



Prêt à dépenser

# IMPLÉMENTER UN MODÈLE DE SCORING

AVRIL 2021

CHRISTELLE TROUSSARD



HOME  
CREDIT

# SOMMAIRE

Problématique

Présentation

1

Analyse et prétraitement des données

Modélisation

2

Méthodologie d'entraînement du modèle

Dashboard

3

Présentation et déploiement

Conclusion

Conclusions – Améliorations possibles

Annexes

# Problématique

Implémenter un modèle de scoring

# Problématique

## Mission :

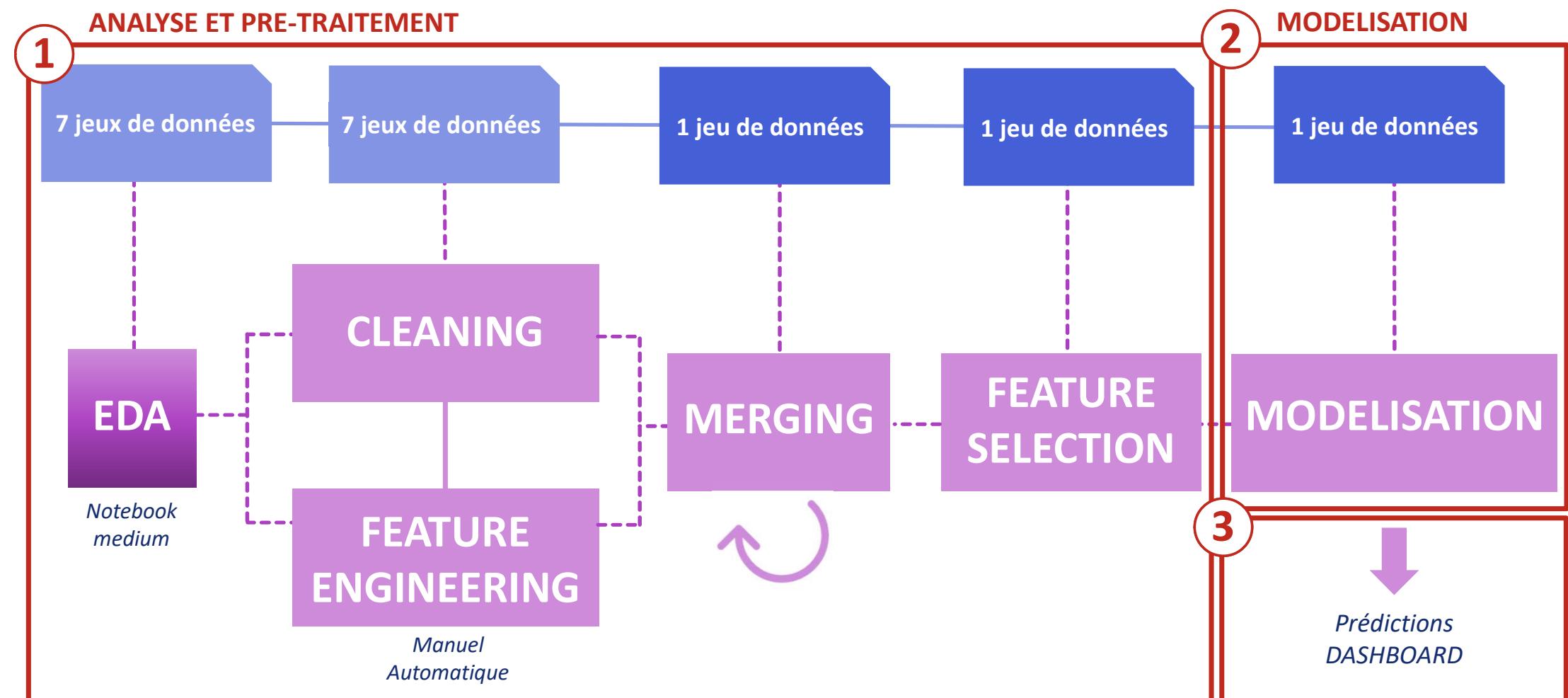
- Développer **un modèle de scoring de la probabilité de défaut de paiement du client.**
- Développer **un dashboard interactif**

## Objectifs :

- **Étayer** la décision d'accorder ou non un prêt à un client potentiel.
- **Expliquer** de façon la plus transparente possible les décisions d'octroi de crédit.
- Permettre aux clients de **disposer de leurs données personnelles** et de les explorer facilement.

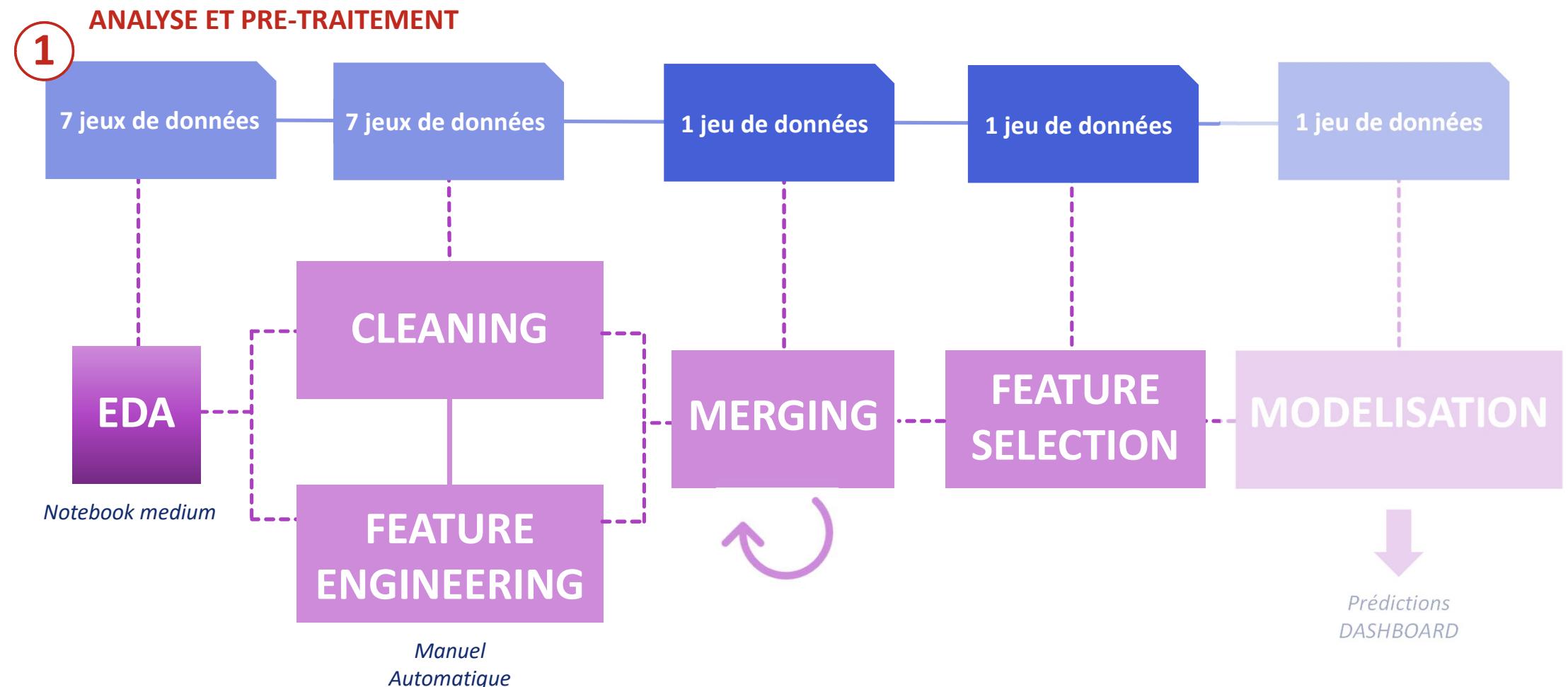


## Pipeline général

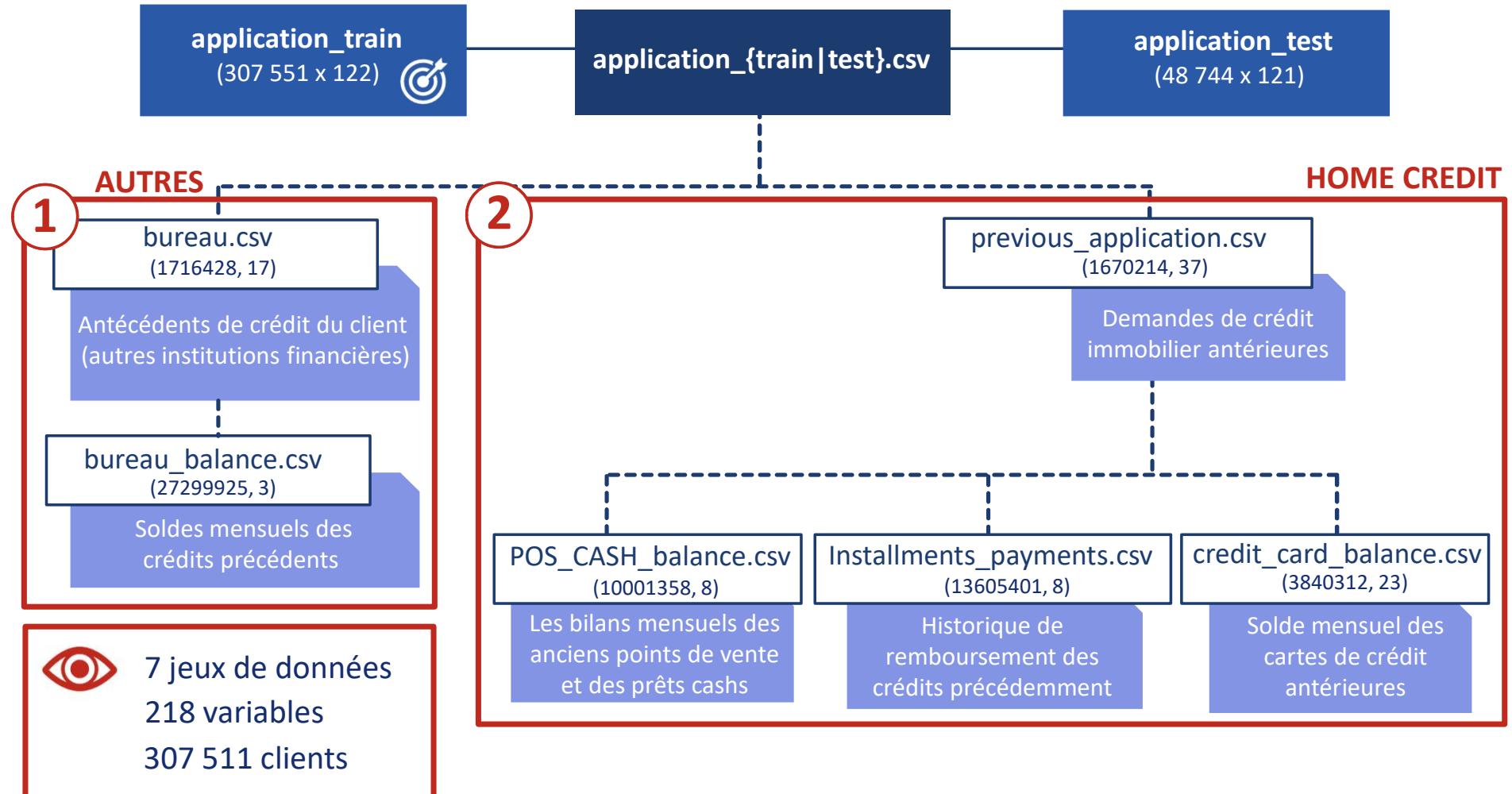


# Présentation

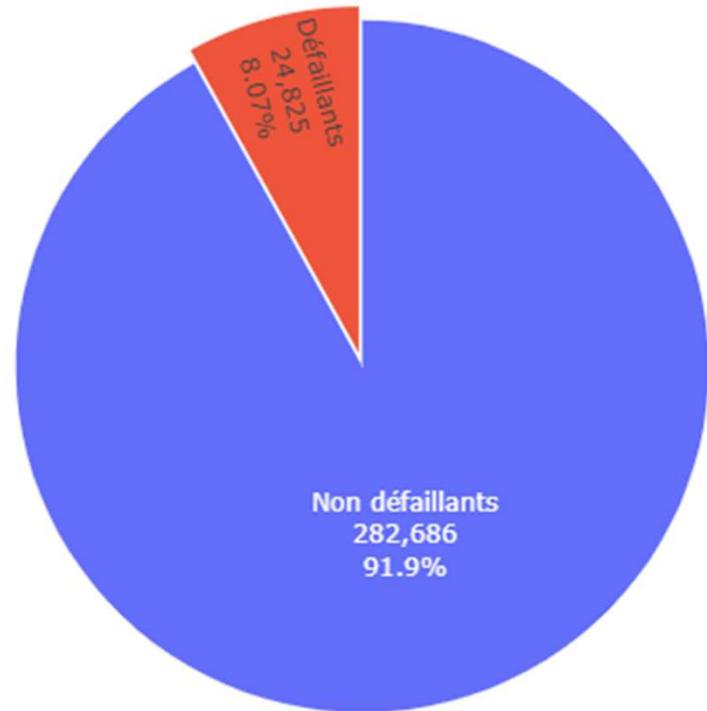
Analyse  
Pré-traitement des données



## Arborescence des 7 jeux de données



## Variable cible



Classification binaire

Variable binaire

Défauillants : 1

Non défauillants : 0

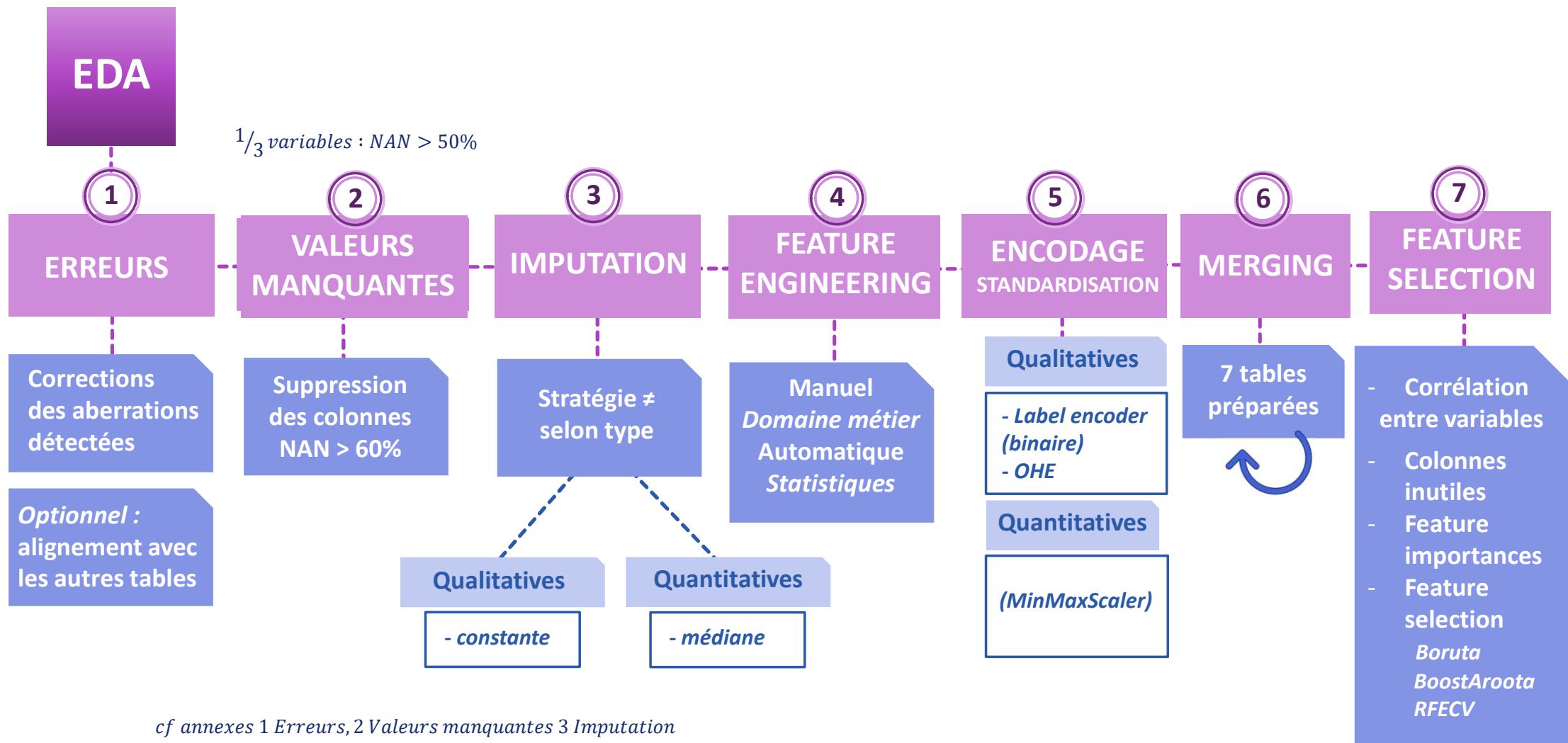
Données déséquilibrées

Défauillants : 8%, classe minoritaire

Non défauillants : 92%, classe majoritaire

Ce déséquilibre devra être pris en compte lors de la construction du modèle, car certains algorithmes sont sensibles au déséquilibre.

## Pipeline preprocessing suivi pour chaque table



# Feature engineering

4

1

## AUTOMATIQUE

Création de variables statistiques

`['count', 'min', 'max', 'mean', 'var']`

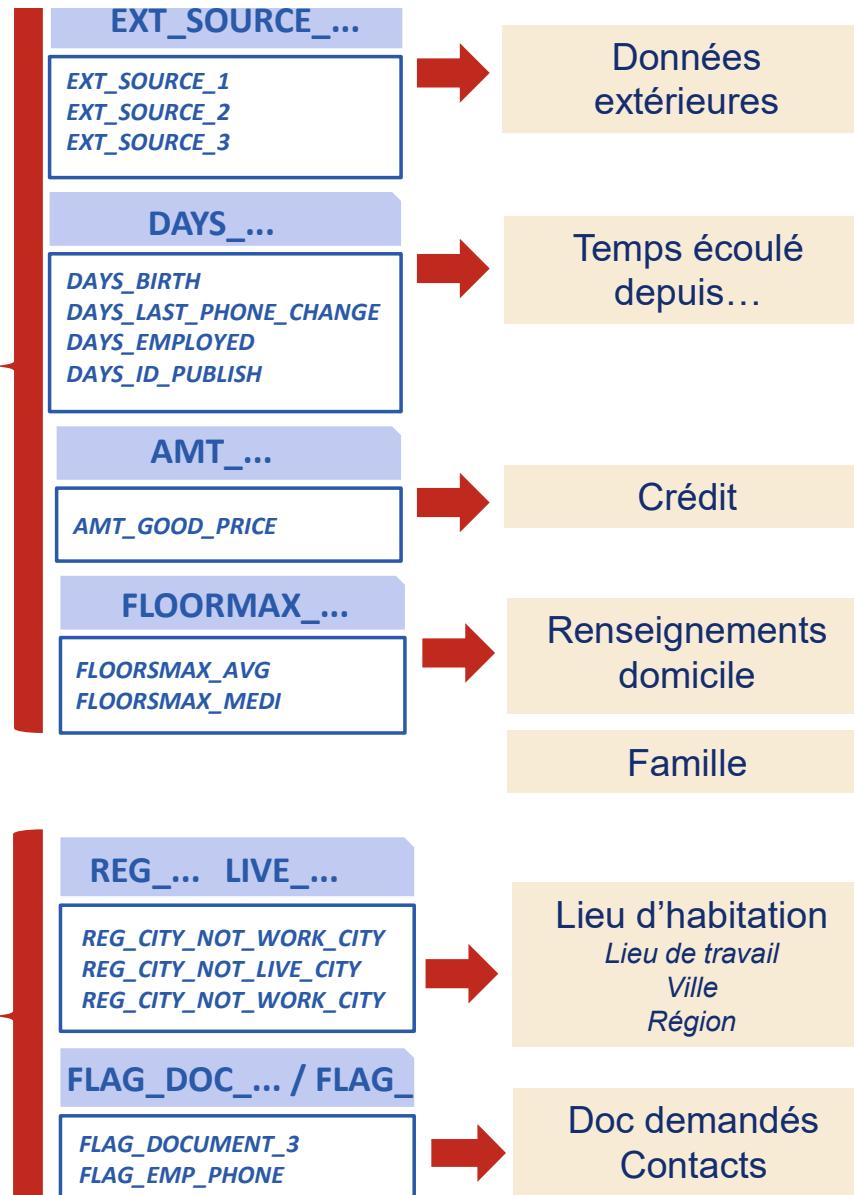
2

## MANUEL

Création de variables « métier »

VARIABLES QUANTITATIVES

VARIABLES QUALITATIVES



*sources moy, mul, min, max, var  
sources somme , somme pondérée*

*diff âge - temps travaillé  
ratio temps travaillé/âge*

*ratio crédit/revenu  
ratio revenu/annuité/âge  
ratio crédit/annuité  
ratio crédit/annuité/âge  
crédit > demande?  
Credit>GoodPrice*

*domicile somme (moy, med, mode)  
domicile mul (moy, med, mode \*revenu)*

*nombre d'adultes dans la famille  
ratio revenu /nbre enfants  
revenu par tête*

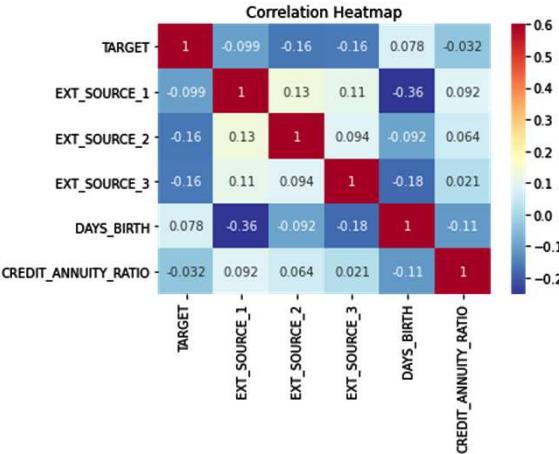
*flag région (sum)*

*flag documents(sum)  
flag contacts sum (phone, email, mobil)*

## Feature engineering

4

Variable qui contient la moyenne de la variable ‘TARGET’ des 500 voisins d'une ligne particulière.  
Les voisins sont calculés en utilisant les **sources externes** et **CREDIT\_ANNUITY\_RATIO**.



```
array([[ 0, 206147, 239933, ..., 164158, 256554, 186757],  
[ 1, 300988, 49779, ..., 267530, 98583, 238550],  
[ 2, 79438, 132624, ..., 93955, 104499, 908],  
...,  
[307508, 253314, 102198, ..., 121411, 274802, 175518],  
[307509, 141421, 167110, ..., 234365, 30088, 288184],  
[307510, 30040, 55730, ..., 154178, 281417, 126690]])
```

Algorithme utilisé : **KNeighborsClassifier** de sklearn  
**k** (nombre de voisins) = 500

1

	EXT_SOURCE_1	EXT_SOURCE_2	EXT_SOURCE_3	CREDIT_ANNUITY_RATIO
0	0.7526	0.7897	0.1595	27.6647
1	0.5650	0.2917	0.4330	12.8249
2	0.5068	0.6998	0.6110	9.5055
3	0.5257	0.5097	0.6127	32.1307
4	0.2021	0.4257	0.5191	19.5060

Création de 2 dataframes :

(à partir de train et test)

neighbors\_train (307 511, 4)

neighbors\_test (48 744, 4)

On récupère la cible ‘TARGET’ :

train\_target = train.TARGET

2

On entraîne le classificateur à l'aide de la méthode **fit()**  
knn.fit(neighbors\_train, train\_target)

3

On récupère les **500 voisins** pour chaque ligne  
train\_500\_neighbors = knn.kneighbors(neighbors\_train)[1] (307511, 500)  
test\_500\_neighbors = knn.kneighbors(neighbors\_test)[1] (48744, 500)

4

On ajoute les moyennes de la cible des 500 voisins dans **une nouvelle colonne** :  
train['TARGET\_NEIGHBORS\_500\_MEAN'] = [train['TARGET'].iloc[ele].mean() for ele in train\_500\_neighbors]  
test['TARGET\_NEIGHBORS\_500\_MEAN'] = [train['TARGET'].iloc[ele].mean() for ele in test\_500\_neighbors]

(Phil)

The most important features that I engineered, in descending order of importance (measured by gain in the LGBM model), were the following:

1) **neighbors\_target\_mean\_500**: The mean TARGET value of the 500 closest neighbors of each row, where each neighborhood was defined by the three external sources and the credit/annuity ratio.

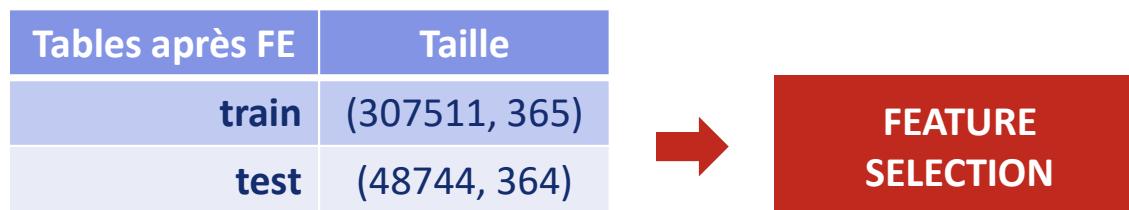
M  
E  
R  
G  
I  
N  
G

Ordre d'assemblage des tables	Nombre de variables Initial (par table)	Nombre de variables après FE (somme cumulée)
Application_train	122	<b>209</b>
bureau	17	
bureau_balance	3	<b>260</b>
previous_application	37	<b>338</b>
POS_CASH_balance	8	<b>342</b>
installments_payments	8	<b>362</b>
credit_card_balance	23	<b>365</b>
<b>TOTAL</b>	<b>218</b>	<b>365</b>

1. Une méthode de filtrage (corrélation de Pearson) est appliquée après FE à chaque table pour éliminer les variables colinéaires.

2. application\_test (48744 clients) :

- traité en parallèle pendant l'étude.
- contient les mêmes variables que le jeu utilisé pour l'entraînement du modèle. (Sauf variable cible)
- utilisé dans la partie dashboard pour simuler des nouveaux clients.



## Feature selection

7

### Jeu après feature engineering

365

(307511, 365)



	Boruta	BoostAroota	Lightgbm
Nombre de variables sélectionnées :	151	162	257
Taille jeu de données	(307511, 151)	(307511, 162)	(307511, 257)

Variables communes



### Jeu après feature selection

109

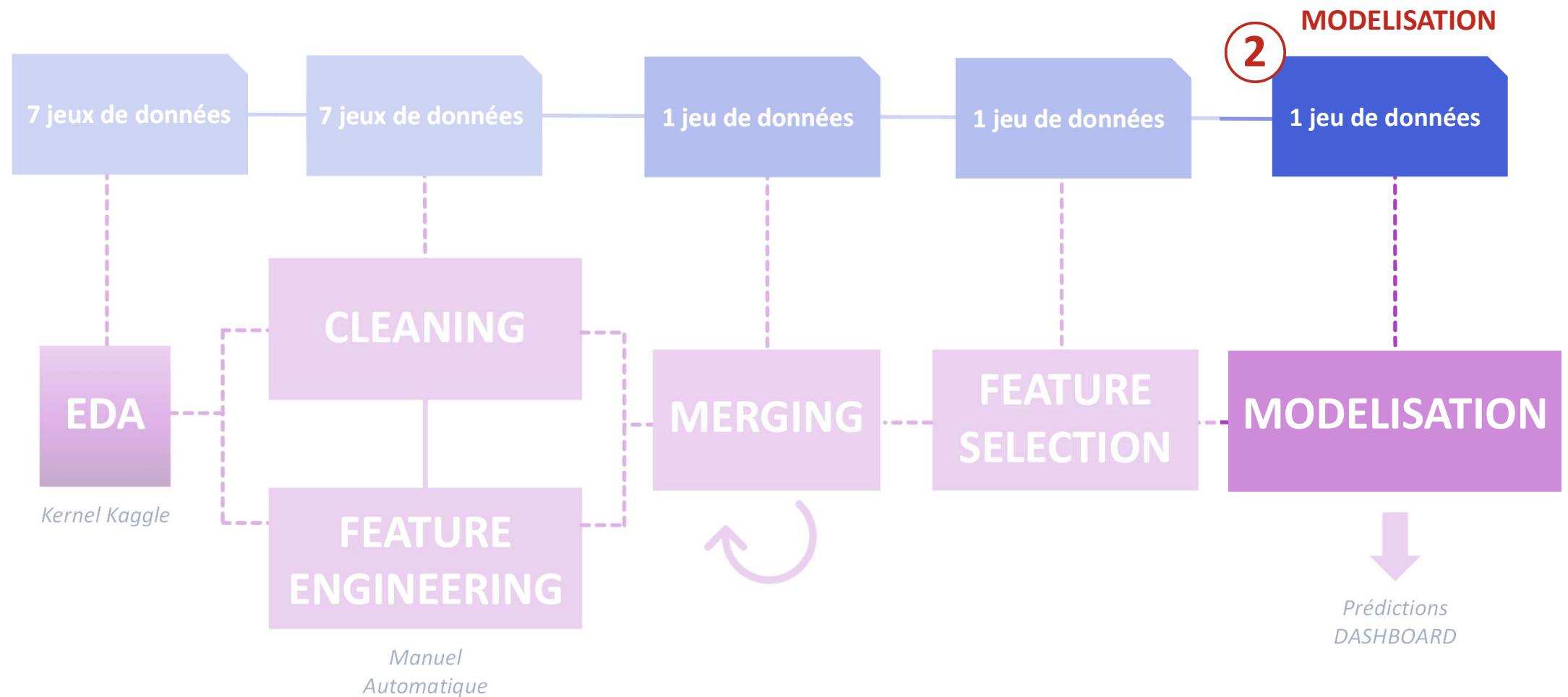
(307511, 109)

	Tables après FS	Taille
train	(307511, 109)	
test	(48744, 108)	
1 ligne par client		

cf annexe feature selection: différentes méthodes

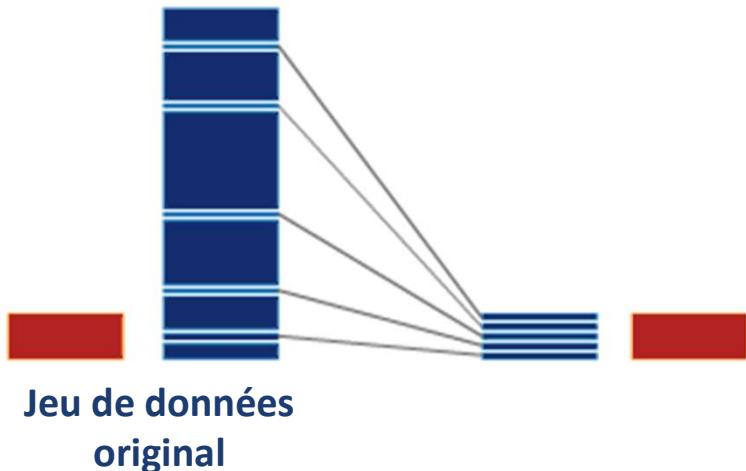
# Modélisation

Méthodologie d'entraînement  
du modèle

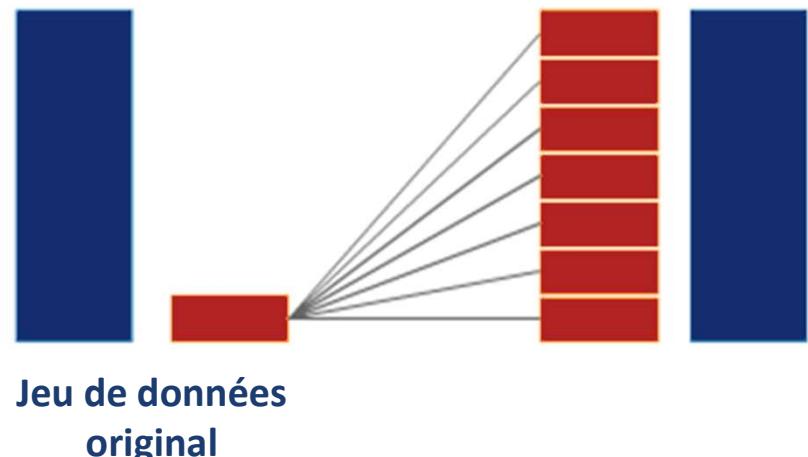


## Rééquilibrage de la cible :

① **Undersampling** = échantillonnage de la classe majoritaire

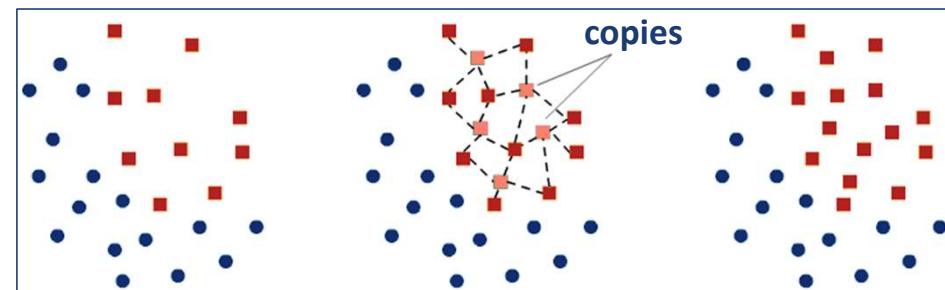


② **Oversampling** = copies de la classe minoritaire



### SMOTE (Synthetic Minority Oversampling Technique)

Consiste à synthétiser des éléments pour la classe minoritaire, à partir de ceux qui existent déjà en choisissant aléatoirement un point de la classe minoritaire et à calculer les k plus proches voisins de ce point.



③ **Modèle** : on peut indiquer à certains modèles le déséquilibre en réglant un hyperparamètre :  
exemple : « **class\_weight = 'balanced'** » pour LightGBM.

*cf annexe resampling : SMOTE*

## Comparaison de modèles

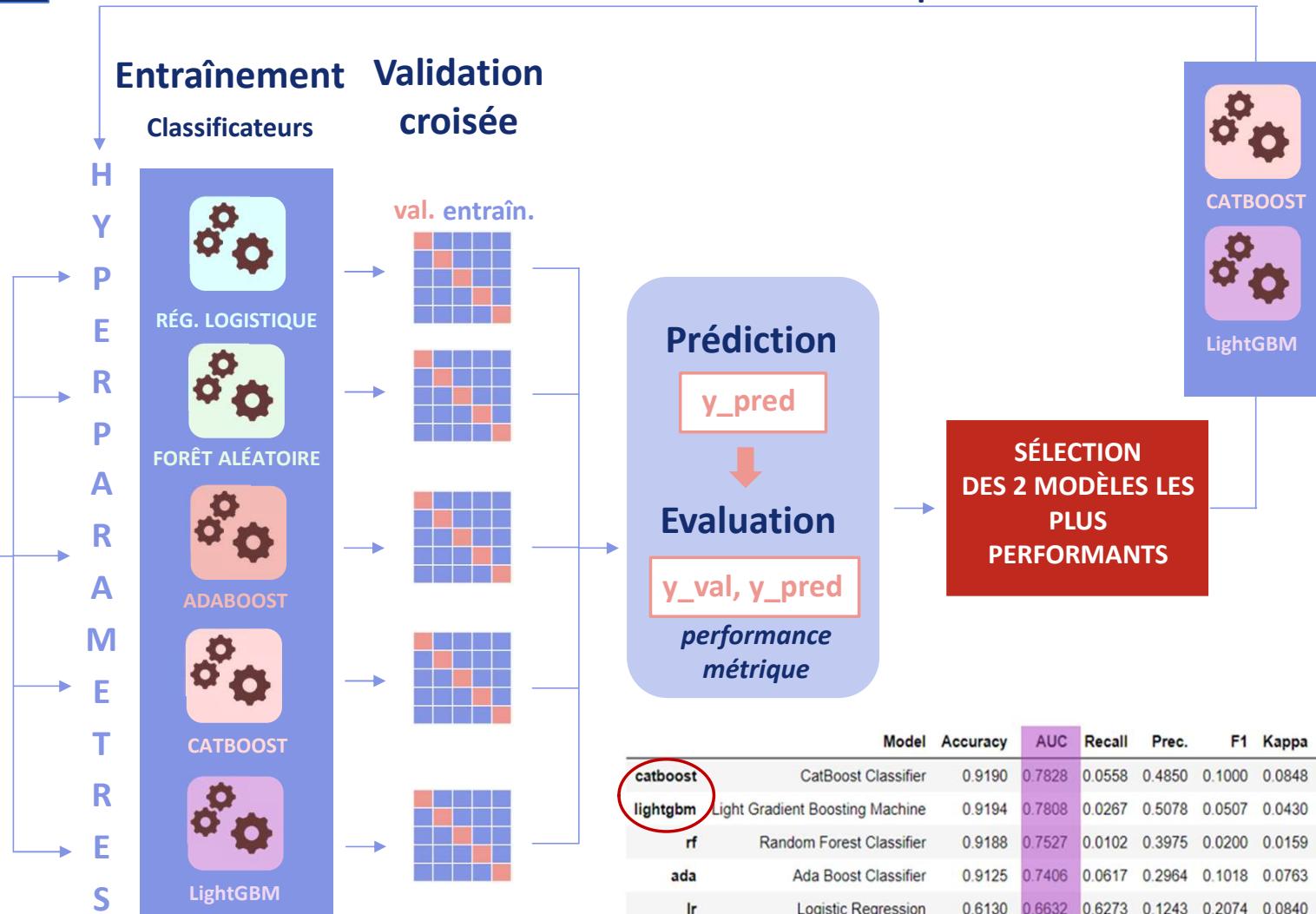
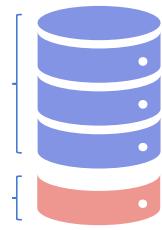
PYCARET

## Optimisation

### Jeu d'entraînement

entraînement

validation



## Problématique métier, métrique et fonction coût

Dans notre problème de classification binaire, le coût des **faux positifs** n'est pas le même que celui des **faux négatifs**.

Matrice de confusion		Classe prédite	
		Classe 0 : non défaillant	Classe 1 : défaillant
Classe réelle	Classe 0 : non défaillant	Vrais négatifs TN ✓	Faux positifs FP
	Classe 1 : défaillant	Faux négatifs FN	Vrais positifs TP ✓

### Minimiser les pertes:

**faux positifs** : non défaillants prédis défaillants

⇒ l'organisme prêteur perd les intérêts que le prêt aurait générés.

**faux négatifs** : défaillants prédis non défaillants

⇒ l'organisme prêteur perd ainsi la somme prêtée.

Un intérêt plus grand sera porté aux **faux négatifs**, encore plus coûteux que les **faux positifs**

CREATION D'UNE METRIQUE ET D'UNE FONCTION COÛT

Choix arbitraire de pénalisation :

- Mauvais prêts : pénalisation de -10
- Bons prêts : gain de 1

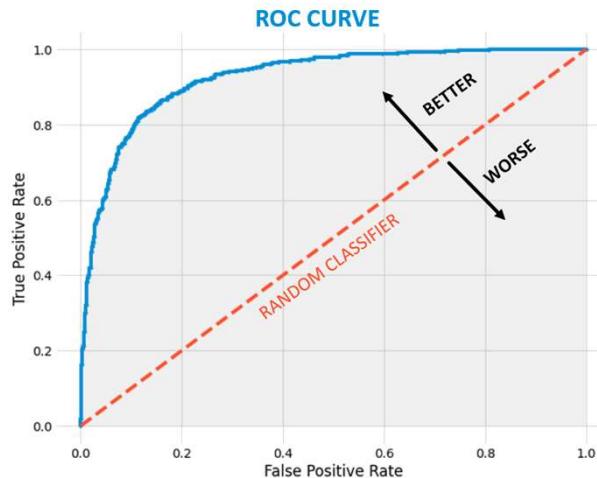
## Optimisation du meilleur modèle

**LightGBM**, plus rapide, retenu. **Optimisation** : selon 2 métriques et sur 2 jeux de données

### METRIQUES

1

Métrique : **Score AUC** (aire sous la courbe).  
Plus le modèle est performant, plus l'aire sous la courbe est maximisé.



2

Métrique : « **bancaire** » créée par nos soins,  
permettant de pénaliser les erreurs les plus  
coûteuses et donc limiter les pertes.

### JEUX DE DONNEES

1

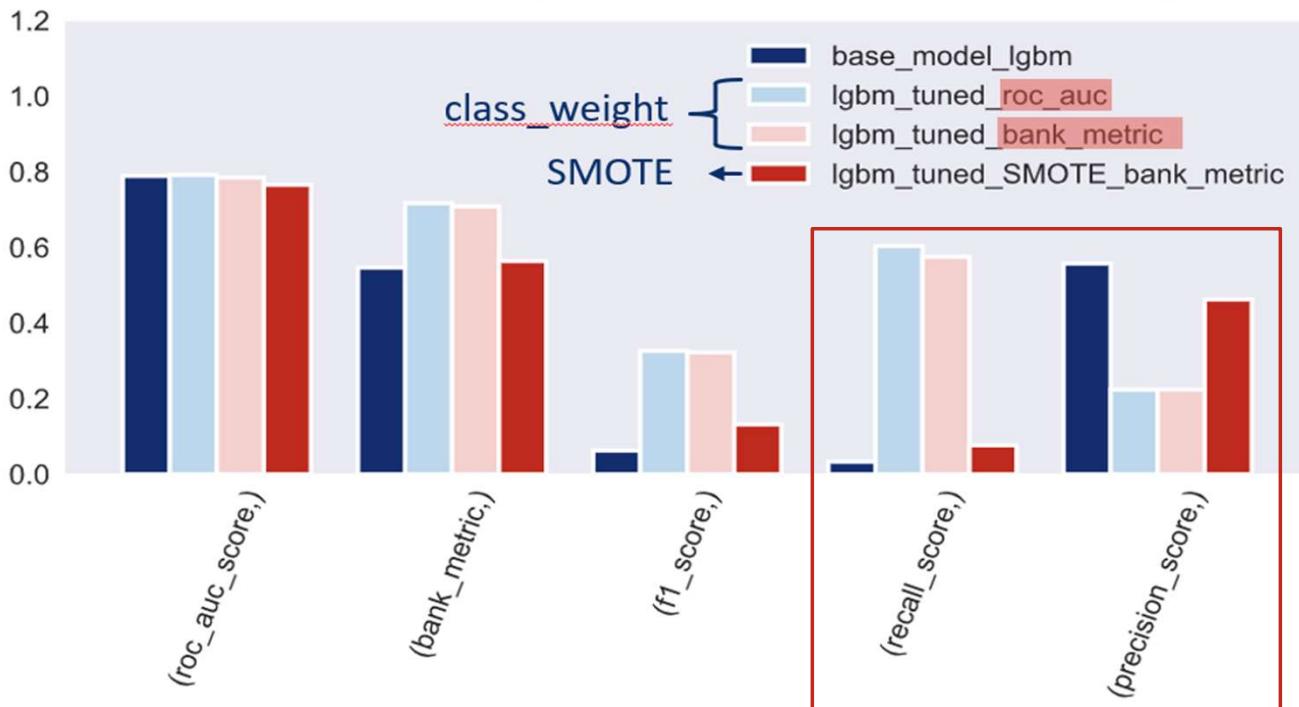
Jeu de données **rééquilibré**  
avec **SMOTE (OVERSAMPLING)**

2

Jeu de données  
**non** rééquilibré  
Réglage LightGBM :  
« **class\_weight = 'balanced'** »

## Optimisation du meilleur modèle

Scores pour LightGBM (jeu de validation) seuil par défaut (0.5)



## Conclusions

### 1. Rappel et précision

But dans le cas d'une classification de prêts bons ou mauvais :

**Maximiser le rappel au détriment de la précision** (diminuer les faux négatifs pour augmenter le rappel)

### 2. LightGBM avec l'hyperparamètre `class_weight = 'balanced'` donne de meilleurs résultats.

C'est la **stratégie de rééquilibrage** que nous choisirons.

## Meilleur modèle et seuil de solvabilité

		Confusion Matrix lgbm_tuned_bank_metric	
		Non défaillants	Défaillants
Classe réelle	Non défaillants	46656	9882
	Défaillants	2109	2856
		Non défaillants	Défaillants
		Classe prédictive	Classe prédictive

		Confusion Matrix lgbm_tuned_roc_auc	
		Non défaillants	Défaillants
Classe réelle	Non défaillants	46144	10394
	Défaillants	1967	2998
		Non défaillants	Défaillants
		Classe prédictive	Classe prédictive

### Conclusion

**faux négatifs :** LightGBM « métrique bancaire » < LightGBM ROC\_AUC ↳ perte de la somme prêtée

**faux positifs :** LightGBM « métrique bancaire » > LightGBM ROC\_AUC ↳ perte des intérêts



MODÈLE CONSERVÉ :  
LightGBM roc\_auc

Métrique bancaire  
utilisée pour fixer le  
seuil de solvabilité

## Meilleur modèle et seuil de solvabilité

non défaillant

**SEUIL = 0,41**

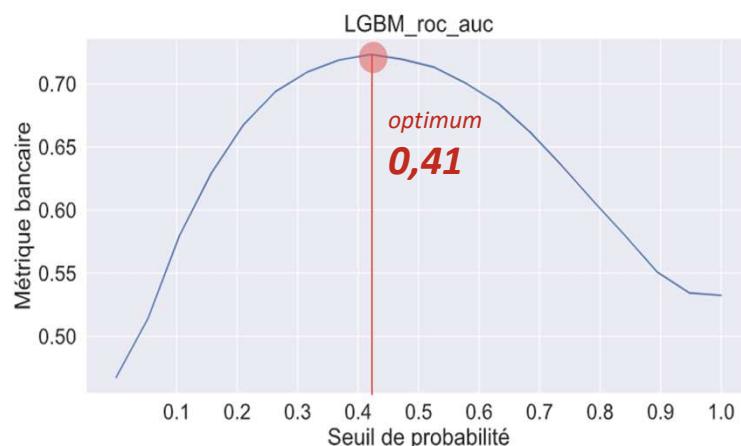
défaillant

0

1

**MODÈLE CONSERVÉ :**  
LightGBM roc\_auc

Métrique bancaire  
utilisée pour fixer le  
seuil de solvabilité



Classe réelle

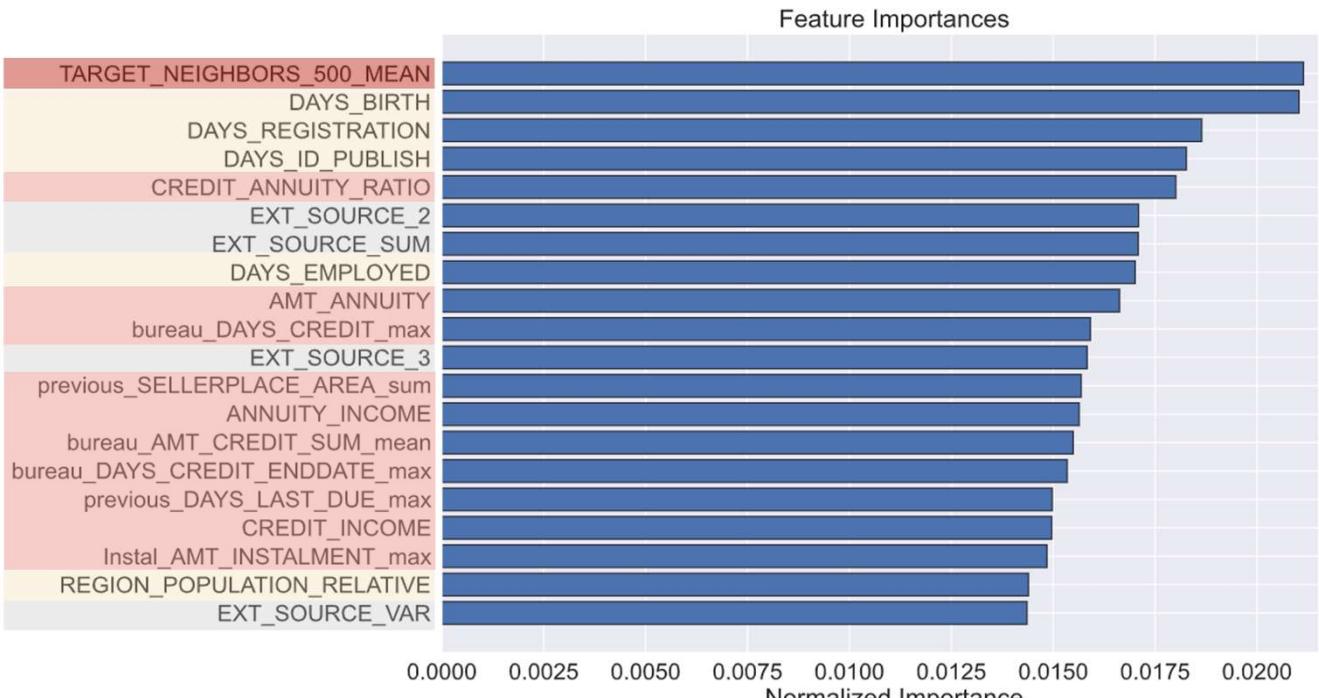
Non défaillants  
Défaillants

Confusion Matrix LGBM\_roc\_auc

		Non défaillants	Défaillants
Classe réelle	Non défaillants	41670	14868
	Défaillants	1458	3507
Classe prédictive			

## Modèle optimisé et interprétabilité LGBMClassifier

	Défaut	Optimisé
n_estimators	100	<b>10 000</b>
learning_rate	0,1	<b>0,05</b>
objective	None	<b>'binary'</b>
class_weight	None	<b>'balanced'</b>
boosting_type	'gbdt'	<b>'gbdt'</b>
num_leaves	31	<b>48</b>
max_depth	-1	<b>11</b>
min_split_gain	0	<b>0,1</b>
min_child_weight	0,001	<b>80</b>
min_child_samples	20	<b>18</b>
subsample	1	<b>0,73</b>
colsample_bytree	1	<b>0,67</b>
reg_alpha	0	<b>0,3</b>
gamma	0	<b>0,15</b>



lgbm_tuned_roc_auc	
roc_auc_score	<b>0.791848</b>
bank_metric	0.716879
f1_score	0.326633
recall_score	0.603827
precision_score	0.223865

Variables bancaires

Variables personnelles

Variables externes

cf annexe feature importance

# Dashboard

Présentation  
Déploiement



GitHub



Streamlit

## SIDEBAR

### INFORMATIONS CLIENTS

âge, sexe,  
situation familiale,  
ancienneté,  
revenu

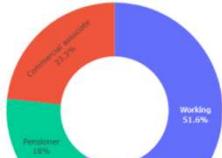
### INFOS GRAPHIQUES

Infos graphiques et  
statistiques supplémentaires  
tirés de l'analyse  
exploratoire de données.

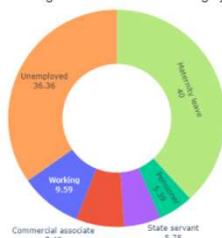
#### Stats : income type

Income type of the selected client : State servant

#### Distribution of income type



#### Percentage of defaulters for each category of income type



apparaît dans l'application principale

**Prêt à dépenser**

Now you can

**Client informations**

- Age: 46 years
- Gender: Male
- Family status: Married
- Education: Secondary / secondary special
- Years employed: 3 years
- Income type: State servant
- Income: 112500 \$
- Contract type: Cash loans

**More informations**

- Age
- Gender
- Family status
- Education
- Years employed
- Income type
- Income
- Contract type

**SHAP explainer**

Explain Results by SHAP

## PRET A DEPENSER DASHBOARD

"A timely return of the loan makes it easier to borrow a second time."

**1 Client ID**  
Please select a client ID  
398791

SK_ID_CURR	Prediction	Prediction_Score	GENDER	YEARS_BIRTH	YEARS_EMPLOYED	
6	398791	1	61.79063761395253	Male	46.25753424657535	3.85

**2 Default Risk Score**  
Select the threshold: (default: 0.41)

Threshold: 0.00 1.00

Prediction of the selected client with the current threshold :: Defaulter

**b** Prediction Score: 61.8

**c** ▲ 27.7

TRUST score for the selected client : LOW

Prediction Score for similar clients : 34.1

## APPLICATION PRINCIPALE

### 1 Sélection d'un client

### 2 Risque de défaut de paiement

### a Réglage du seuil de solvabilité

Seuil réglable de 0 à 1 (défaut : 0,41)

### b Jauge de prédition

Score de prédition de 0 à 100 associé à un qualificatif  
0-20: excellent, 20-40 good, 40-60 average, 60-80 low, 80-100 weak

### c Comparaison avec le score des 20 plus proches voisins

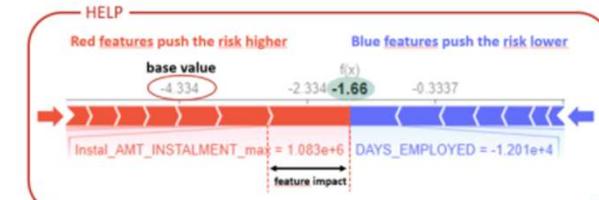
## SHAP

apparaît dans l'application principale

### SHAP : explain results

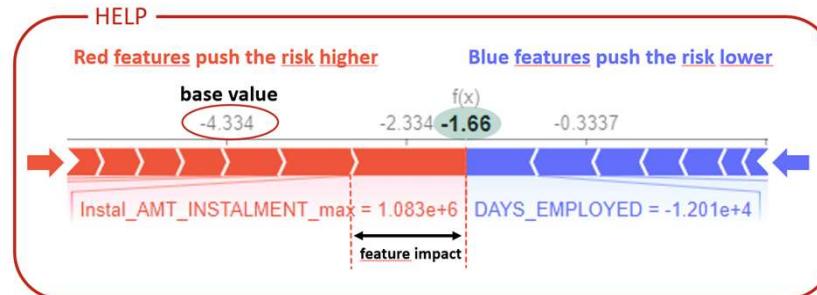
How most important features impacts Class prediction?  
Force plot shows, how opposite are the features strengths

#### SHAP HELP

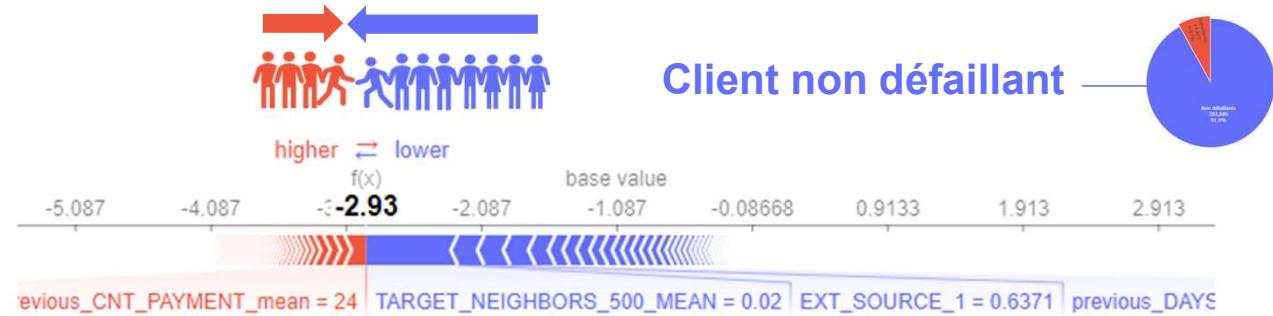


#### SHAP Force plot for the selected client





*Exemple avec deux clients*



# Démo



Dépôt github :

[https://github.com/Rabenco/App\\_credit](https://github.com/Rabenco/App_credit)



En local :

app\_credit.py

A distance :

[https://share.streamlit.io/rabenco/app\\_credit/main/app\\_credit.py](https://share.streamlit.io/rabenco/app_credit/main/app_credit.py)

main 1 branch 0 tags Go to file Add file Code

Rabenco Update requirements.txt 20a51ae yesterday 14 commits

- Images Add files via upload yesterday
- README.md Update README.md yesterday
- app\_credit.py Update app\_credit.py yesterday
- best\_model.joblib Add files via upload yesterday
- data\_final.csv Add files via upload yesterday
- requirements.txt Update requirements.txt yesterday

README.md

## App\_credit

La société « Prêt à dépenser » propose des crédits à la consommation. Notre étude vise à développer un modèle de scoring de la probabilité de défaut de paiement du client pour étayer la décision d'accorder ou non un prêt à un client, en s'appuyant sur des sources de données variées. Dans un souci de transparence, l'entreprise souhaite également développer un tableau de bord (dashboard) interactif pour que les chargés de clientèle puissent à la fois expliquer les décisions d'octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

SIDE BAR

INFORMATIONS CLIENTS

Client informations

INFOGRAPHIQUES

PRET A DEPENSER DASHBOARD

APPLICATION PRINCIPALE

- 1 Sélection d'un client
- 2 Risque de défaut de paiement
- 3 Réglage du seuil de solvabilité
- 4 Jouage de prédition
- 5 Comparaison avec le score des 20 plus proches voisins

SHAP : explain results

[https://share.streamlit.io/rabenco/app\\_credit/main/app\\_credit.py](https://share.streamlit.io/rabenco/app_credit/main/app_credit.py)

# Conclusion

## Conclusion

**Notre étude** portait sur un problème de **classification binaire présentant un déséquilibre de classe**.

**Modèle final** : **LightGBM** optimisé sur la métrique **ROC\_AUC**.

**Mise en place** de stratégies pour optimiser le meilleur modèle et obtenir une performance maximale:

- **différentes solutions de rééquilibrage de classe** testées et comparées
- **création de nouvelles variables** facilement explicables (demande client)
- **création d'une métrique métier** et fixation d'un **seuil de solvabilité optimum**.

## Améliorations possibles

- **Optimisation** plus fine des hyperparamètres du modèle
- **Modification** de la métrique créée, avec l'aide d'un expert métier
- **Création de variables** plus pertinentes avec l'expert

# Annexes

## Aberrations détectées

## Alignement des 2 jeux de données

```
feature CODE_GENDER has different values: {'XNA'}  
feature NAME_INCOME_TYPE has different values: {'Maternity leave'}  
feature NAME_FAMILY_STATUS has different values: {'Unknown'}
```

### Variable 'CODE\_GENDER'

Le jeu d'entraînement contient seulement 4 valeurs nommés 'XNA' pour la colonne renseignant le genre.

### Variable 'NAME\_INCOME\_TYPE'

La colonne 'NAME\_INCOME\_TYPE' prend la valeur 'Maternity leave' uniquement pour le jeu d'entraînement , et pour seulement 5 emprunteurs.

**Variable 'NAME\_FAMILY\_STATUS'** De la même manière, pour la colonne NAME\_FAMILY\_STATUS, il y a seulement deux fois la valeur Unknown et uniquement pour le jeu d'entraînement.

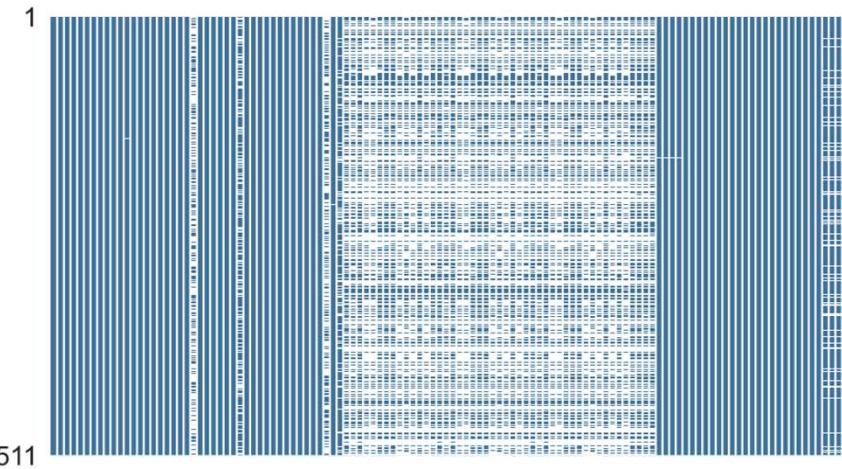
## Correction des aberrations détectées lors de l'EDA

# Suppression des aberrations détectées chez la variable "**"DAYS EMPLOYED "**

# Suppression des aberrations détectées chez les variables "**"OBS"**

## Valeurs manquantes

	Nombre de valeurs manquantes	% de valeurs manquantes
COMMONAREA_MODEI	214865	69.870000
COMMONAREA_MODE	214865	69.870000
COMMONAREA_AVG	214865	69.870000
NONLIVINGAPARTMENTS_MODE	213514	69.430000
NONLIVINGAPARTMENTS_MEDI	213514	69.430000
NONLIVINGAPARTMENTS_AVG	213514	69.430000
FONDKAPREMONT_MODE	210295	68.390000
LIVINGAPARTMENTS_MEDI	210199	68.350000
LIVINGAPARTMENTS_AVG	210199	68.350000
LIVINGAPARTMENTS_MODE	210199	68.350000
FLOORSMIN_MODE	208642	67.850000
FLOORSMIN_AVG	208642	67.850000
FLOORSMIN_MEDI	208642	67.850000
YEARS_BUILD_MEDI	204488	66.500000
YEARS_BUILD_MODE	204488	66.500000
YEARS_BUILD_AVG	204488	66.500000
OWN_CAR_AGE	202929	65.990000
LANDAREA_AVG	182590	59.380000
LANDAREA_MODE	182590	59.380000
LANDAREA_MEDI	182590	59.380000
BASEMENTAREA_MEDI	179943	58.520000
BASEMENTAREA_MODE	179943	58.520000
BASEMENTAREA_AVG	179943	58.520000
EXT_SOURCE_1	173378	56.380000
NONLIVINGAREA_MODE	169682	55.180000
NONLIVINGAREA_AVG	169682	55.180000
NONLIVINGAREA_MEDI	169682	55.180000
ELEVATORS_MODE	163891	53.300000
ELEVATORS_AVG	163891	53.300000
ELEVATORS_MEDI	163891	53.300000



## Suppression des colonnes

NAN > 60%

## Variables supprimées

```
[ 'OWN_CAR_AGE', 'YEARS_BUILD_AVG', 'COMMONAREA_AVG',
  'FLOORSMIN_AVG', 'LIVINGAPARTMENTS_AVG',
  'NONLIVINGAPARTMENTS_AVG', 'YEARS_BUILD_MODE',
  'COMMONAREA_MODE', 'FLOORSMIN_MODE',
  'LIVINGAPARTMENTS_MODE', 'NONLIVINGAPARTMENTS_MODE',
  'YEARS_BUILD_MEDI', 'COMMONAREA_MEDI', 'FLOORSMIN_MEDI',
  'LIVINGAPARTMENTS_MEDI', 'NONLIVINGAPARTMENTS_MEDI',
  'FONDKAPREMONT_MODE' ]
```

## Imputation : trois façons de procéder

3

1

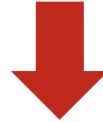
### Imputation BASIQUE

Qualitatives



Imputation avec une constante  
(Imputation avec le mode)

Quantitatives



Imputation avec la médiane

2

### Imputation BASIQUE PLUS

Qualitatives



Imputation avec une constante  
(Imputation avec le mode)

Quantitatives



1)Imputation avec la médiane  
2) EXT\_SOURCE : XGBRegressor

3

Librairie verstack  
NaNImputer()

### Imputation AVANCÉE

Qualitatives



Imputation avec  
XGBClassifier

Quantitatives



Imputation avec  
XGBRegressor

### Méthodes de FEATURE SELECTION

#### FILTRAGE

*Corrélation de Pearson*  
*Chi 2*  
*F Test*  
*ANOVA*  
*Information Gain*

#### WRAPPER

*Boruta*  
*BoostAroota*  
*Heuristics:*  
*Forward Selection,*  
*Backward Elimination,*  
*Recursive Feature Elimination*  
*Methodical:*  
*Best First Search, DFS*  
*Stochastic:*  
*Random Hill Climbing,*  
*Simulated Annealing*

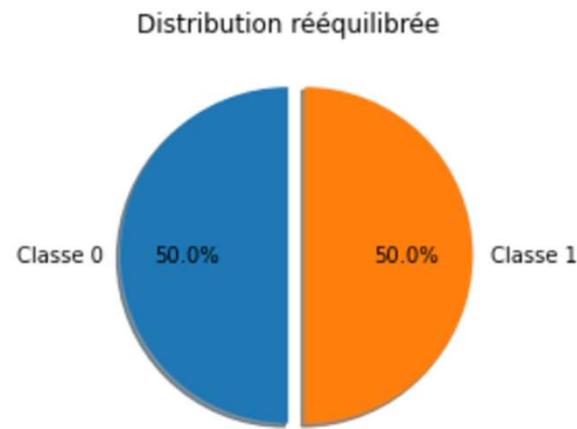
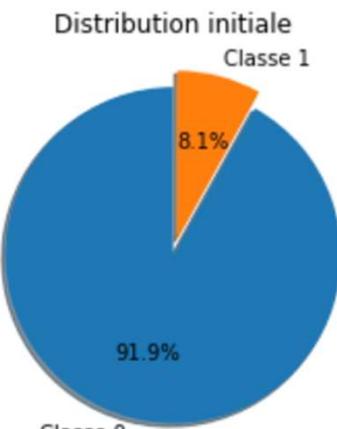
#### EMBEDDED

*Lasso Regression*  
*Ridge Regression*  
*Elastic Nets*  
*Decision Trees*  
*RF*  
*LightGBM*

La feature selection est un processus de sélection d'un sous-ensemble de variables qui sont les plus pertinentes pour la modélisation et l'objectif commercial du problème.

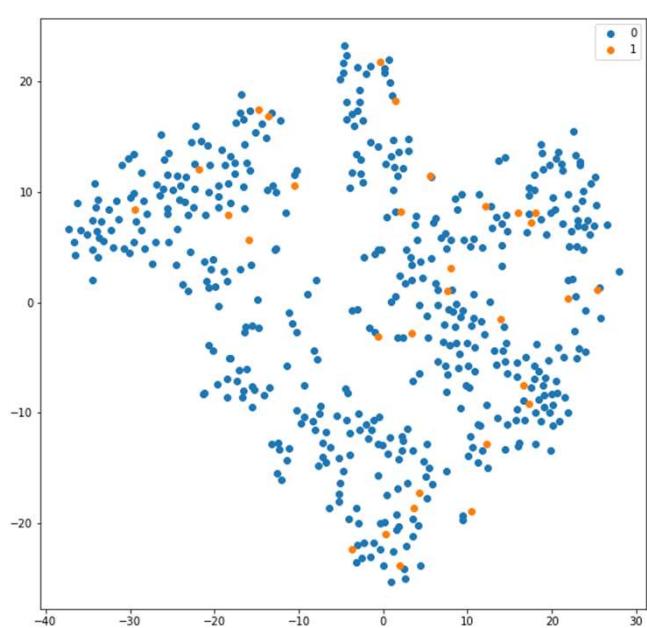
## Rééquilibrage de la cible : resampling

- Non défaillants
- Défaillants



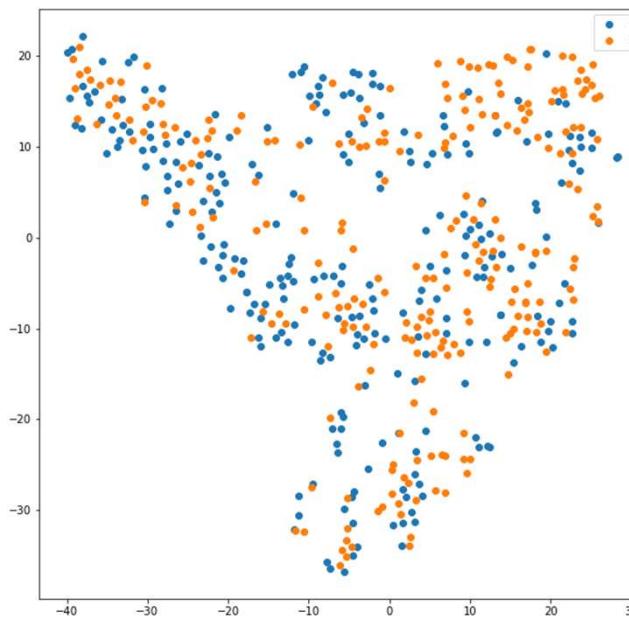
Initial

Échantillon : 500  
1 0.10  
0 0.90



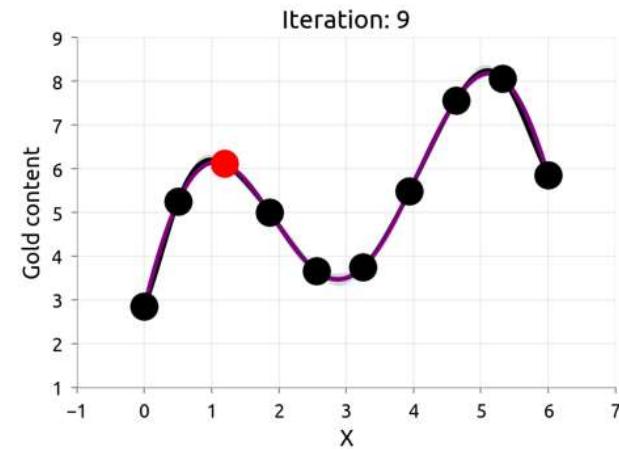
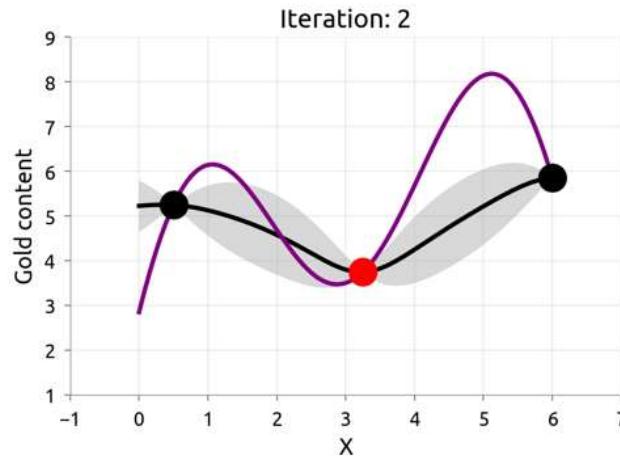
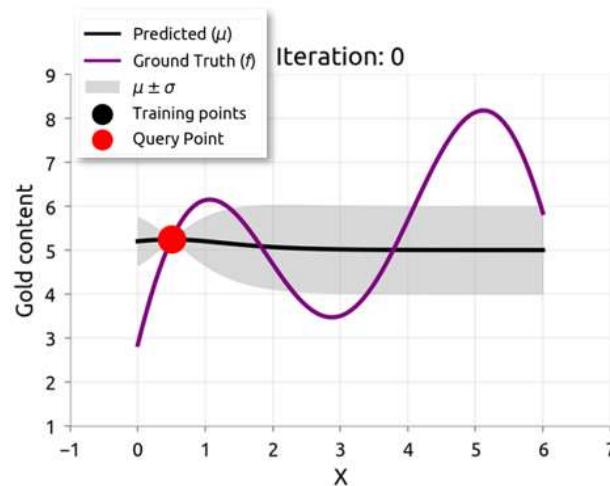
SMOTE

Échantillon : 500  
1 0.45  
0 0.55



## Optimisation du meilleur modèle

## Optimisation bayésienne



+ Trace gardée des résultats d'évaluation passés  
pour former un modèle probabiliste

L'optimisation bayésienne construit un modèle de probabilité de la fonction objectif afin de **proposer des choix plus intelligents pour le prochain ensemble d'hyperparamètres à évaluer**. Au fur et à mesure que le nombre d'observations augmente, la distribution postérieure s'améliore et l'algorithme devient plus sûr des régions de l'espace des paramètres qui valent la peine d'être explorées et de celles qui ne le sont pas.

## Optimisation modèle catboost

Catboost	Hyperparamètres	Description	Note	Nos hyperparamètres
Number of trees	iterations	Le nombre maximum d'arbres qui peuvent être construits lors de la résolution de problèmes d'apprentissage automatique.		iterations: 323
Learning rate	learning_rate	Le taux d'apprentissage.	Ce paramètre est utilisé pour réduire le pas du gradient. Il affecte le temps global de formation : plus la valeur est petite, plus le nombre d'itérations nécessaires à la formation est élevé.	learning_rate: 0.0673344419215237
Tree depth	depth	Profondeur de l'arbre.	La profondeur optimale varie de 4 à 10. Les valeurs comprises entre <b>6 et 10</b> sont recommandées.	depth: 8
L2 regularization	l2_leaf_reg	Coefficient au niveau du terme de régularisation L2 de la fonction de coût.	Toute valeur positive est autorisée.	l2_leaf_reg: 21
Random strength	random_strength	La quantité d'aléatoire à utiliser pour noter les divisions lorsque la structure de l'arbre est sélectionnée.	Utilisez ce paramètre pour éviter un ajustement excessif du modèle.	random_strength: 3.230824361824754e-06
Bagging temperature	bagging_temperature	Définit les paramètres du bootstrap bayésien.	Utilisez le bootstrap bayésien pour attribuer des poids aléatoires aux objets.	bagging_temperature: 0.41010395885331385
Border count	border_count Alias: max_bin	Le nombre de divisions pour les variables numériques.	CPU: 254	border_count : 254
Tree growing policy	min_data_in_leaf Alias: min_child_samples  max_leaves Alias: num_leaves	Le nombre minimum d'échantillons d'entraînement dans une feuille.  Le nombre maximum de feuilles dans l'arbre résultant.	CatBoost ne recherche pas de nouveaux splits dans les feuilles dont le nombre d'échantillons est inférieur à la valeur spécifiée.  Il n'est pas recommandé d'utiliser des valeurs supérieures à <b>64</b> , car cela peut ralentir considérablement le processus de formation.	
	scale_pos_weight	Le poids de la classe 1 dans la classification binaire. La valeur est utilisée comme un multiplicateur pour les poids des objets de la classe 1.		scale_pos_weight: 0.7421091918485163

## Optimisation modèle LGBM Classifier

LGBM	Hyperparamètres	Description	Note	Nos hyperparamètres
	num_estimators	Le nombre maximum d'arbres qui peuvent être construits lors de la résolution de problèmes d'apprentissage automatique.	utiliser un très grand nombre d'itérations si utilisation de l'arrêt précoce.	num_estimators: 10 000
	learning_rate	Le taux d'apprentissage.	En général, nous utilisons un taux d'apprentissage de 0,05 ou moins pour la formation, tandis qu'un taux d'apprentissage de 0,10 ou plus est utilisé pour modifier les hyperparamètres.	learning_rate: 0,05
	max_depth	Profondeur de l'arbre.	Une valeur plus grande est généralement meilleure, mais la vitesse d'overfitting augmente. Typique : 6, généralement [3, 12].	max_depth: 11
	lambda_l1 lambda_l2	Régularisation L1 pour le boosting Régularisation L2 pour le boosting		reg_alpha: 0,3 reg_lambda: 0,15
	colsample_bytree	Rapport de sous-échantillonnage des colonnes lors de la construction de chaque arbre.		colsample_bytree: 0,67
	subsample	Rapport de sous-échantillon de l'instance d'apprentissage.		subsample: 0,73
	num_leaves min_split_gain min_child_weight min_child_samples	<i>Le nombre maximum de feuilles dans l'arbre résultant.</i> Réduction de la perte minimale requise pour effectuer une partition supplémentaire sur un nœud feuille de l'arbre. Le nombre minimum d'échantillons d'entraînement dans une feuille. <i>Somme minimale de poids d'instance (hessian) nécessaire dans un enfant (feuille).</i> Nombre minimum de données nécessaires dans un enfant (feuille).	Cette technique est extrêmement utile lorsque vous essayez de construire des arbres profonds, mais que vous essayez également d'éviter de construire des branches inutiles de ces arbres (overfitting).	num_leaves: 48 min_split_gain: 0,1 min_child_weight: 80 min_child_samples: 18
	objective	<b>Binary</b> Description : Application sigmoïde comme fonction d'activation. Entropie croisée comme fonction de perte.		binary

## Feature importance : variables importantes LightGBM

	<b>TARGET_NEIGHBORS</b>	Valeur TARGET moyenne des 500 voisins les plus proches de chaque ligne, où chaque voisinage est défini par les trois sources externes (1,2,3) et le ratio crédit/intérêts.	<b>Variables bancaires</b>
	<b>DAYS_BIRTH</b>	Âge du client en jours au moment de la demande	
	<b>DAYS_REGISTRATION</b>	Combien de jours avant la demande le client a-t-il modifié <b>son inscription</b> ?	<b>Variables personnelles</b>
	<b>DAYS_ID_PUBLISH</b>	<b>Combien de jours</b> avant la demande le client a-t-il modifié le document d'identité avec lequel il avait demandé le prêt?	
	<b>CREDIT_ANNUITY_RATIO</b>	<b>Ratio</b> montant du crédit/annuité	<b>Variables externes</b>
	<b>EXT_SOURCE_2</b>	Score normalisé provenant d'une source de données externe	
	<b>EXT_SOURCE_SUM</b>	Somme des 3 variables EXT_SOURCE	
	<b>DAYS_EMPLOYED</b>	<b>Combien de jours</b> avant la demande la personne a-t-elle commencé son emploi actuel	
	<b>AMT_ANNUITY</b>	Intérêts du prêt	
	<b>DAYS_CREDIT_max</b>	<b>Combien de jours</b> avant la demande actuelle le client a-t-il fait une demande de crédit auprès de Home Credit	<b>Feature engineering</b>
	<b>EXT_SOURCE_3</b>	Score normalisé provenant d'une source de données externe	
	<b>SELLERPLACE_AREA_sum</b>	Zone de vente du <b>lieu de vente</b> de la demande précédente	
	<b>ANNUITY_INCOME</b>	Ratio intérêt/revenu du client	
	<b>AMT_CREDIT_SUM_mean</b>	Score normalisé provenant d'une source de données externe	
	<b>DAYS_CREDIT_ENDDATE_max</b>	<b>Durée restante</b> du crédit CB (en jours) au moment de la demande chez Home Credit	

- app\_train
- bureau
- previous