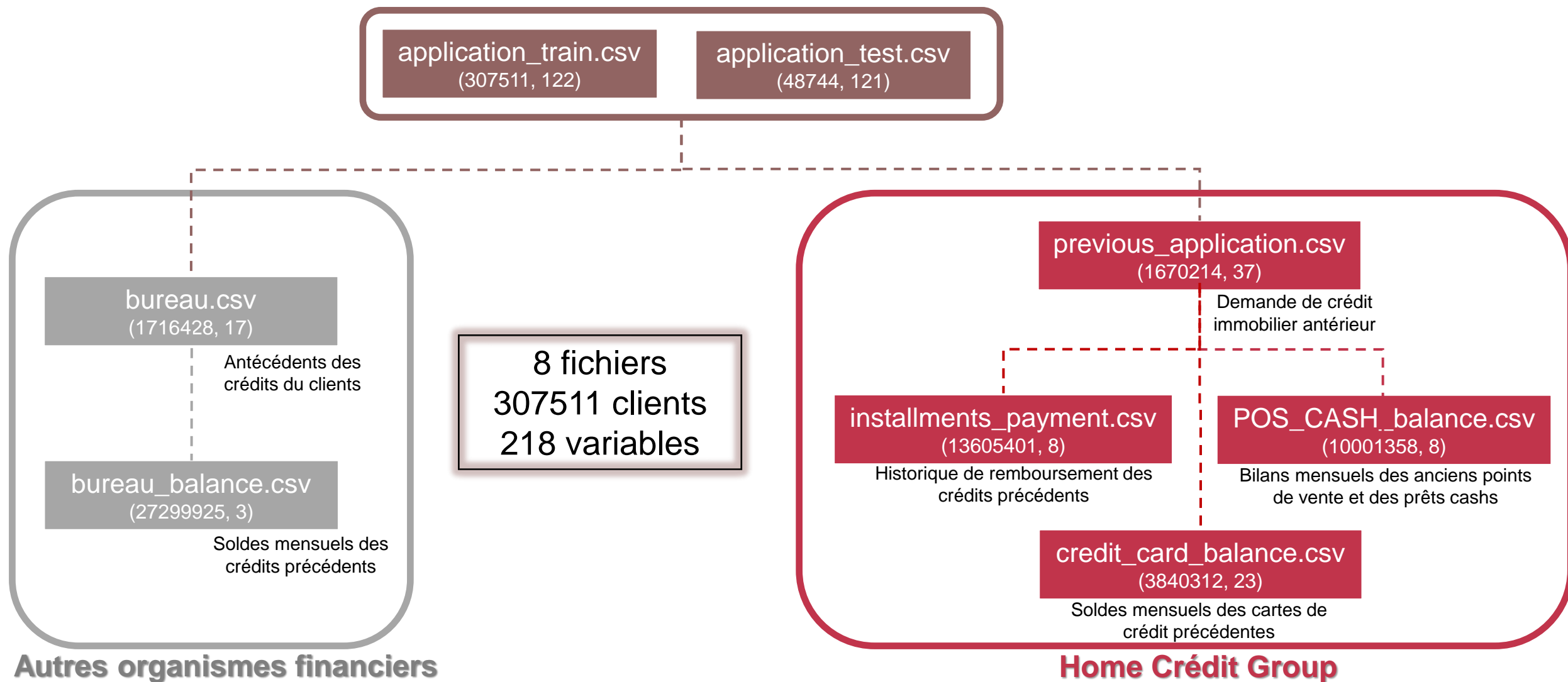
- Prédictions
- Visualisations

Déploiement

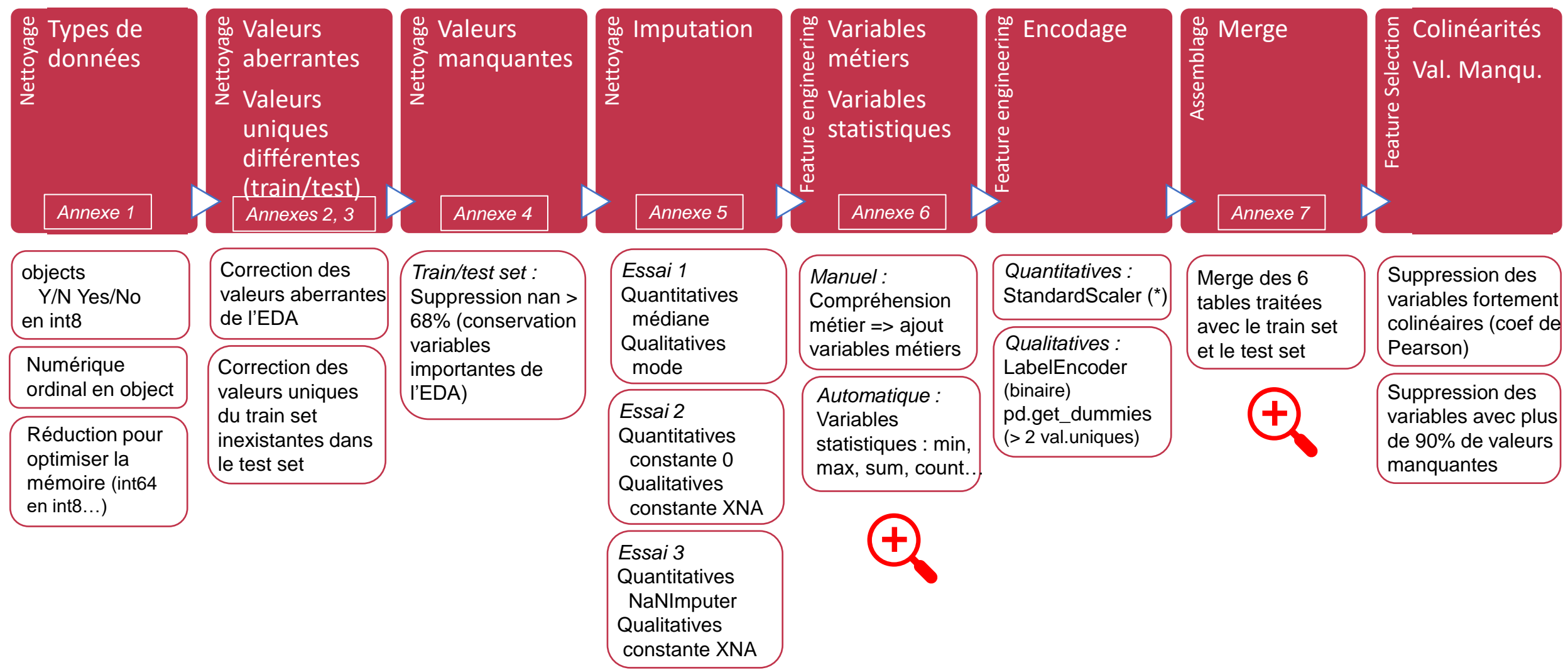
Streamlit

- ① Problématique
- ② **Données**
- ③ Modélisation
- ④ Dashboard
- ⑤ Conclusions

# 2 Données – Jeux de données



# 2 Données – Pré-processing pour les 8 fichiers





## Automatique

- Création de variables statistiques : quantitatives : min, max, sum, mean, var
- Création de variables statistiques : qualitatives : sum, count, mean

## Manuel

- revenu, de rente et de crédit : ratio/différence
- Jours en années, changement de jours : ratio
- Âge de la voiture, ancienneté d'emploi : ratio/différence
- Flag sur les téléphones : ratio/différence
- Membres de la famille : ratio/différence
- Note de la région où vit le client : ratio/différence
- Données externes : ratio, moyenne, max, min
- Informations sur le bâtiment : somme, multiplication
- Défauts de paiements et les défauts observables : somme/ratio
- Flag sur les documents : somme, moyenne, variance, écart-type
- Modification du demandeur : somme/ratio

# ② Données – Assemblage – Merge



Dataframe initial	Nbr lignes var. initiales	Nbr lignes var. après FE	Merge avec application_train/test et suppr var. colinéaires + > 90% nan
application_train/test	(307511, <b>122</b> ) (48744, <b>121</b> )	(307507, 206) (48744, 205)	
credit_card_balance	(3840312, 23)	agg_ccb_cat (103558, 21) agg_ccb_num (103558, 68)	(307507, 246) (48744, 245)
installments_payments	(13605401, 8)	agg_pay_num (339587, 30)	(307507, 265) (48744, 264)
POS_CASH_balance	(10001358, 10)	agg_pos_num (337252, 27)	(307507, 285) (48744, 284)
previous_application	(1670214, 37)	agg_prev_num (338857, 114)	(307507, 552) (48744, 551)
bureau_balance	(27299925, 3)	agg_bureau_balance_par_demandeur (305811, 12)	(307507, 555) (48744, 554)
bureau	(1716428, 17)	agg_bureau_num (305811, 60)	(307507, <b>615</b> ) (48744, <b>614</b> )



**train set : 615 variables**  
**test set : 614 variables**



**FEATURE SELECTION  
NÉCESSAIRE**



**+ 493**

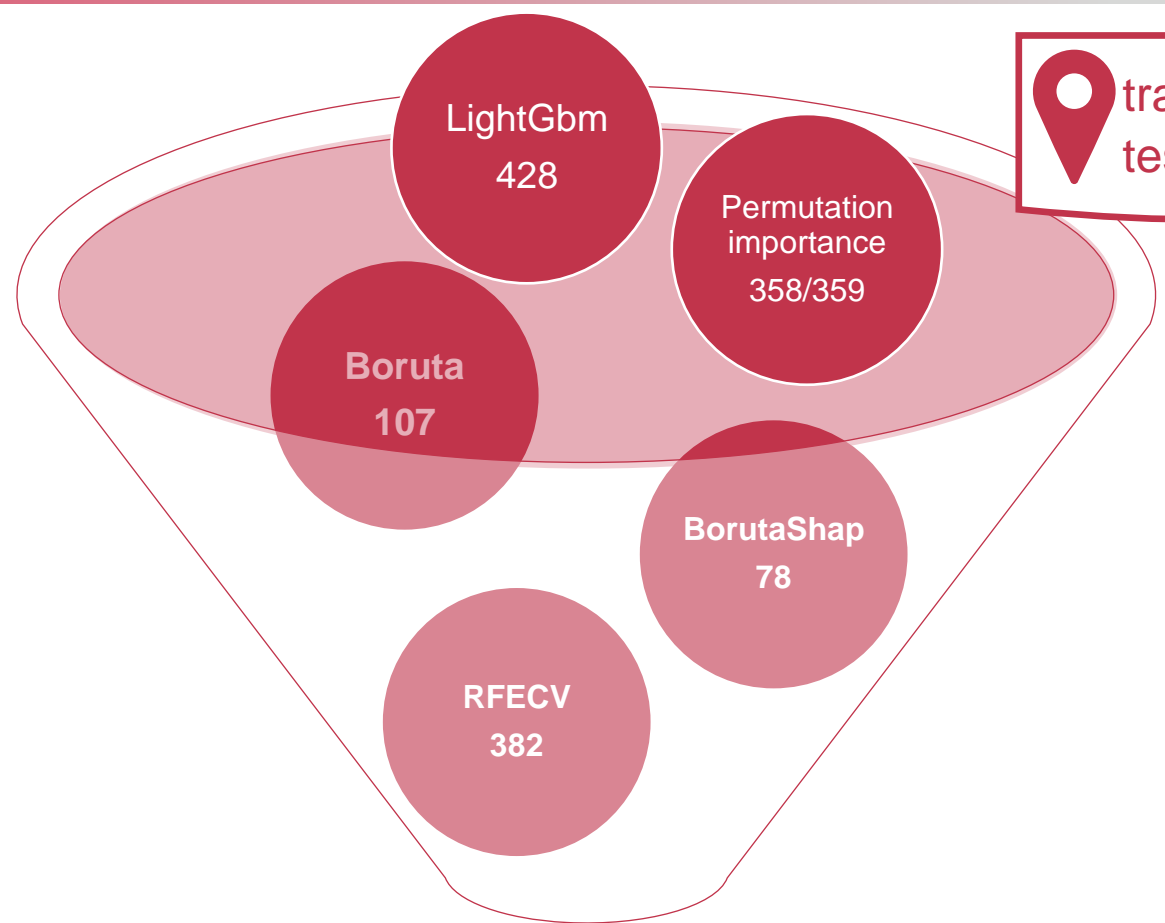
# 2 Données – Features Selection



Nbr répétition	Nbr variables	%_variables/615
(0, 1]	99	19.9195
(1, 2]	53	10.6640
(2, 3]	57	11.4688
(3, 4]	182	36.6197

(4, 5]	28	5.6338
(5, 6]	78	15.6942

+ IDENTIFIANT  
+ TARGET

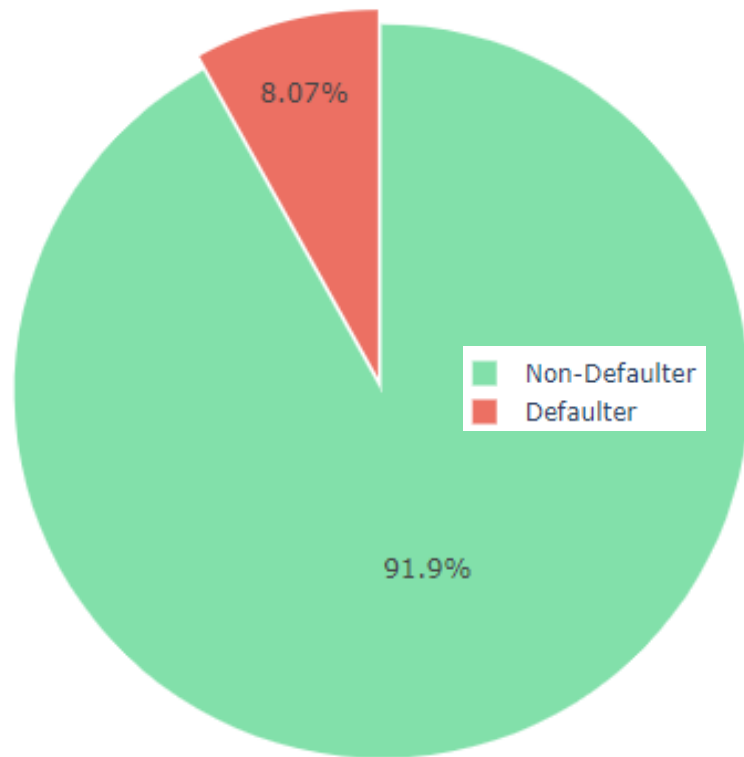


train set : **615** variables  
test set : 614 variables

train set : **108** variables  
test set : 107 variables

**DONNÉES PRÊTES POUR MODELISATION**

- ① Problématique
- ② Données
- ③ Modélisation**
- ④ Dashboard
- ⑤ Conclusions



## Variable cible binaire

Classe 0

Clients **non défaillants**

92% majoritaire

Classe 1

Clients **défaillants**

8% minoritaire

## Modélisation

**Classification binaire**

Avec classes  
**déséquilibrées**



# 3 Modélisation – Choix des métriques



## Rappel :

Classe 0 négative = non défaillant

Classe 1 positive = défaillant

## Matrice de confusion :

Classe réelle	+	<b>TP</b> Vrais positifs	<b>FN</b> Faux négatifs
	-	<b>FP</b> Faux positifs	<b>TN</b> Vrais négatifs
		+	-
		Classe prédite	

## Minimiser les pertes argents :

Prédiction	Réalité	PERTE
+ défaillant	- non-défaillant	Intérêt du prêt non accordé
- non-défaillant	+ défaillant	Somme empruntée en partie ou totale

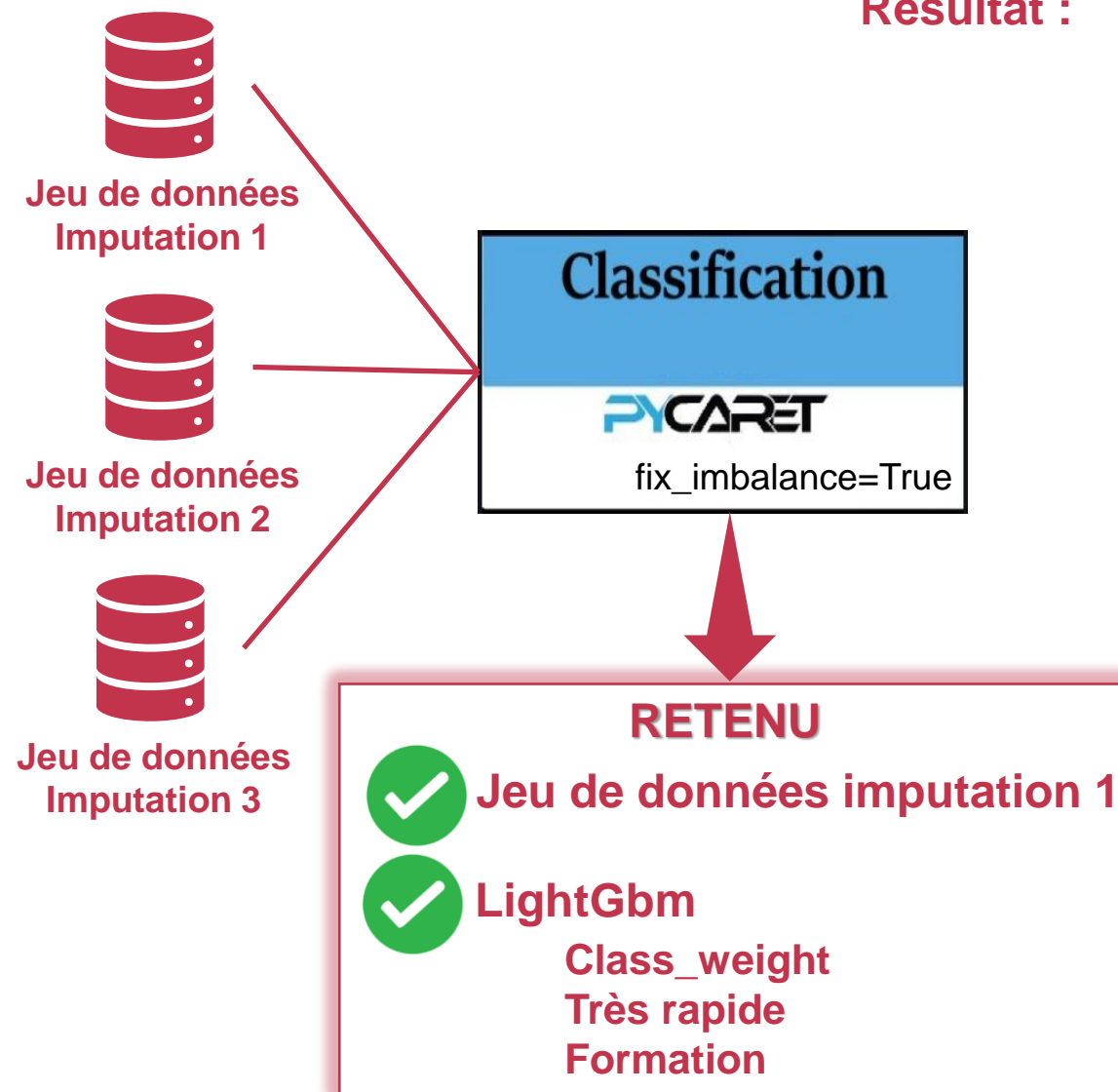
- ➔ **minimiser** le nombre de **faux positifs**  
**maximiser** la métrique **Précision**
- ➔ **minimiser** le nombre de **faux négatifs**  
**maximiser** les métriques **Recall** ou **Fbeta 10**

**Perte > pour FN que FP : privilégier Recall/F10**  
**Compromis FN/FP** (si FN ↗ FP ↘ et vis versa)  
**Tester métrique métier / fonction coût** pour jouer sur le taux de FN/FP et privilégier les bons prêts

# 3 Modélisation – Pycaret Première idée



Résultat :

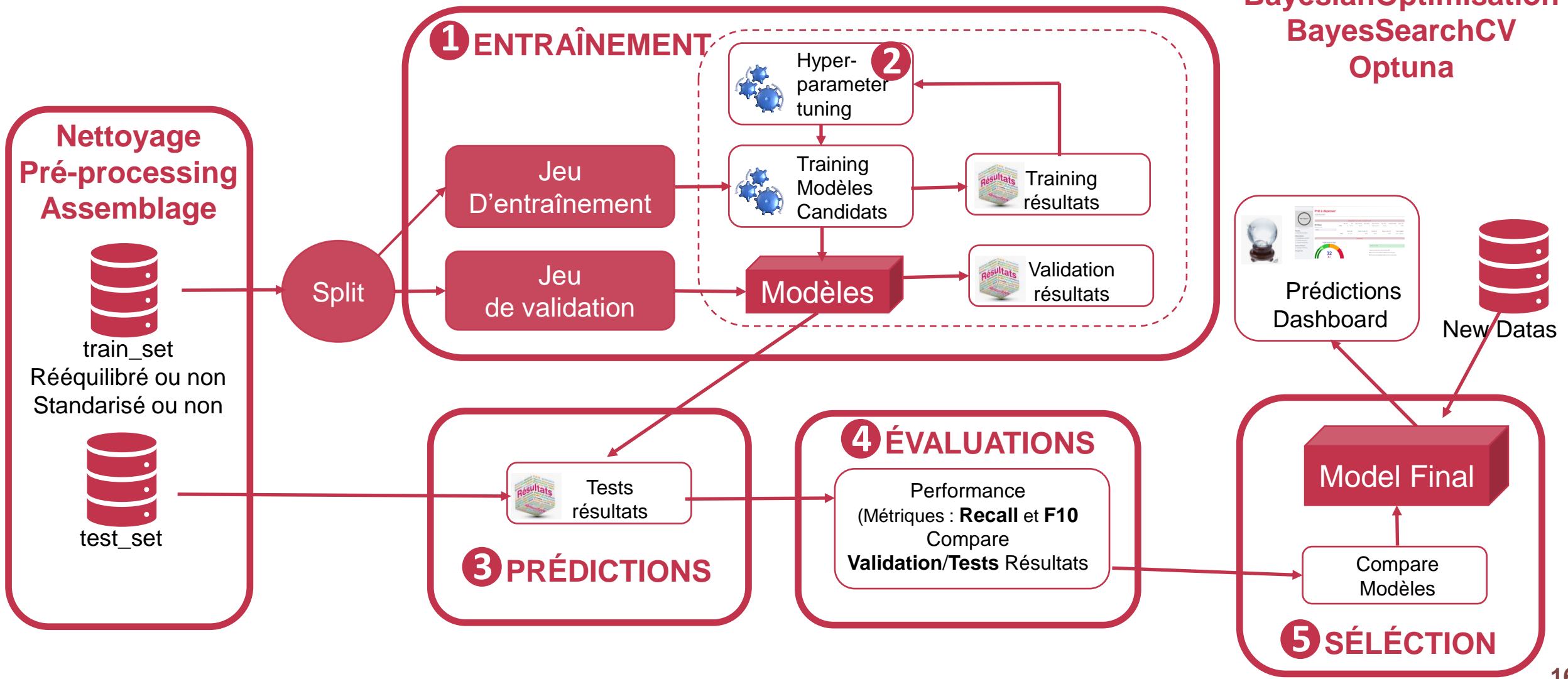


	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.9179	0.7685	0.0683	0.4452	0.1184	0.0991	0.1498	65.995
xgboost	Extreme Gradient Boosting	0.9158	0.7562	0.0800	0.3930	0.1329	0.1086	0.1481	51.939
lightgbm	Light Gradient Boosting Machine	0.9162	0.7550	0.0503	0.3640	0.0883	0.0701	0.1104	10.222
rf	Random Forest Classifier	0.9124	0.7342	0.0586	0.2897	0.0974	0.0722	0.0987	49.923
gbc	Gradient Boosting Classifier	0.9019	0.7168	0.1046	0.2466	0.1468	0.1037	0.1146	125.717
et	Extra Trees Classifier	0.9018	0.7124	0.0947	0.2331	0.1347	0.0924	0.1030	42.072
ada	Ada Boost Classifier	0.8723	0.6979	0.1959	0.2010	0.1984	0.1290	0.1291	32.810
lda	Linear Discriminant Analysis	0.7316	0.6739	0.4904	0.1484	0.2278	0.1185	0.1498	7.882
knn	K Neighbors Classifier	0.6839	0.5661	0.3829	0.1040	0.1636	0.0419	0.0556	116.716
nb	Naive Bayes	0.1113	0.5592	0.9762	0.0816	0.1506	0.0019	0.0174	5.304
qda	Quadratic Discriminant Analysis	0.1459	0.5553	0.9502	0.0828	0.1523	0.0044	0.0266	10.489
dt	Decision Tree Classifier	0.8286	0.5473	0.2119	0.1370	0.1664	0.0758	0.0780	10.933
lr	Logistic Regression	0.6327	0.4720	0.1981	0.0948	0.1270	0.0455	0.0500	22.488
svm	SVM - Linear Kernel	0.5116	0.0000	0.5890	0.0956	0.1613	0.0286	0.0541	9.513
ridge	Ridge Classifier	0.7315	0.0000	0.4898	0.1482	0.2275	0.1182	0.1493	5.319

OPTIMISATION

## 2 OPTIMISATION

BayesianOptimisation  
BayesSearchCV  
Optuna





# 3 Modélisation – Rééquilibrage des classes

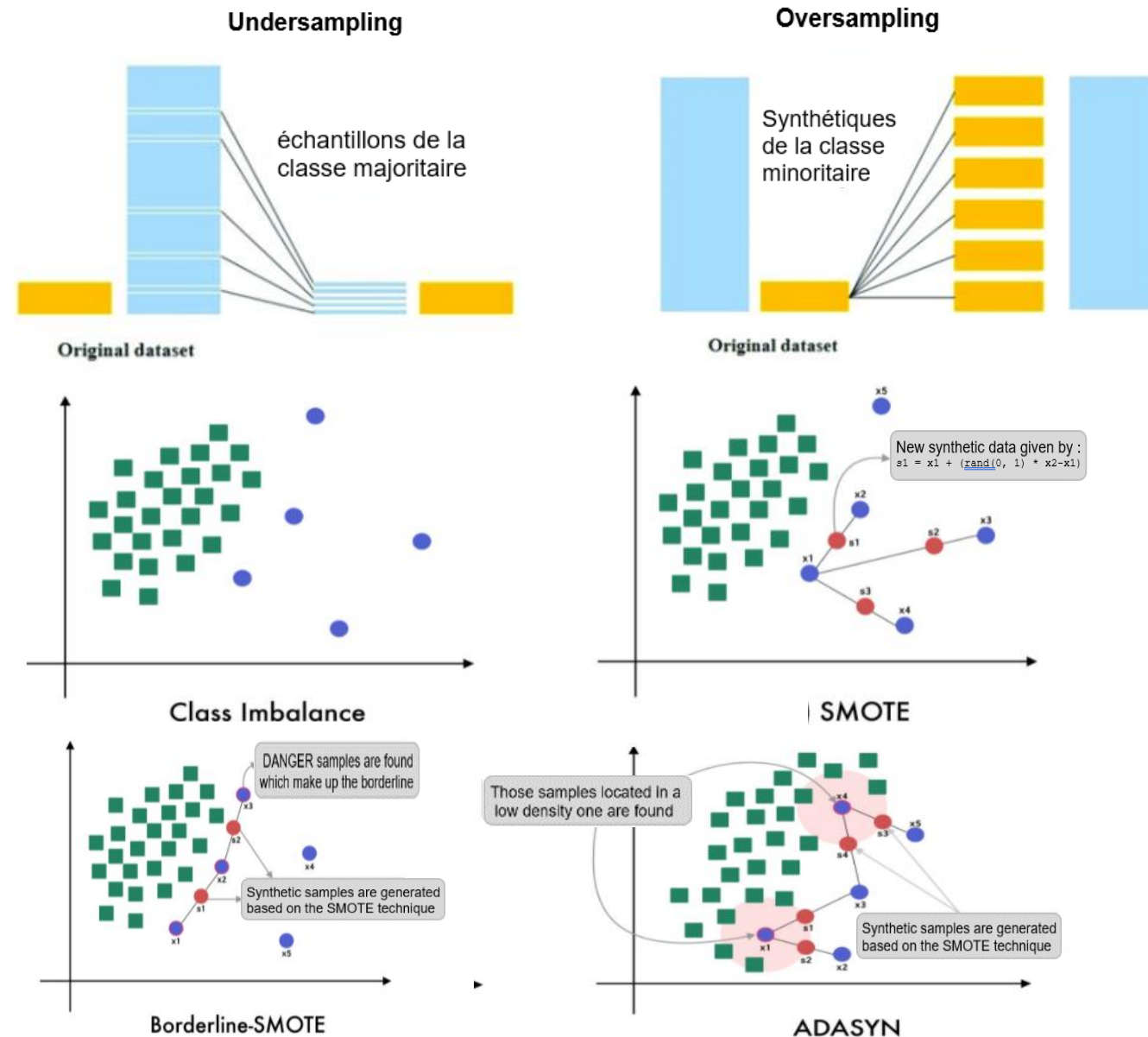
## Techniques testées

LightGBM class\_weight

UNDERSAMPLING  
SMOTE

OVERSAMPLING  
SMOTE, BORDERLINESMOTE,  
ADASYN

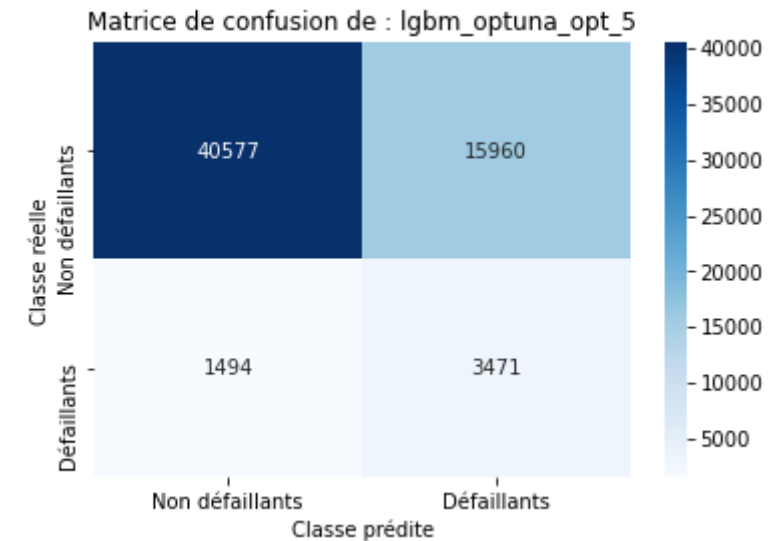
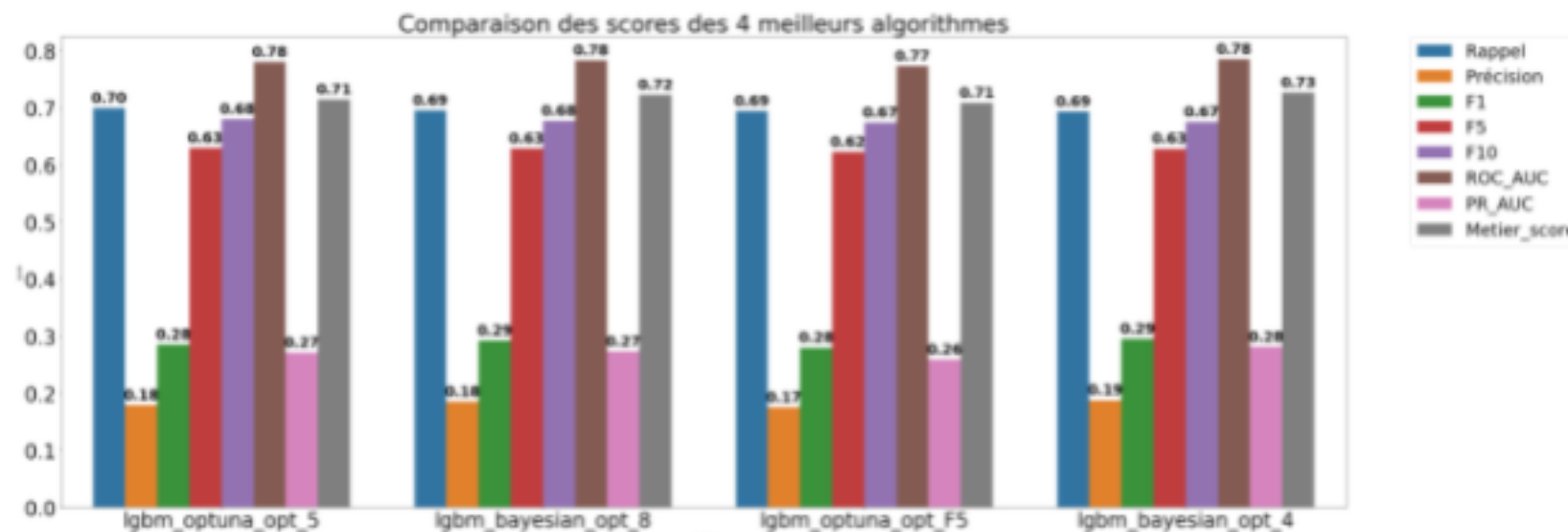
OVERSAMPLING suivi  
par UNDERSAMPLING  
SMOTE



# 3 Modélisation – Bilan 4 meilleurs modèles



Modèle	Jeu_donnees	FN	FP	TP	TN	Metrique	Optimisation	Class_weight	Rappel	Précision	F1	F5	F10	ROC_AUC	PR_AUC	Metier_score	D
lgbm_optuna_opt_5	train	1494	15960	3471	40577	F10	optuna	oui	0.6991	0.1786	0.2846	0.6286	0.6795	0.7795	0.2702	0.7135	
lgbm_bayesian_opt_8	train	1515	15278	3450	41259	F10	bayes_opt	oui	0.6949	0.1842	0.2912	0.6279	0.6763	0.7826	0.2728	0.7219	
lgbm_optuna_opt_F5	train	1522	16272	3443	40265	F5	optuna	non	0.6935	0.1746	0.2790	0.6223	0.6736	0.7727	0.2579	0.7079	
lgbm_bayesian_opt_4	train	1526	14937	3439	41600	roc_auc	bayes_opt	oui	0.6926	0.1871	0.2947	0.6275	0.6746	0.7843	0.2801	0.7260	



**RETENU pour Dashboard**



**lgbm\_otuna\_opt\_5**

# 3 Modélisation – Modèle optimisé



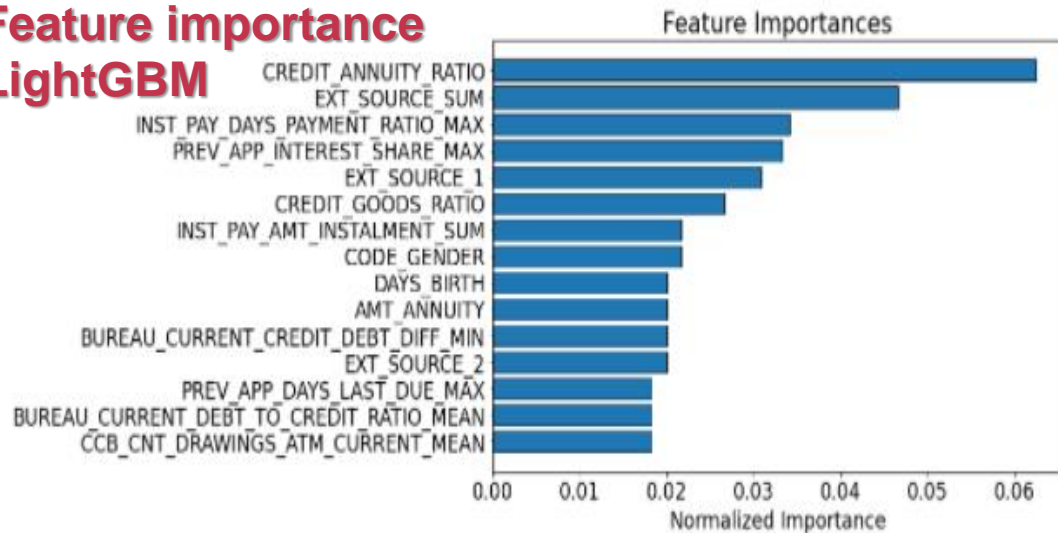
Hyperparamètre	Défaut	Optimisé
n_estimators	100	<b>100</b>
learning_rate	0,1	<b>0,1</b>
objective	None	<b>binary</b>
boosting_type	gbdt	<b>gbdt</b>
class_weight	None	<b>balanced</b>
colsample_bytree	1	<b>0.65731418761953</b>
max_depth	-1	<b>9</b>
min_child_samples	20	<b>96</b>
min_child_weight	0,001	<b>0.5685528790757488</b>
num_leaves	31	<b>21</b>
reg_alpha	0	<b>1.7033609851586964e-06,</b>
reg_lambda	0	<b>0.012745771755334187</b>
subsample	1	<b>0.8190208924749053</b>
subsample_freq	0	<b>1</b>

# 3 Modélisation - Interprétabilité

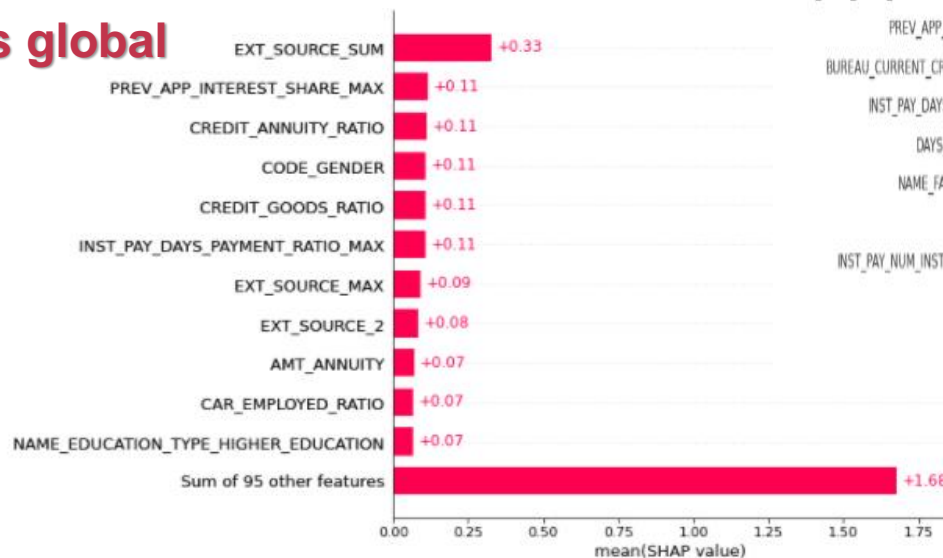


## GLOBAL

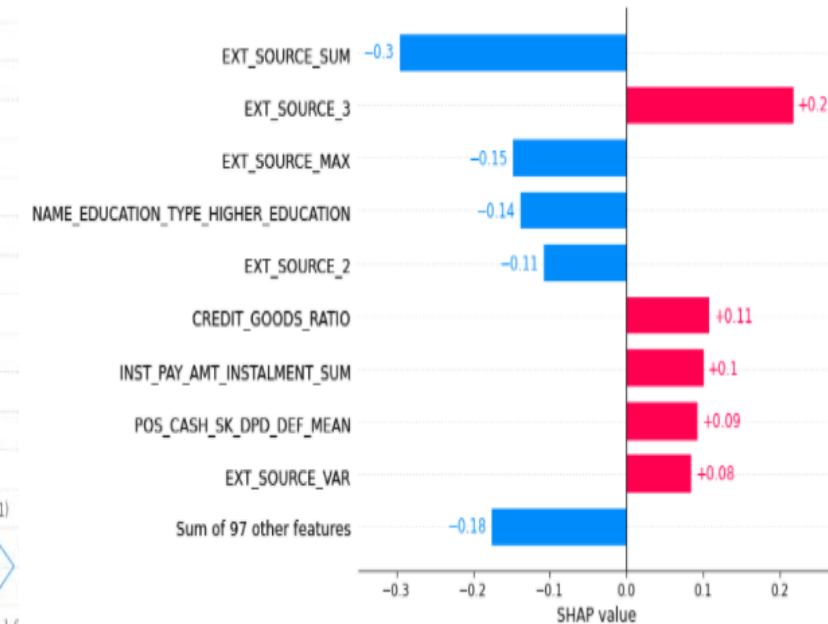
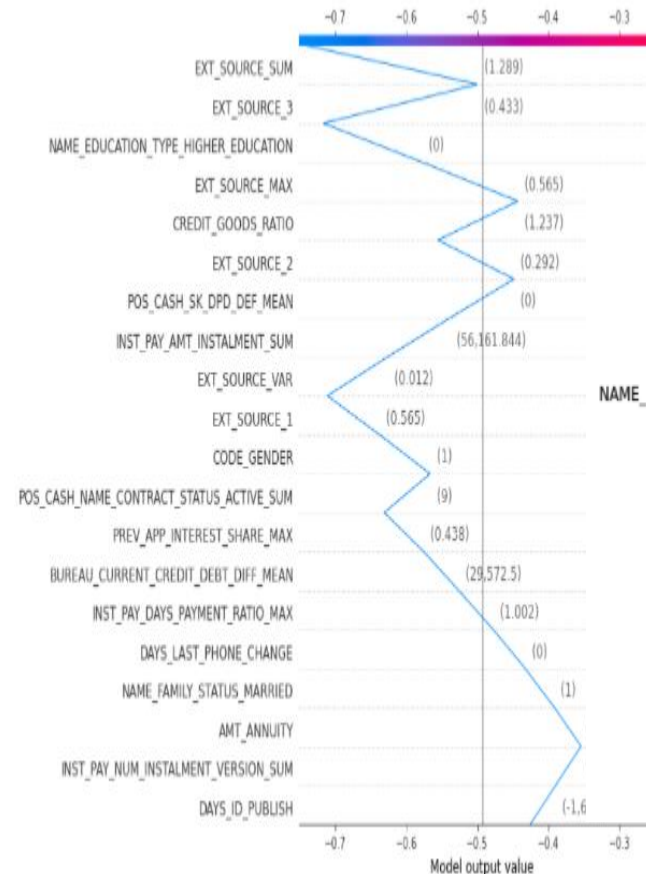
### Feature importance LightGBM



### Shap values global



## POUR UN CLIENT



- ① Problématique
- ② Données
- ③ Modélisation
- ④ Dashboard**
- ⑤ Conclusions

Prêt à dépenser

Dashboard - Aide à la décision

Plus infos

☐ Voir toutes infos clients ?

Clients similaires

☐ Voir graphiques comparatifs ?  
☐ Comparer traits stricts ?  
☐ Comparer demande prêt ?

Facteurs d'influence

☐ Voir facteurs d'influence

Stats générales

☐ Voir les distributions

## Prêt à dépenser

### DASHBOARD

#### Informations sur le client / demande de prêt

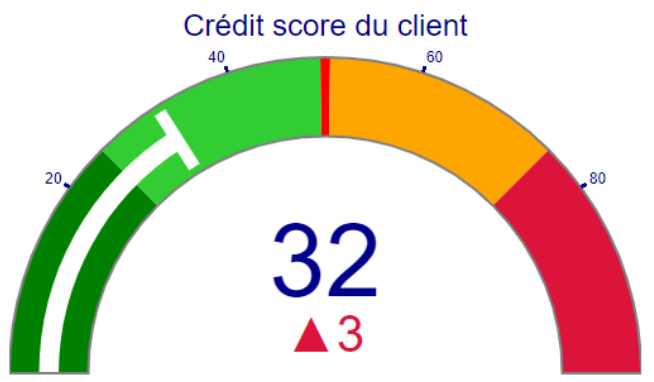
#### ID Client

Sélectionnez un client :

100001

	Âge (ans)	Sexe	Statut familial	Nbre enfants	Niveau éducation	Type revenu	Ancienneté emploi	Revenus (\$)	
100001	52	Féminin	Married	0	Higher education	Working	6	135000	
	Type de prêt		Montant du crédit (\$)		Annuités (\$)		Montant du bien (\$)		Type de logement
100001	Cash loans		568800		20560.5		450000		House / apartment

#### Crédit Score

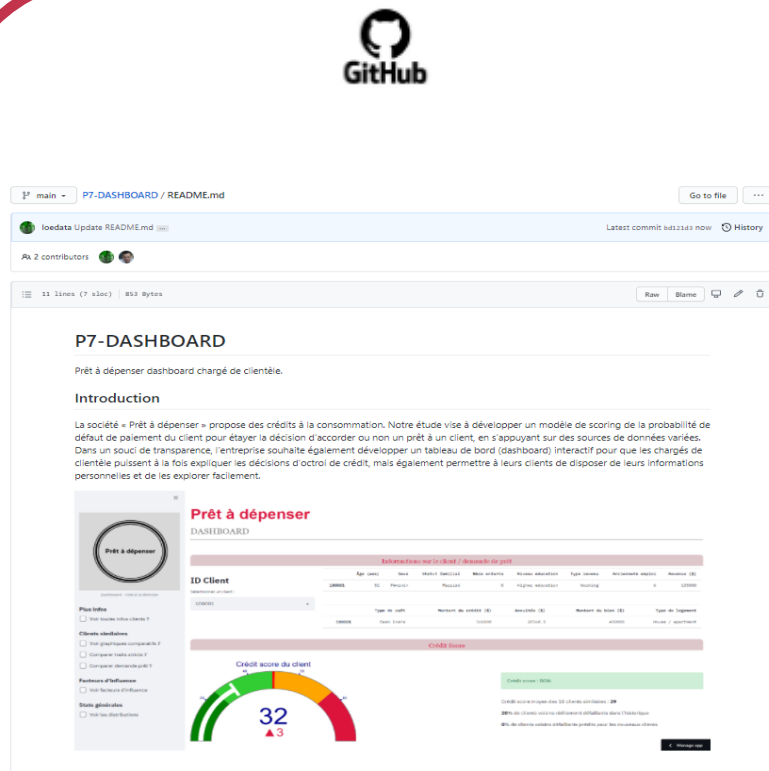


Crédit score : BON

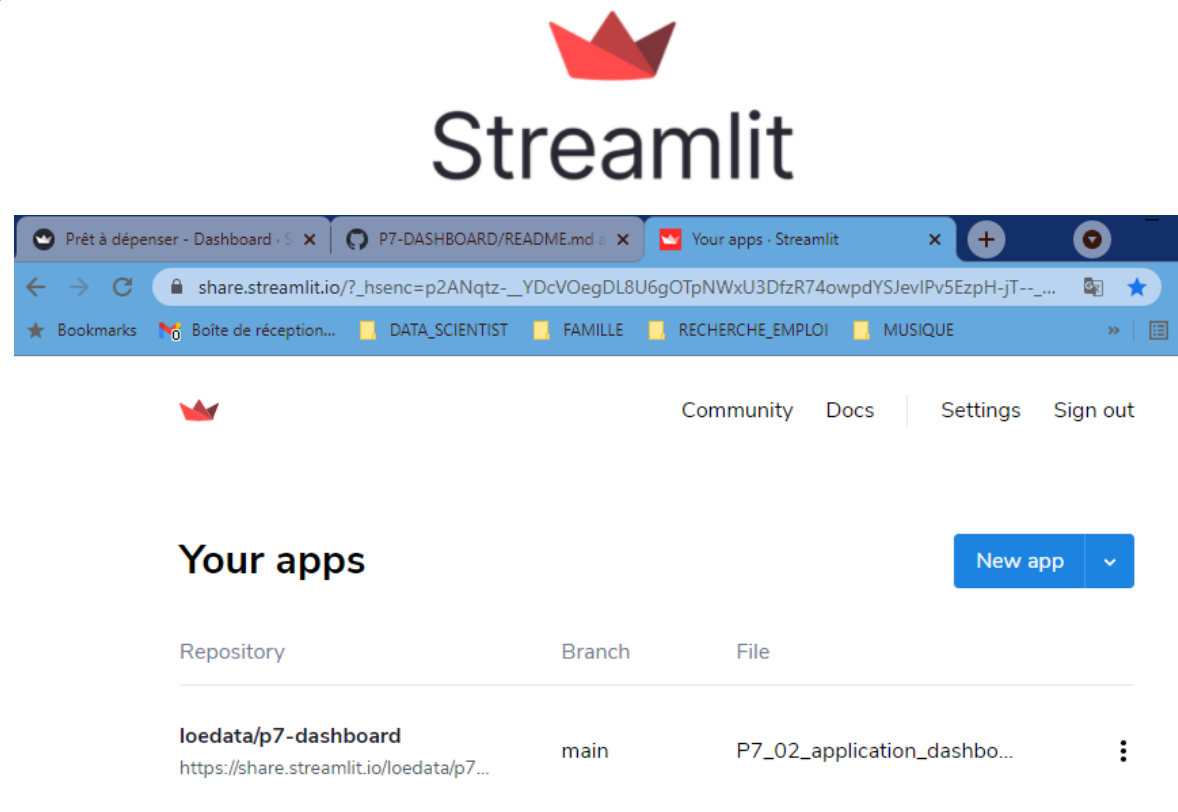
Crédit score moyen des 10 clients similaires : 29

20% de clients voisins réellement défaillants dans l'historique

0% de clients voisins défaillants prédits pour les nouveaux clients



<https://github.com/loedata/P7-DASHBOARD>



Local : P7\_02\_application\_dashboard.py

Distance : [https://share.streamlit.io/loedata/p7-dashboard/main/P7\\_02\\_application\\_dashboard.py](https://share.streamlit.io/loedata/p7-dashboard/main/P7_02_application_dashboard.py)



Dashboard - Aide à la décision

## Plus infos

☒ Voir toutes infos clients ?

## Clients similaires

☐ Voir graphiques clients ?

☐ Comparer traits stricts ?

☐ Comparer demande prêt ?

## Facteurs d'influence

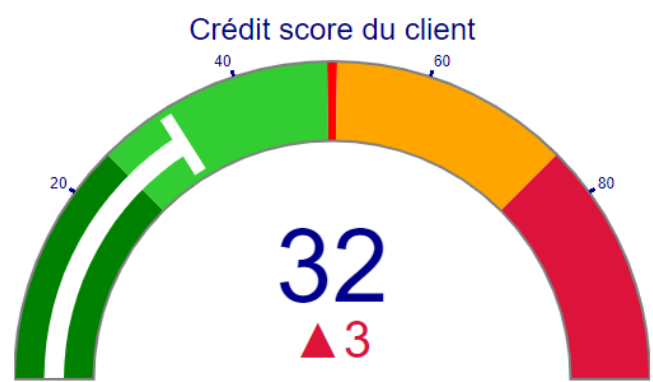
☐ Voir facteurs d'influence

## Stats générales

☐ Voir les distributions

100001	Cash loans	568800	20560.5	450000	House / apartment
--------	------------	--------	---------	--------	-------------------

## Crédit Score



Crédit score : BON

Crédit score moyen des 10 clients similaires : 29

20% de clients voisins réellement défaillants dans l'historique


0% de clients voisins défaillants prédits pour les nouveaux clients

## Plus infos

Toutes les informations du client courant

	SK_ID_CURR	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	NAME_TYPE_SUITE	NAME
0	100001	Cash loans	F	N	Y	0	135000	568800	20560.5	450000	Unaccompanied	
	SK_ID_CURR	CREDIT_ANNUITY_RATIO	PREV_APP_AMT_ANNUITY_M...	PREV_APP_DAYS_FIRST_DU...	PREV_APP_AMT_ANNUITY_M...	PREV_APP_INTEREST_SHAR...	POS_CASH_NAME_CONTRACT...	BUREAU_AMT_CREDIT_SUM...				
0	100001	27.671875	3951	-1709	3951	0.328857421875	0.77783203125	145336				





Prêt à dépenser

Dashboard - Aide à la décision

Plus infos

- ☐ Voir toutes infos clients ?

Clients similaires

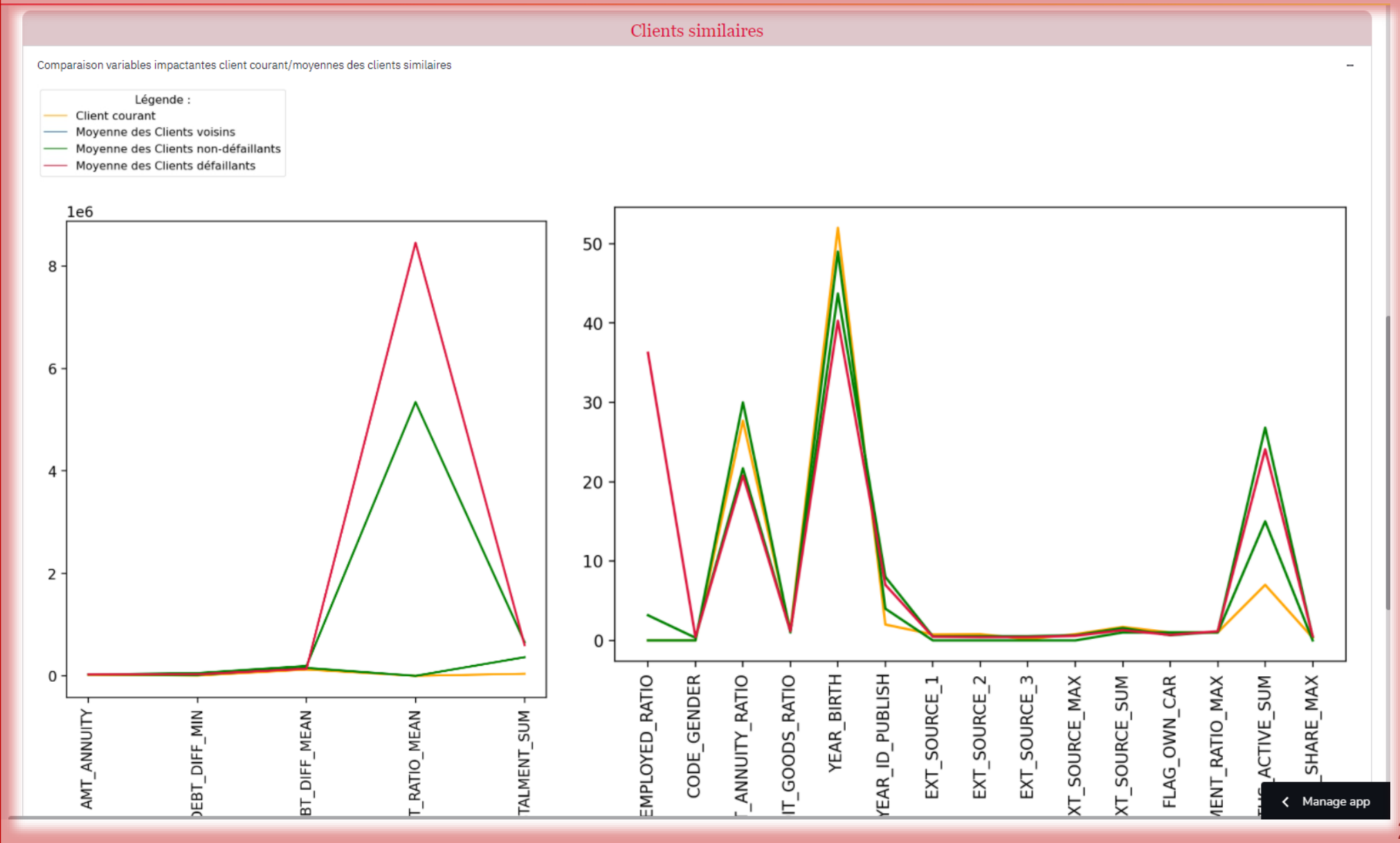
- ☒ Voir graphiques comparatifs ?
- ☐ Comparer traits stricts ?
- ☐ Comparer demandes ?

Facteurs d'influence

- ☐ Voir facteurs d'influence

Stats générales

- ☐ Voir les distributions



Prêt à dépenser

Dashboard - Aide à la décision

Plus infos

☐ Voir toutes infos clients ?

Clients similaires

☒ Voir graphiques comparatifs ?

☐ Comparer crédits stricts ?

☐ Comparer demandes

Facteurs d'influence

☐ Voir facteurs d'influence

Stats générales

☐ Voir les distributions

BUREAU\_CI

BUREAU\_CUR

BUREAU\_CURRENT

I

INST.

POS\_CASH\_NAME\_

P

Feature(s) importance(s) à visualiser :

AMT\_ANNUITY

X

BUREAU\_CURRENT\_CREDIT\_DEBT\_DIFF\_MIN

BUREAU\_CURRENT\_CREDIT\_DEBT\_DIFF\_MEAN

BUREAU\_CURRENT\_DEBT\_TO\_CREDIT\_RATIO\_MEAN

CAR\_EMPLOYED\_RATIO

CODE\_GENDER

CREDIT\_ANNUITY\_RATIO

CREDIT\_GOODS\_RATIO

EXT\_SOURCE\_1

175000

150000

125000

Position du client

< Manage app

Prêt à dépenser



52  
▲ 3



0% de clients voisins défaillants prédits pour les nouveaux clients

Dashboard - Aide à la décision

☐ Voir toutes infos clients ?

☐ Voir graphiques comparatifs ?

✓ Comparer traits stricts ?

☐ Comparer demande prêt ?

☐ Voir facteurs d'influence☐ Voir les distributions

### Comparaison traits stricts

Client courant

	Âge (ans)	Sexe	Statut familial	Nbre enfants	Niveau éducation	Type revenu	Ancienneté emploi	Revenus (\$)
100001	52	Féminin	Married	0	Higher education	Working	6	135000


10 clients similaires

	Âge (ans)	Sexe	Statut familial	Nbre enfants	Niveau éducation	Type revenu	Ancienneté emploi	Revenus (\$)
77677	42	Féminin	Married	0	Higher education	Working	13	135000.000000
257447	47	Féminin	Married	0	Higher education	State servant	13	315000.000000
109458	48	Féminin	Married	0	Higher education	Working	14	171000.000000
212270	48	Féminin	Married	1	Higher education	Commercial associate	2	202500.000000
139230	51	Féminin	Married	0	Higher education	Working	3	225000.000000
97944	54	Féminin	Married	0	Higher education	Pensioner	0	90000.000000
229248	53	Féminin	Married	0	Higher education	State servant	5	157500.000000
213485	51	Féminin	Married	0	Higher education	Working	25	292500.000000
293935	44	Féminin	Married	0	Higher education	Working	5	153000.000000
181115	50	Féminin	Married	0	Higher education	Commercial associate	1	495000.000000

Auteur : loe.rabier@gmail.com - 17/08/2021

Manage app

0% de clients voisins défaillants prédits pour les nouveaux clients



Prêt à dépenser

Dashboard - Aide à la décision

**Plus infos**

☐ Voir toutes infos clients ?

**Clients similaires**

☐ Voir graphiques clients

☐ Comparer traits stricts ?

☒ Comparer demande prêt ?

**Facteurs d'influence**

☐ Voir facteurs d'influence

**Stats générales**

☐ Voir les distributions



Clients similaires

Comparaison demande de prêt

Client courant

	Type de prêt	Montant du crédit (\$)	Annuités (\$)	Montant du bien (\$)	Type de logement
100001	Cash loans	568800	20560.5	450000	House / apartment

10 clients similaires

	Type de prêt	Montant du crédit (\$)	Annuités (\$)	Montant du bien (\$)	Type de logement
77677	Cash loans	665892.000000	21609.000000	477000.000000	House / apartment
257447	Cash loans	905688.000000	29214.000000	756000.000000	House / apartment
109458	Cash loans	526491.000000	19039.500000	454500.000000	House / apartment
212270	Cash loans	1226511.000000	35860.500000	1071000.000000	House / apartment
139230	Cash loans	1298655.000000	35842.500000	1134000.000000	House / apartment
97944	Cash loans	781920.000000	23706.000000	675000.000000	House / apartment
229248	Cash loans	545040.000000	20677.500000	450000.000000	House / apartment
213485	Cash loans	840951.000000	33480.000000	679500.000000	House / apartment
293935	Cash loans	545040.000000	17712.000000	450000.000000	House / apartment
181115	Cash loans	1206954.000000	34717.500000	945000.000000	House / apartment

Prêt à dépenser

Dashboard - Aide à la décision

## Plus infos

☐ Voir toutes infos clients ?

## Clients similaires

☐ Voir graphiques comparatifs ?

☐ Comparer traits stricts ?

☐ Comparer demande

## Facteurs d'influence

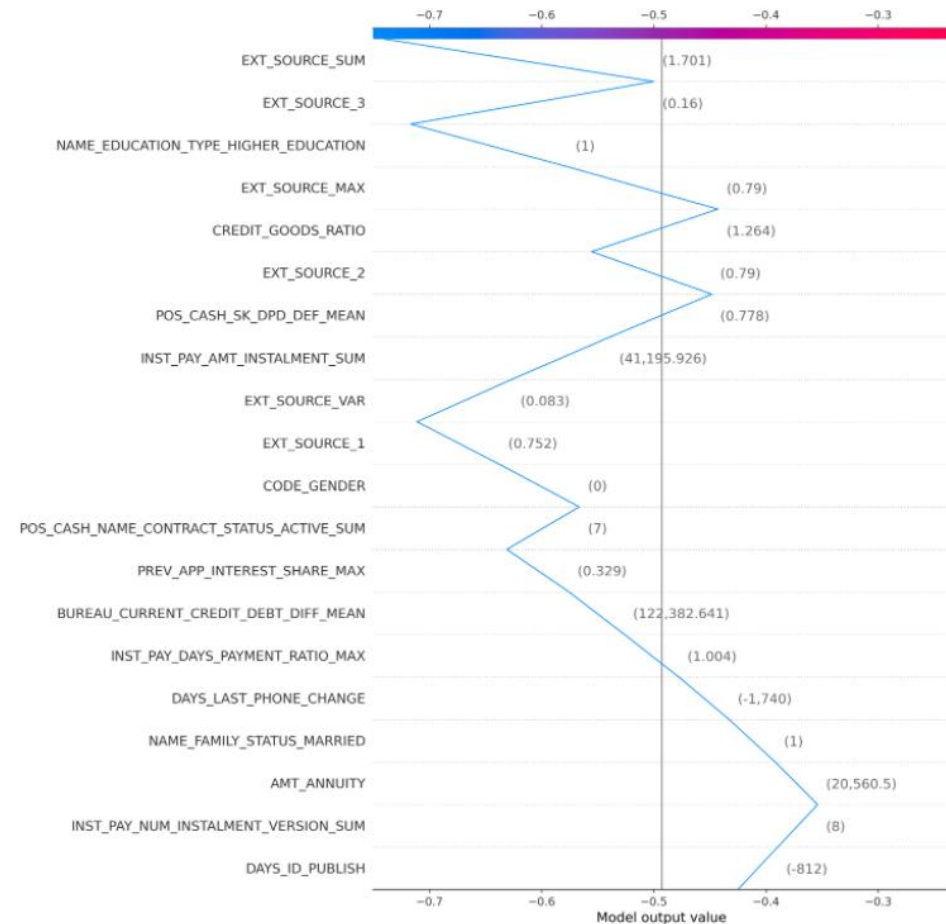
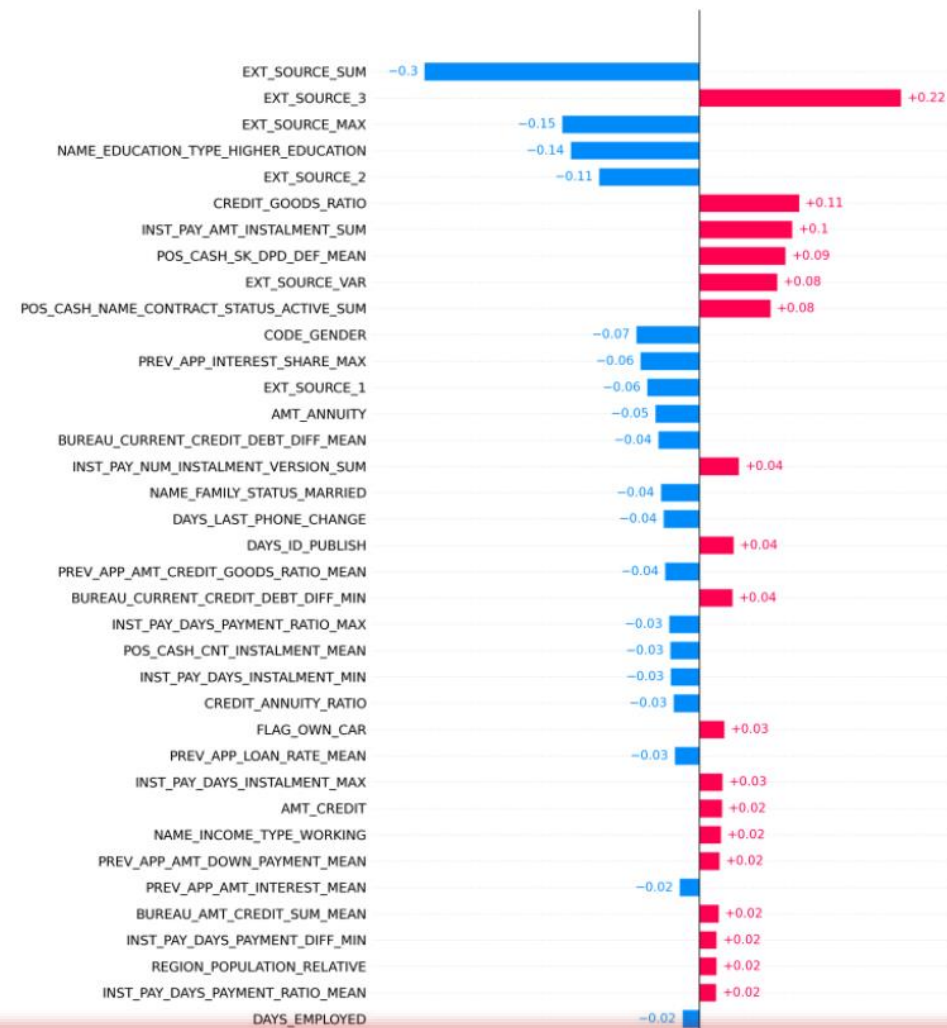
☒ Voir facteurs d'influence

## Stats générales

☐ Voir les distributions

## Variables importantes

Facteurs d'influence du client courant



< Manage app



Credit score moyen des 10 clients similaires : 27

20% de clients voisins réellement défaillants dans l'historique

0% de clients voisins défaillants prédits pour les nouveaux clients

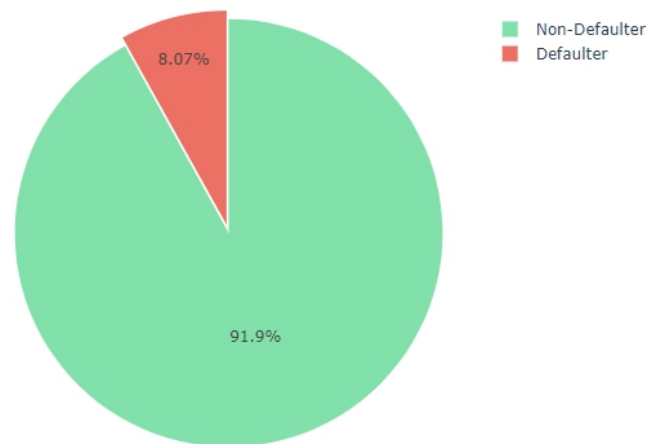
## Distribution des variables générale/pour les défaillants

Distribution des variables

Choisir une variable :

Variable cible

Distribution de la TARGET



### Plus infos

☐ Voir toutes infos clients ?

### Clients similaires

☐ Voir graphiques comparatifs ?

☐ Comparer traits stricts ?

☐ Comparer demande prêt ?

### Facteurs d'influence

☐ Voir facteurs d'influence

### Stats générales

☒ Voir les distributions

- ① Problématique
- ② Données
- ③ Modélisation
- ④ Dashboard
- ⑤ Conclusions**



Problématique de classification binaire avec classes déséquilibrées



Modèle final LightGBM optimisé avec optuna sur la métrique F10 interprétable



Autre modèle XGBoost (recall)



Optimisation des hyperparamètres de LightGBM



Échange avec notre client :

- variables métiers ajoutées adéquates?
- compromis FN/FP (seuil? Taux?)
- revoir métrique métier bancaire inefficace lors des essais
- source externe très influente mais ininterprétable

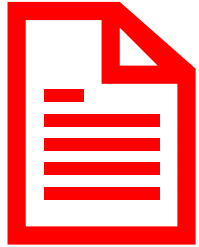


Dashboard : fonctions avancées d'optimisation de Streamlit (cache...) – phase de réentraînement du modèle



Éthique : sexe, âge, transparence, automatisation





## Annexes



Jeu de données	Variables	Valeurs uniques	Ancien type	Nouveau type
application_train	REGION_RATING_CLIENT	1, 2 ou 3	int64	object
application_train	REGION_RATING_CLIENT_W_CITY	1, 2 ou 3	int64	object
application_train	EMERGENCYSTATE_MODE	Yes/No	Object	int8 (1/0)
application_train	FLAG_OWN_CAR	Y/N	Object	int8 (1/0)
application_train	FLAG_OWN_REALTY	Y/N	Object	int8 (1/0)

Jeu de données	Variables	Valeurs aberrantes	Nouvelle valeur
application_train	DAYS_EMPLOYED	365243 (1000 ans)	np.nan
credit_card_balance	AMT_PAYMENT_CURRENT	4000000 (80 ans)	np.nan
previous_application	DAYS_FIRST_DRAWING	365243 (1000 ans)	np.nan
previous_application	DAYS_FIRST_DUE	365243 (1000 ans)	np.nan
previous_application	DAYS_LAST_DUE_1ST_VERSION	365243 (1000 ans)	np.nan
previous_application	DAYS_LAST_DUE	365243 (1000 ans)	np.nan
previous_application	DAYS_TERMINATION	365243 (1000 ans)	np.nan
previous_application	SELLERPLACE_AREA	4000000	np.nan
bureau	DAYS_CREDIT_ENDDATE	> -80*365	np.nan
bureau	DAYS_ENDDATE_FACT	> -80*365	np.nan
bureau	DAYS_CREDIT_UPDATE	> -80*365	np.nan

Jeu de données	Variables	Valeur unique non connue du test set	Correction
application_train	CODE_GENDER	XNA	Suppression des 4 valeurs concernées dans le train set
application_train	NAME_INCOME_TYPE	XNA	Remplacement par np.nan des 5 valeurs concernées dans le train set
application_train	NAME_FAMILY_STATUS	Maternity leave	Remplacement par np.nan des 2 valeurs concernées dans le train set

## application\_train/test

- Parmi les variables importantes repérées lors de l'analyse exploratoire pour départager les non-défaillants des défaillants, la variable FLOORSMIN\_AVG est celle qui a le plus de valeurs manquantes (67.85%).

- On fixera **le seuil de suppression** des variables ayant de nombreuses valeurs manquantes à **68%**.

- Les variables supprimées :

COMMONAREA\_AVG  
LIVINGAPARTMENTS\_AVG  
NONLIVINGAPARTMENTS\_AVG  
COMMONAREA\_MODE  
LIVINGAPARTMENTS\_MODE  
NONLIVINGAPARTMENTS\_MODE  
COMMONAREA\_MEDI  
LIVINGAPARTMENTS\_MEDI  
NONLIVINGAPARTMENTS\_MEDI  
FONDKAPREMONT\_MODE

	Nombres de valeurs manquantes	% de valeurs manquantes
COMMONAREA_AVG	214862	69.8700
COMMONAREA_MODE	214862	69.8700
COMMONAREA_MEDI	214862	69.8700
NONLIVINGAPARTMENTS_MODE	213512	69.4300
NONLIVINGAPARTMENTS_MEDI	213512	69.4300
NONLIVINGAPARTMENTS_AVG	213512	69.4300
FONDKAPREMONT_MODE	210293	68.3900
LIVINGAPARTMENTS_MODE	210197	68.3600
LIVINGAPARTMENTS_MEDI	210197	68.3600
LIVINGAPARTMENTS_AVG	210197	68.3600
FLOORSMIN_MODE	208640	67.9500
FLOORSMIN_AVG	208640	67.8500
FLOORSMIN_MEDI	208640	67.8500
YEARS_BUILD_MODE	204486	66.5000
YEARS_BUILD_AVG	204486	66.5000
YEARS_BUILD_MEDI	204486	66.5000
OWN_CAR_AGE	202927	65.9900
LANDAREA_MEDI	182588	59.3800
LANDAREA_AVG	182588	59.3800
LANDAREA_MODE	182588	59.3800
BASEMENTAREA_MEDI	179942	58.5200
BASEMENTAREA_AVG	179942	58.5200
BASEMENTAREA_MODE	179942	58.5200
EXT_SOURCE_1	173376	56.3800
NONLIVINGAREA_AVG	169680	55.1800
NONLIVINGAREA_MEDI	169680	55.1800
NONLIVINGAREA_MODE	169680	55.1800

# 5 Annexe Données – Nettoyage – Imputation



Préparation de 3 jeux de données avec différentes imputations pour retenir celui qui donne les meilleurs résultats lors de la modélisation avec Pycaret

ESSAI n°1

- Quantitatives Médiane
- Qualitatives Mode

ESSAI n°2

- Quantitatives Constante 0
- Qualitatives Constante XNA

ESSAI n°3

- Quantitatives NaNImputer (verstack)
- Qualitatives Constante XNA



ESSAI n°1		Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier		0.9180	0.7763	0.0716	0.4501	0.1236	0.1037	0.1546	64.293
lightgbm	Light Gradient Boosting Machine		0.9166	0.7674	0.0487	0.3737	0.0862	0.0688	0.1108	9.127
xgboost	Extreme Gradient Boosting		0.9159	0.7647	0.0819	0.3981	0.1359	0.1114	0.1514	49.868

ESSAI n°2		Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier		0.9179	0.7769	0.0735	0.4477	0.1263	0.1060	0.1561	65.299
lightgbm	Light Gradient Boosting Machine		0.9169	0.7672	0.0539	0.3937	0.0947	0.0768	0.1214	8.570
xgboost	Extreme Gradient Boosting		0.9157	0.7645	0.0838	0.3948	0.1382	0.1133	0.1523	52.706

ESSAI n°3		Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier		0.9183	0.7796	0.0723	0.4603	0.1250	0.1054	0.1579	63.151
lightgbm	Light Gradient Boosting Machine		0.9175	0.7731	0.0466	0.4025	0.0834	0.0679	0.1146	9.175
xgboost	Extreme Gradient Boosting		0.9162	0.7690	0.0808	0.4044	0.1346	0.1107	0.1521	49.295

## Automatique

- Création de variables statistiques : quantitatives : min, max, sum, mean, var
- Création de variables statistiques : quantitatives : sum, count, mean

## Manuel

- revenu, de rente et de crédit : ratio/différence
- Jours en années, changement de jours : ratio
- Âge de la voiture, ancienneté d'emploi : ratio/différence
- Flag sur les téléphones : ratio/différence
- Membres de la famille : ratio/différence
- Note de la région où vit le client : ratio/différence
- Données externes : ratio, moyenne, max, min
- Informations sur le bâtiment : somme, multiplication
- Défauts de paiements et les défauts observables : somme/ratio
- Flag sur les documents : somme, moyenne, variance, écart-type
- Modification du demandeur : somme/ratio

Dataframe	Nouvelle variables	Description
Variables de revenu, de rente et de crédit : ratio / différence		
application_train/test	CREDIT_INCOME_RATIO	Ratio : Montant du crédit du prêt / Revenu du demandeur
	CREDIT_ANNUITY_RATIO	Ratio : Montant du crédit du prêt / Annuité de prêt
	ANNUITY_INCOME_RATIO	Ratio : Annuité de prêt / Revenu du demandeur
	INCOME_ANNUITY_DIFF	Différence : Revenu du demandeur - Annuité de prêt
	CREDIT_GOODS_RATIO	Ratio : Montant du crédit du prêt / prix des biens pour lesquels le prêt est accordé / Crédit est supérieur au prix des biens ?
	INCOME_GOODS_DIFF	Différence : Revenu du demandeur - prix des biens pour lesquels le prêt est accordé
	INCOME_AGE_RATIO	Ratio : Annuité de prêt / Âge du demandeur au moment de la demande
	CREDIT_AGE_RATIO	Ratio : Montant du crédit du prêt / Âge du demandeur au moment de la demande
	INCOME_EXT_RATIO	Ratio : Revenu du demandeur / Score normalisé de la source de données externe 3
	CREDIT_EXT_RATIO	Ratio : Montant du crédit du prêt / Score normalisé de la source de données externe
	HOUR_PROCESS_CREDIT_MUL	Multiplication : Revenu du demandeur * heure à laquelle le demandeur a fait sa demande de prêt
Variables sur l'âge		
application_train/test	YEARS_BIRTH	Âge du demandeur au moment de la demande DAYS_BIRTH en années
	AGE_EMPLOYED_DIFF	Différence : Âge du demandeur - Ancienneté dans l'emploi à date demande
	EMPLOYED_AGE_RATIO	Ratio : Ancienneté dans l'emploi à date demande / Âge du demandeur
	LAST_PHONE_BIRTH_RATIO	Ratio : nombre de jours avant la demande où le demandeur a changé de téléphone \ âge du client



# 6 Annexe Données – Feature Engineering – Manuel



Dataframe	Nouvelle variables	Description
Variables sur l'âge		
application_train/test	LAST_PHONE_EMPLOYED_RATIO	Ratio : nombre de jours avant la demande où le demandeur a changé de téléphone \ ancienneté dans l'emploi
Variables sur la voiture		
application_train/test	CAR_EMPLOYED_DIFF	Différence : Âge de la voiture du demandeur - Ancienneté dans l'emploi à date demande
	CAR_EMPLOYED_RATIO	Ratio : Âge de la voiture du demandeur / Ancienneté dans l'emploi à date demande
	CAR_AGE_DIFF	Différence : Âge du demandeur - Âge de la voiture du demandeur
	CAR_AGE_RATIO	Ratio : Âge de la voiture du demandeur / Âge du demandeur
Variables sur les contacts		
application_train/test	FLAG_CONTACTS_SUM	Somme : téléphone portable? + téléphone professionnel? + téléphone professionnel fixe? + téléphone portable joignable? + adresse de messagerie électronique?
Variables sur les membres de la famille		
application_train/test	CNT_NON_CHILDREN	Différence : membres de la famille - enfants (adultes)
	CHILDREN_INCOME_RATIO	Ratio : nombre d'enfants / Revenu du demandeur
	PER_CAPITA_INCOME	Ratio : Revenu du demandeur / membres de la famille : revenu par tête
Variables sur la région		
application_train/test	REGIONS_INCOME_MOY	Moyenne : moyenne de notes de la région/ville où vit le client * revenu du demandeur
	REGION_RATING_MAX	Max : meilleure note de la région/ville où vit le client
	REGION_RATING_MIN	Min : plus faible note de la région/ville où vit le client

Dataframe	Nouvelle variables	Description
Variables sur la région		
application_train/test	REGION_RATING_MEAN	Moyenne : des notes de la région et de la ville où vit le client
	REGION_RATING_MUL	Multiplication : note de la région/ note de la ville où vit le client
	FLAG_REGIONS_SUM	Somme : des indicateurs : - Indicateur si l'adresse permanente du client ne correspond pas à l'adresse de contact (1=différent ou 0=identique - au niveau de la région) - Indicateur si l'adresse permanente du client ne correspond pas à l'adresse professionnelle (1=différent ou 0=identique - au niveau de la région) - Indicateur si l'adresse de contact du client ne correspond pas à l'adresse de travail (1=différent ou 0=identique - au niveau de la région). - Indicateur si l'adresse permanente du client ne correspond pas à l'adresse de contact (1=différent ou 0=identique - au niveau de la ville) - Indicateur si l'adresse permanente du client ne correspond pas à l'adresse professionnelle (1=différent ou 0=même - au niveau de la ville). - Indicateur si l'adresse de contact du client ne correspond pas à l'adresse de travail (1=différent ou 0=identique - au niveau de la ville).
Variables sur les sources externes : sum, min, multiplication, max, var, scoring		
application_train/test	EXT_SOURCE_SUM	Somme : somme des scores des 3 sources externes
	EXT_SOURCE_MEAN	Moyenne : moyenne des scores des 3 sources externes
	EXT_SOURCE_MUL	Multiplication : des scores des 3 sources externes
	EXT_SOURCE_MAX	Max : Max parmi les 3 scores des 3 sources externes
	EXT_SOURCE_MIN	Min : Min parmi les 3 scores des 3 sources externes
	EXT_SOURCE_VAR	

# 6 Annexe Données – Feature Engineering – Manuel



Dataframe	Nouvelle variables	Description
Variables sur les sources externes : sum, min, multiplication, max, var, scoring		
application_train/test	WEIGHTED_EXT_SOURCE	Scoring : scoring des scores des 3 sources externes, score 1 poids 2...
Variables sur le bâtiment		
application_train/test	APARTMENTS_SUM_AVG	Somme : Informations normalisées sur l'immeuble où vit le demandeur des moyennes de la taille de l'appartement, de la surface commune, de la surface habitable, de l'âge de l'immeuble, du nombre d'ascenseurs, du nombre d'entrées, de l'état de l'immeuble et du nombre d'étages.
	APARTMENTS_SUM_MODE	Somme : Informations normalisées sur l'immeuble où vit le demandeur des modes de la taille de l'appartement, de la surface commune, de la surface habitable, de l'âge de l'immeuble, du nombre d'ascenseurs, du nombre d'entrées, de l'état de l'immeuble et du nombre d'étages.
	APARTMENTS_SUM_MEDI	Somme : Informations normalisées sur l'immeuble où vit le demandeur des médianes de la taille de l'appartement, de la surface commune, de la surface habitable, de l'âge de l'immeuble, du nombre d'ascenseurs, du nombre d'entrées, de l'état de l'immeuble et du nombre d'étages.
	INCOME_APARTMENT_AVG_MUL	Multiplication : somme des moyennes des infos sur le bâtiment * revenu du demandeur
	INCOME_APARTMENT_MODE_MUL	Multiplication : somme des modes des infos sur le bâtiment * revenu du demandeur
	INCOME_APARTMENT_MEDI_MUL	Multiplication : somme des médianes des infos sur le bâtiment * revenu du demandeur
Variables sur les défauts de paiements et les défauts observables		
application_train/test	OBS_30_60_SUM	Somme : nombre d'observations de l'environnement social du demandeur avec des défauts observables de 30 DPD (jours de retard) + nombre d'observations de l'environnement social du demandeur avec des défauts observables de 60 DPD (jours de retard)

Dataframe	Nouvelle variables	Description
Variables sur les défauts de paiements et les défauts observables		
application_train/test	DEF_30_60_SUM	Somme : nombre d'observations de l'environnement social du demandeur avec des défauts de paiement de 30 DPD (jours de retard) + nombre d'observations de l'environnement social du demandeur avec des défauts de paiement de 60 DPD (jours de retard)
	OBS_DEF_30_MUL	Multiplication : nombre d'observations de l'environnement social du demandeur avec des défauts observables de 30 DPD (jours de retard) * nombre d'observations de l'environnement social du demandeur avec des défauts observables de 60 DPD (jours de retard)
	OBS_DEF_60_MUL	Multiplication : nombre d'observations de l'environnement social du demandeur avec des défauts de paiement de 30 DPD (jours de retard) * nombre d'observations de l'environnement social du demandeur avec des défauts de paiement de 60 DPD (jours de retard)
	SUM_OBS_DEF_ALL	Somme : nombre d'observations de l'environnement social du demandeur avec des défauts de paiement ou des défauts observables avec 30 DPD (jours de retard) et 60 DPD.
	OBS_30_CREDIT_RATIO	Ratio : Montant du crédit du prêt / nombre d'observations de l'environnement social du demandeur avec des défauts observables de 30 DPD (jours de retard)
	OBS_60_CREDIT_RATIO	Ratio : Montant du crédit du prêt / nombre d'observations de l'environnement social du demandeur avec des défauts observables de 60 DPD (jours de retard)
	DEF_30_CREDIT_RATIO	Ratio : Montant du crédit du prêt / nombre d'observations de l'environnement social du demandeur avec des défauts de paiement de 30 DPD (jours de retard)
	DEF_60_CREDIT_RATIO	Ratio : Montant du crédit du prêt / nombre d'observations de l'environnement social du demandeur avec des défauts de paiement de 60 DPD (jours de retard)

Dataframe	Nouvelle variables	Description
Variables sur les indicateurs des documents fournis ou non		
application_train/test	FLAGS_DOCUMENTS_SUM	Somme : tous les indicateurs des documents fournis ou non
	FLAGS_DOCUMENTS_AVG	Moyenne : tous les indicateurs des documents fournis ou non
	FLAGS_DOCUMENTS_VAR	Variance : tous les indicateurs des documents fournis ou non
	FLAGS_DOCUMENTS_STD	Ecart-type : tous les indicateurs des documents fournis ou non
Variables sur le détail des modifications du demandeur : jour/heure...		
application_train/test	DAYS_DETAILS_CHANGE_SUM	Somme : nombre de jours avant la demande de changement de téléphone + nombre de jours avant la demande de changement enregistré sur la demande + nombre de jours avant la demande le client où il a changé la pièce d'identité avec laquelle il a demandé le prêt
	AMT_ENQ_SUM	Somme : nombre de demandes de renseignements sur le client adressées au Bureau de crédit une heure + 1 jour + 1 mois + 3 mois + 1 an et 1 jour avant la demande
	ENQ_CREDIT_RATIO	Ratio : somme du nombre de demandes de renseignements sur le client adressées au Bureau de crédit une heure + 1 jour + 1 mois + 3 mois + 1 an et 1 jour avant la demande \ Montant du crédit du prêt
Variables sur les sources externes		
application_train/test	TARGET_NEIGHBORS_500_MEAN	Imputation de la moyenne des 500 valeurs cibles des voisins les plus proches pour chaque application du train set ou test set.

Dataframe	Nouvelle variables	Description
credit_card_balance	AMT_DRAWING_SUM	Somme : Montant retiré au guichet automatique pendant le mois du crédit précédent + Montant prélevé au cours du mois du crédit précédent + Montant des autres prélèvements au cours du mois du crédit précédent + Montant des prélèvements ou des achats de marchandises au cours du mois de la crédibilité précédente.
	CNT_DRAWING_SUM	Somme : Nombre de retraits au guichet automatique durant ce mois sur le crédit précédent + Nombre de retraits pendant ce mois sur le crédit précédent + Nombre d'autres retraits au cours de ce mois sur le crédit précédent + Nombre de retraits de marchandises durant ce mois sur le crédit précédent + Nombre d'échéances payées sur le crédit précédent
	BALANCE_LIMIT_RATIO	Ratio : Solde au cours du mois du crédit précédent \ Limite de la carte de crédit au cours du mois du crédit précédent
	MIN_PAYMENT_RATIO	Ratio : Combien le client a-t-il payé pendant le mois sur le crédit précédent ? / Versement minimal pour ce mois du crédit précédent
	PAYMENT_MIN_DIFF	Différence : Combien le client a-t-il payé pendant le mois sur le crédit précédent ? - Versement minimal pour ce mois du crédit précédent
	MIN_PAYMENT_TOTAL_RATIO	Ratio : Combien le client a-t-il payé au total pendant le mois sur le crédit précédent ? / Versement minimal pour ce mois du crédit précédent
	PAYMENT_MIN_DIFF	Différence : Combien le client a-t-il payé au total pendant le mois sur le crédit précédent ? - Versement minimal pour ce mois du crédit précédent
	AMT_INTEREST_RECEIVABLE	Différence : Montant total à recevoir sur le crédit précédent - Montant à recevoir pour le principal du crédit précédent
	SK_DPD_RATIO	Ratio : DPD (jours de retard) au cours du mois sur le crédit précédent \ DPD (Days past due) au cours du mois avec tolérance (les dettes avec de faibles montants de prêt sont ignorées) du crédit précédent

Dataframe	Nouvelle variables	Description
installments_payments	DAYS_PAYMENT_RATIO	atio : La date à laquelle le versement du crédit précédent était censé être payé (par rapport à la date de demande du prêt actuel) \ Quand les échéances du crédit précédent ont-elles été effectivement payées (par rapport à la date de demande du prêt actuel) ?
	DAYS_PAYMENT_DIFF	Différence : La date à laquelle le versement du crédit précédent était censé être payé (par rapport à la date de demande du prêt actuel) - Quand les échéances du crédit précédent ont-elles été effectivement payées (par rapport à la date de demande du prêt actuel) ?
	AMT_PAYMENT_RATIO	Ratio : Ce que le client a effectivement payé sur le crédit précédent pour ce versement \ Quel était le montant de l'acompte prescrit du crédit précédent sur cet acompte ?
	AMT_PAYMENT_DIFF	Différence : Quel était le montant de l'acompte prescrit du crédit précédent sur cet acompte ? - Ce que le client a effectivement payé sur le crédit précédent pour ce versement
POS_CASH_balance	SK_DPD_RATIO	Ratio : DPD (jours de retard) au cours du mois du crédit précédent \ DPD au cours du mois avec tolérance (les dettes de faible montant sont ignorées) du crédit précédent
	TOTAL_TERM	omme : Nombre d'échéances payées sur le crédit précédent + Versements restant à payer sur le crédit précédent
previous_application	AMT_CREDIT_APPLICATION_RATIO	Ratio : montant de crédit demandé lors de la demande précédente \ Montant final du crédit sur la demande précédente
	AMT_DECLINED	Différence : montant de crédit demandé lors de la demande précédente - Montant final du crédit sur la demande précédente
	AMT_CREDIT_GOODS_RATIO	Ratio : Montant final du crédit sur la demande précédente \ Prix du bien demandé (le cas échéant) sur la demande précédente



# 6 Annexe Données – Feature Engineering – Manuel



Dataframe	Nouvelle variables	Description
previous_application	AMT_CREDIT_GOODS_DIFF	Différence : Montant final du crédit sur la demande précédente - Prix du bien demandé (le cas échéant) sur la demande précédente
	CREDIT_DOWNPAYMENT_RATIO	Ratio : Acompte sur la demande précédente \ Montant final du crédit sur la demande précédente
	GOOD_DOWNPAYMET_RATIO	Ratio : Acompte sur la demande précédente \ Prix du bien demandé (le cas échéant) sur la demande précédente
	INTEREST_DOWNPAYMENT	Multiplication : Taux d'acompte normalisé sur le crédit antérieur * Acompte sur la demande précédente
	INTEREST_CREDIT	Multiplication : Montant final du crédit sur la demande précédente * Taux d'intérêt normalisé sur le crédit antérieur
	INTEREST_CREDIT_PRIVILEGED	Multiplication : Montant final du crédit sur la demande précédente * Taux d'intérêt normalisé sur le crédit antérieur
	APPLICATION_AMT_TO_DECISION_RATIO	Ratio : montant de crédit demandé lors de la demande précédente \ nombre de jours pour prendre la décision
	AMT_APPLICATION_TO_SELLERPLACE_AREA	Ratio : montant de crédit demandé lors de la demande précédente \ Zone de vente du vendeur
	ANNUITY	Ratio : Montant final du crédit sur la demande précédente \ Durée du crédit précédent à la demande de la demande précédente
	ANNUITY_GOODS	Ratio : Prix du bien demandé (le cas échéant) sur la demande précédente \ Durée du crédit précédent
	DAYS_FIRST_LAST_DUE_DIFF	Différence : Par rapport à la date de demande de la demande actuelle quelle était la première échéance de la demande précédente? \ Par rapport à la date d'application de l'application actuelle quand la première échéance était-elle censée être celle de l'application précédente ?



Dataframe	Nouvelle variables	Description
previous_application	AMT_CREDIT_NFLAG_LAST_APPL_DAY	Multiplication : Montant final du crédit sur la demande précédente * Indicateur indiquant si la demande était la dernière demande par jour du client
	AMT_CREDIT_YIELD_GROUP	Multiplication : Montant final du crédit sur la demande précédente * Taux d'intérêt groupé en petit moyen et élevé de la demande précédente
	LOAN_RATE	Ratio : Annuité de prêt / Montant final du crédit sur la demande précédente
	AMT_INTEREST	<a href="https://www.kaggle.com/c/home-credit-default-risk/discussion/64598">https://www.kaggle.com/c/home-credit-default-risk/discussion/64598</a>
	INTEREST_SHARE	<a href="https://www.kaggle.com/c/home-credit-default-risk/discussion/64598">https://www.kaggle.com/c/home-credit-default-risk/discussion/64598</a>
	INTEREST_RATE	<a href="https://www.kaggle.com/c/home-credit-default-risk/discussion/64598">https://www.kaggle.com/c/home-credit-default-risk/discussion/64598</a>
bureau_balance	MONTHS_BALANCE	Rendre le nombre de mois positif
	STATUS_MONTHS_RATIO	Ratio du status par mois
bureau	CREDIT_DURATION	Durée du crédit
	FLAG_OVERDUE_RECENT	Retard sur le crédit
	MAX_AMT_OVERDUE_DURATION_RATIO	Ratio : Montant maximal des impayés sur le crédit du Credit Bureau jusqu'à présent (à la date de demande du prêt dans notre échantillon) \ Durée du crédit
	CURRENT_AMT_OVERDUE_DURATION_RATIO	Ratio : Montant actuel en retard sur le crédit du Bureau de crédit \ Durée du crédit
	AMT_OVERDUE_DURATION_LEFT_RATIO	Ratio : Montant actuel en retard sur le crédit du Bureau de crédit \ Durée restante du crédit CB (en jours)
	CNT_PROLONGED_MAX_OVERDUE_MUL	Ratio : Nombre de fois où le crédit du Bureau de crédit a été prolongé \ Durée restante du crédit CB (en jours)
	CNT_PROLONGED_DURATION_RATIO	Ratio : Nombre de fois où le crédit du Bureau de crédit a été prolongé \ Montant maximal des impayés sur le crédit

Dataframe	Nouvelle variables	Description
bureau	CURRENT_DEBT_TO_CREDIT_RATIO	Ratio : Dette actuelle sur le crédit \ Limite de crédit actuelle de la carte de crédit
	AMT_ANNUITY_CREDIT_RATIO	Différence : Limite de crédit actuelle de la carte de crédit - Dette actuelle sur le crédit
	CURRENT_CREDIT_DEBT_DIFF	Différence : nombre de jours avant la demande de prêt où la dernière information sur la solvabilité du Credit Bureau a été fournie - Durée restante du crédit CB (en jours)
	CREDIT_ENDDATE_UPDATE_DIFF	Différence : nombre de jours avant la demande de prêt où la dernière information sur la solvabilité du Credit Bureau a été fournie - Durée restante du crédit CB (en jours)

# 7 Annexe Données – Assemblage – Merge



Dataframe initial	Nbr lignes var. initiales	Nbr lignes var. après FE	Merge avec et suppr var. colinéaires + > 90% nan
application_train/test	(307511, 122) (48744, 121)	(307507, 206) (48744, 205)	
credit_card_balance	(3840312, 23)	agg_ccb_cat (103558, 21) agg_ccb_num (103558, 68)	application_train/test (307507, 246) (48744, 245)
installments_payments	(13605401, 8)	agg_pay_num (339587, 30)	application_train/test (307507, 265) (48744, 264)
POS_CASH_balance	(10001358, 10)	agg_pos_num (337252, 27)	application_train/test (307507, 285) (48744, 284)
previous_application	(1670214, 37)	agg_prev_num (338857, 114)	application_train/test (307507, 552) (48744, 551)
bureau_balance	(27299925, 3)	agg_bureau_balance_par_demandeur (305811, 12)	application_train/test (307507, 555) (48744, 554)
bureau	(1716428, 17)	agg_bureau_num (305811, 60)	application_train/test (307507, 615) (48744, 614)



**train set : 615 variables**  
**test set : 614 variables**



**FEATURE SELECTION  
NÉCESSAIRE**



La **sélection de variables** consiste, étant donné des données dans un espace de grande dimension, à trouver un **sous-sensé de variables pertinentes**

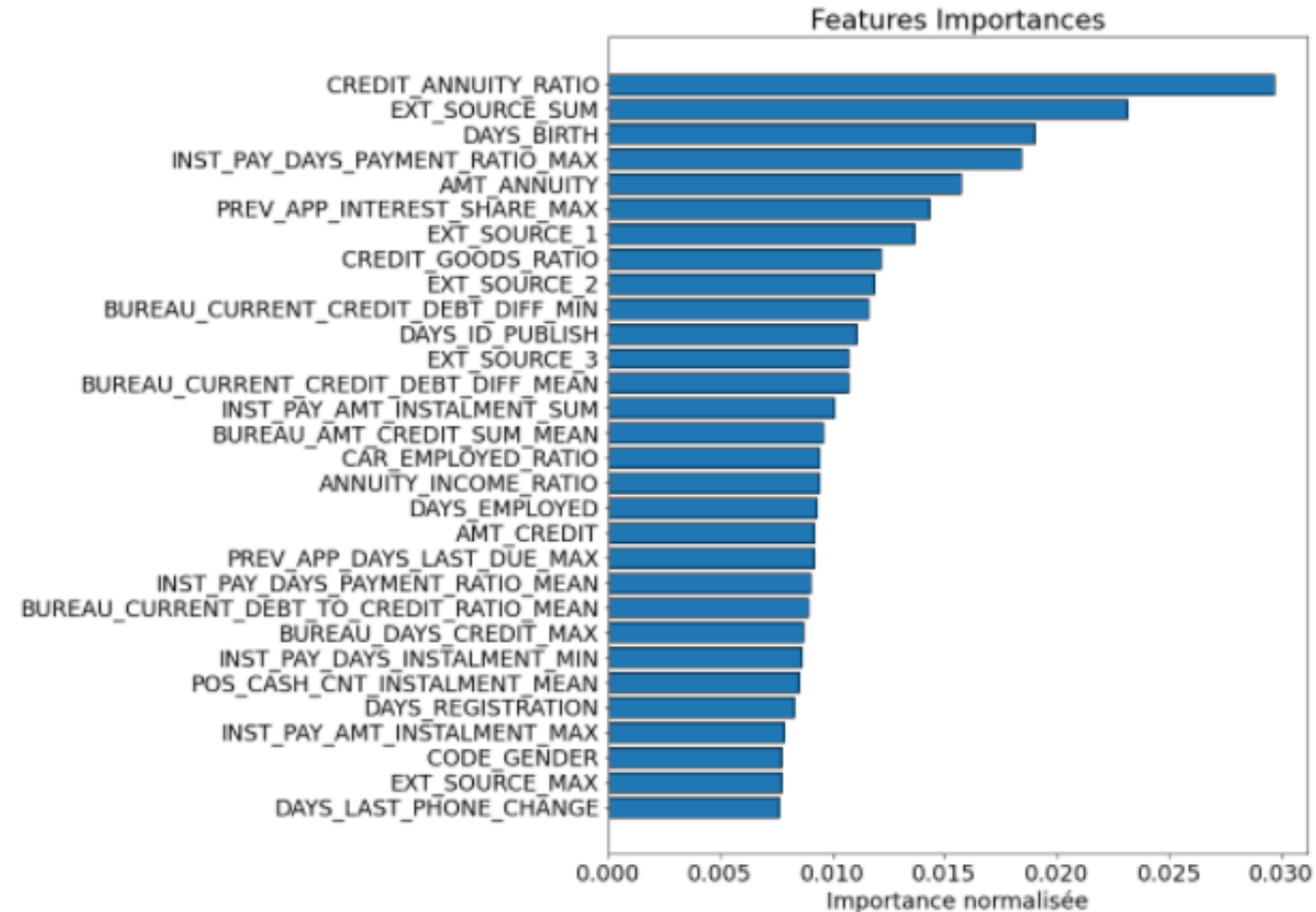
→ **minimiser la perte d'information** venant de la suppression de toutes les autres variables.

## Embedded Methods

LightGbm contient des méthodes intégrées qui apprennent quelles variables contribuent le mieux à la précision du modèle pendant sa création.

Une valeur est calculée et liée à chaque variable du jeu de données servant à entraîner le modèle.

Une fois normalisée, le diagramme ci-contre permet de représenter l'ordre des variables les plus pertinentes pour notre modèle.



# 9 Annexe Données – Feature Selection – Boruta



Boruta est le nom d'un package R qui implémente un nouvel algorithme de sélection de caractéristiques (wrapper method).

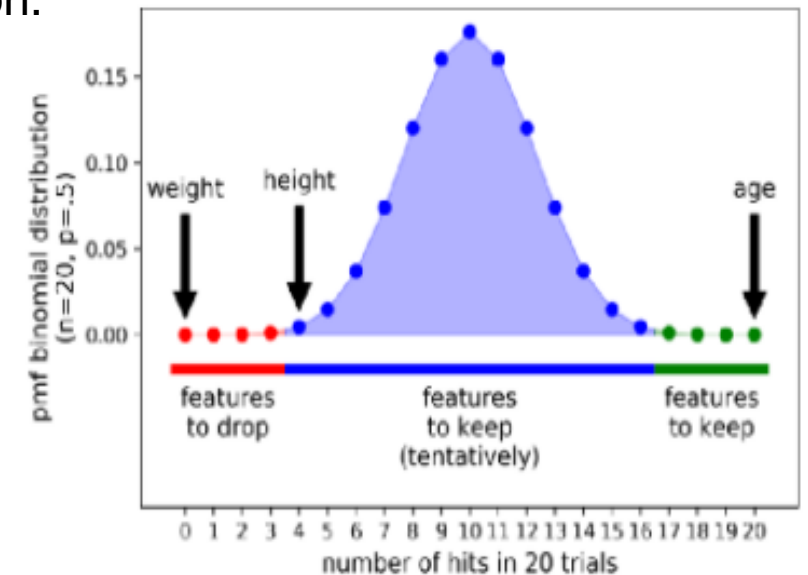
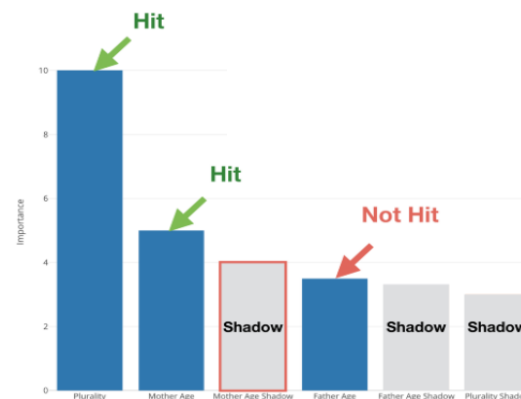
Il **permuté aléatoirement les variables** comme le permutation importance, mais **s'exécute sur toutes les variables en même temps et concatène les caractéristiques mélangées avec les caractéristiques originales**. Le résultat concaténé est utilisé pour ajuster le modèle.

Les caractéristiques mélangées (aussi appelées caractéristiques **fantômes**) sont essentiellement des bruits dont la distribution marginale est identique à celle de la caractéristique originale.

Nous comptons le nombre de fois où une variable est plus performante que le "meilleur" bruit et calculons le degré de confiance dans le fait qu'elle soit meilleure que le bruit (la valeur p) ou non.

Les caractéristiques qui sont assurément meilleures sont marquées "**confirmées**", et celles qui sont assurément égales aux bruits sont marquées "**rejetées**".

Ensuite, nous éliminons les caractéristiques marquées et répétons le processus jusqu'à ce que toutes les caractéristiques soient marquées ou qu'un certain nombre d'itérations soit atteint.



Binomial distribution and positioning of the features

La librairie **BorutaShap** (wrapper method), comme son nom l'indique, combine l'algorithme de sélection de caractéristiques **Boruta** avec la technique **SHAP** (SHapley Additive exPlanations) permet de trouver un ensemble de caractéristiques optimal minimal plutôt que de trouver toutes les caractéristiques pertinentes pour la variable cible. Cela conduit à une sélection non biaisée et stable des attributs importants et non importants. Contrairement au package R original, qui limite l'utilisateur à un modèle Random Forest, BorutaShap permet à l'utilisateur de choisir n'importe quel apprenant basé sur un arbre comme modèle de base.

## Algorithme :

1. Commencez par créer de nouvelles copies de toutes les caractéristiques de l'ensemble de données et nommez-les shadow + feature\_name, mélangez ces caractéristiques nouvellement ajoutées pour supprimer leurs corrélations avec la variable de réponse.
2. Exécutez un classificateur sur les données étendues avec les caractéristiques aléatoires incluses. Ensuite, classez les caractéristiques en utilisant une métrique d'importance des caractéristiques;
3. Créez un seuil en utilisant le score d'importance maximum des caractéristiques fantômes. Attribuez ensuite un hit à toute caractéristique qui a dépassé ce seuil.
4. Pour chaque caractéristique non attribuée, effectuez un test T bilatéral d'égalité. Les attributs dont l'importance est nettement inférieure au seuil sont considérés comme "sans importance" et sont retirés du processus. Les attributs dont l'importance est significativement plus élevée que le seuil sont considérés comme "importants".
5. Supprimez tous les attributs fantômes et répétez la procédure jusqu'à ce qu'une importance ait été attribuée à chaque caractéristique, ou que l'algorithme ait atteint la limite d'exécution fixée précédemment.

Permutation importance (scikit-learn) mesure l'augmentation de l'erreur de prédiction du modèle après avoir permuté les valeurs de la caractéristique, ce qui rompt la relation entre la caractéristique et le véritable résultat.

Le concept est très simple :

Nous mesurons l'importance d'une caractéristique en calculant l'augmentation de l'erreur de prédiction du modèle après permutation de la caractéristique.

Une caractéristique est "importante" si le brassage de ses valeurs augmente l'erreur du modèle, car dans ce cas, le modèle s'est appuyé sur la caractéristique pour la prédiction.

Une caractéristique est "sans importance" si le remaniement de ses valeurs laisse l'erreur du modèle inchangée, car dans ce cas, le modèle n'a pas tenu compte de la caractéristique pour la prédiction.

La mesure de l'importance de la caractéristique de permutation a été introduite par Breiman (2001)<sup>37</sup> pour les forêts aléatoires.

Sur la base de cette idée, Fisher, Rudin et Dominici (2018)<sup>38</sup> ont proposé une version agnostique de l'importance de la caractéristique et l'ont appelée confiance du modèle.



**RFE** (Recursive feature elimination, wrapper method : cela signifie qu'un algorithme d'apprentissage automatique différent est donné et utilisé dans le noyau de la méthode, puis enveloppé par RFE et utilisé pour aider à sélectionner les caractéristiques.) est une méthode de sélection de caractéristiques qui ajuste un modèle et élimine la ou les caractéristiques les plus faibles jusqu'à ce que le nombre spécifié de caractéristiques soit atteint.

Les caractéristiques sont classées par les attributs `coef_` ou `feature_importances_` du modèle, et en éliminant récursivement un petit nombre de caractéristiques par boucle, RFE tente d'éliminer les dépendances et la colinéarité qui peuvent exister dans le modèle.

RFE requiert un nombre spécifique de caractéristiques à conserver, mais il est souvent impossible de savoir à l'avance combien de caractéristiques sont valides.

Pour trouver le nombre optimal de caractéristiques, la validation croisée est utilisée avec RFE pour évaluer différents sous-ensembles de caractéristiques et sélectionner la collection de caractéristiques ayant le meilleur score.

Le visualiseur **RFECV** (Cross Validation) trace le nombre de caractéristiques dans le modèle avec leur score de test validé par croisement et leur variabilité, et visualise le nombre de caractéristiques sélectionné.



## Recall :

$$Rappel = \frac{TP}{TP + FN}$$

la métrique pour déterminer le **taux de vrais positif** est le Rappel (Sensibilité) / **Recall**, elle mesure parmi toutes les observations positives combien ont été classées comme positives. Pour ne pas avoir de pertes, il faut détecter tous les défaillants (classe positive) donc **maximiser la métrique recall**.

## Précision :

$$Précision = \frac{TP}{TP + FP}$$

elle mesure le nombre d'observations prédites comme positives (client défaillant) qui le sont en réalité. Si le client est prédit défaillant alors qu'il ne le sera pas le prêt ne sera pas accordé et les intérêts ne seront pas empochés. Il faut donc **maximiser la 'Precision'**.

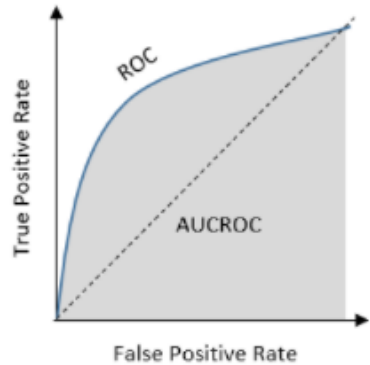
**F-mesure ou F1 :** compromis entre le rappel et précision.

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

$$F_1 = F_{\beta=1} = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = \frac{2 \cdot recall \cdot precision}{recall + precision}$$

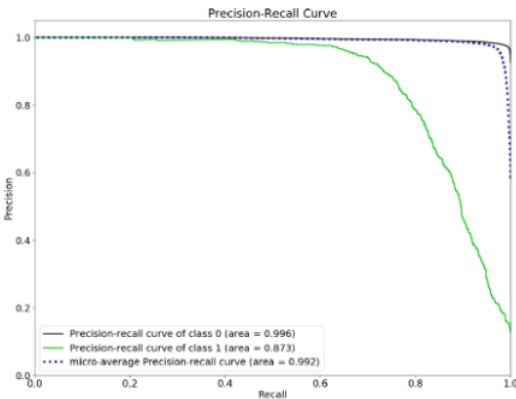
- Si on augmente le recall, la précision diminue, il s'agit de faire un compromis entre ces 2 métriques qui dépendent l'une de l'autre, en prenant en compte le métier/problème de l'entreprise et les coûts associés pour répondre à ces questions.
- Dans notre cas, il faut trouver le plus grand nombre d'observations réellement positives (client prédit défaillant et bien défaillant) et la perte est moins grande si on prédit un client défaillant mais qu'il n'est pas défaillant (faux positifs), donc **on donnera priorité à maximiser le recall au dépend de la précision** (on parle bien de Precision pas d'accuracy).
- Le réglage du paramètre beta pour le Fbeta score permet de donner plus de poids au recall (beta>1) qu'à la précision (0<beta<1).
- Le score F1 est la moyenne harmonique entre le recall et la précision.

## Score ROC\_AUC :



le score ROC AUC est équivalent au calcul de la corrélation de rang entre les prédictions et la cible. Du point de vue de l'interprétation, il est plus utile car il nous indique que cette métrique montre à quel point votre modèle est bon pour classer les prédictions. Elle vous indique la probabilité qu'une instance positive choisie au hasard soit classée plus haut qu'une instance négative choisie au hasard.

## Score PR\_AUC :



calcul de l'aire sous la courbe précision-rappel pour obtenir un nombre qui vous donne des informations sur la performance du modèle. A utiliser :

- lorsque vous voulez communiquer la décision de précision/rappel à d'autres parties prenantes et que vous voulez choisir le seuil qui correspond au problème de l'entreprise.
- lorsque vos données sont fortement déséquilibrées. Puisque l'AUC de PR se concentre principalement sur la classe positive (PPV et TPR), elle se soucie moins de la classe négative fréquente.
- quand vous vous souciez plus de la classe positive que de la classe négative. Si vous vous souciez davantage de la classe positive et donc du PPV et du TPR, vous devriez opter pour la courbe Precision-Recall et la PR AUC (précision moyenne).

## Métier bancaire :

```
def custom_score_2(y_reel, y_pred, taux_tn=1, taux_fp=-1, taux_fn=-10, taux_tp=0):  
    """  
    Métrique métier tentant de minimiser le risque d'accord prêt pour la  
    banque en pénalisant les faux négatifs.  
    Parameters  
    -----  
    y_reel : classe réelle, obligatoire (0 ou 1).  
    y_pred : classe prédite, obligatoire (0 ou 1).  
    taux_tn : Taux de vrais négatifs, optionnel (1 par défaut),  
              le prêt est remboursé : la banque gagne de l'argent ==>  
              à encourager.  
    taux_fp : Taux de faux positifs, optionnel (0 par défaut),  
              le prêt est refusé par erreur : la banque perd les intérêts,  
              manque à gagner mais ne perd pas réellement d'argent (erreur de  
              type I) ==> à pénaliser.  
    taux_fn : Taux de faux négatifs, optionnel (-10 par défaut),  
              le prêt est accordé mais le client fait défaut : la banque perd  
              de l'argent (erreur de type II). ==> à pénaliser  
    taux_tp : Taux de vrais positifs, optionnel (1 par défaut),  
              Le prêt est refusé à juste titre : la banque ne gagne ni ne perd  
              d'argent.  
    Returns  
    -----  
    score : gain normalisé (entre 0 et 1) un score élevé montre une meilleure  
            performance  
    """  
    # Matrice de Confusion  
    (tn, fp, fn, tp) = confusion_matrix(y_reel, y_pred).ravel()  
    # Gain total  
    gain_tot = tn * taux_tn + fp * taux_fp + fn * taux_fn + tp * taux_tp  
    # Gain maximum : toutes les prédictions sont correctes  
    gain_max = (fp + tn) * taux_tn + (fn + tp) * taux_tp  
    # Gain minimum : on accorde aucun prêt, la banque ne gagne rien  
    gain_min = (fp + tn) * taux_fp + (fn + tp) * taux_fn  
  
    custom_score = (gain_tot - gain_min) / (gain_max - gain_min)  
  
    # Gain normalisé (entre 0 et 1) un score élevé montre une meilleure  
    # performance  
    return custom_score
```

# 17 Annexe - Bayesian Optimisation



Librairie bayes\_opt du MIT : l'optimisation bayésienne fonctionne en construisant une distribution postérieure de fonctions (processus gaussien) qui décrit au mieux la fonction que l'on veut optimiser.

Au fur et à mesure que le nombre d'observations augmente, la distribution postérieure s'améliore, et l'algorithme devient plus certain des régions de l'espace des paramètres qui méritent d'être explorées et de celles qui ne le méritent pas, comme le montre l'image ci-dessous.

