IMPLÉMENTEZ UN MODÈLE DE SCORING

Prêt à dépenser

Alexandre Delaguillaumie

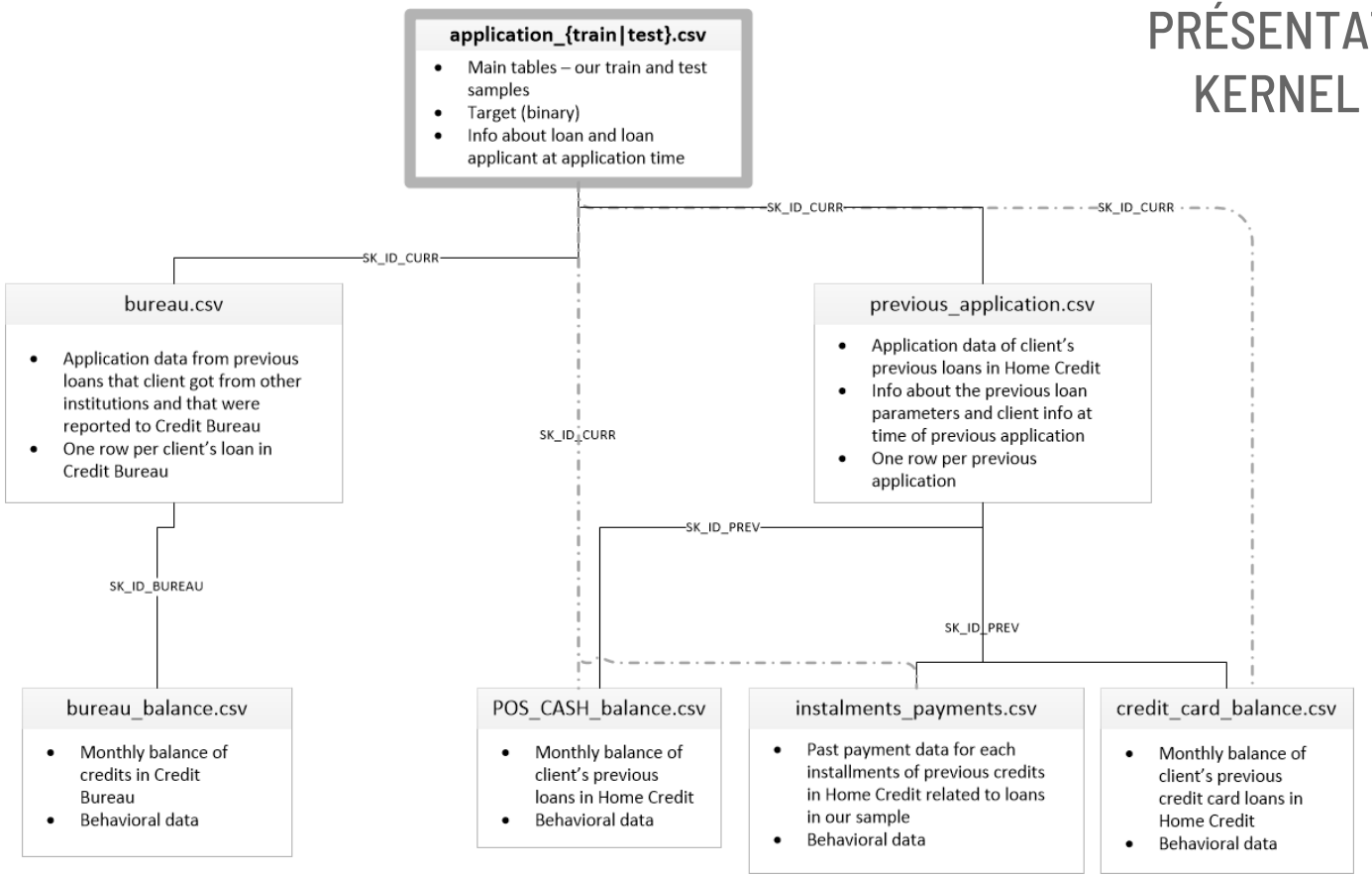
OBJECTIFS

Construire un modèle de scoring qui donnera une prédiction sur la probabilité de faillite d'un client de façon automatique

Construire un dashboard interactif à destination des gestionnaires de la relation client

Mettre en production le modèle de scoring de prédiction à l'aide d'une API et du dashboard

PRÉSENTATION DU KERNEL KAGGLE



PRÉSENTATION DU KERNEL KAGGLE

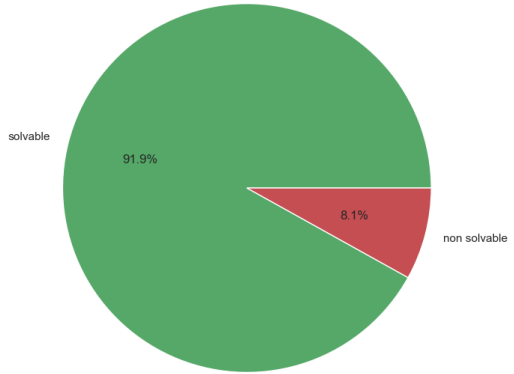
DIMENSIONS

application_train shape :----- (307 511, 122)
application_test shape :------ (48 744, 121)
bureau shape :---------- (1 716 428, 17)
bureau_balance shape :-------- (27 299 925, 3)
POS_CASH_balance shape :------ (10 001 358, 8)
credit_card_balance shape :--- (3 840 312, 23)
installments_payments shape :- (13 605 401, 8)
previous_application shape:--- (1 670 214, 37)

NUMBER OF NAN

application_train shape :----- 9 152 465
application_test shape :------ 1 404 419
bureau shape :---------- 3 939 947
bureau_balance shape :-------- 0
POS_CASH_balance shape :------ 52 158
credit_card_balance shape :--- 5 877 356
installments_payments shape :- 5 810
previous_application shape:--- 11 109 336

DÉSÉQUILIBRE ENTRE LES CLASSES



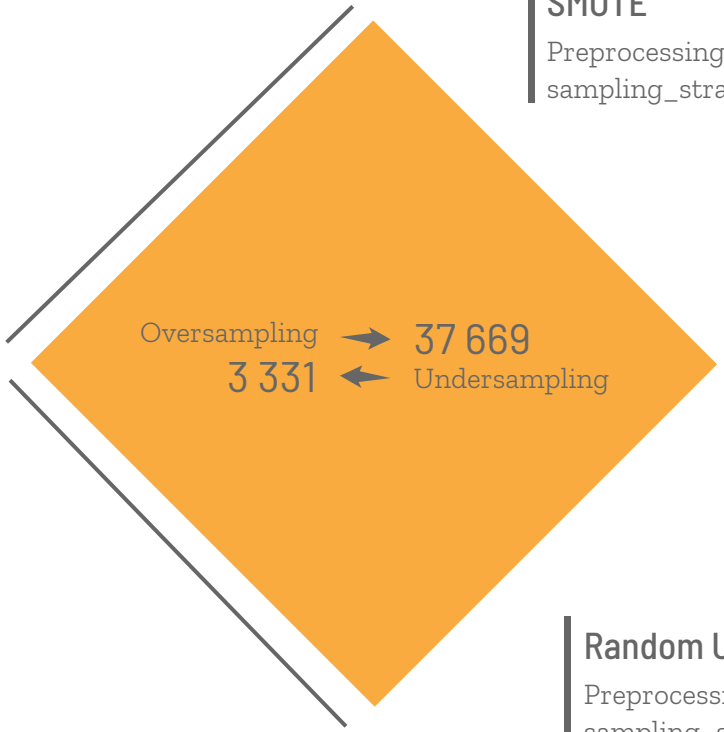
PREPROCESSING & FEATURE ENGINEERING

- Feature engineering et preprocess des tables
- Merge des tables
- Traitement des outliers (inf)
- Suppression des NaN (>45%)
- Sampling + train_test_split
- Pipeline de preprocessing

LE TRAITEMENT DU DÉSÉQUILIBRE DES CLASSES

SMOTE

Preprocessing +
sampling_strategy= 'minority'



STRATÉGIE

3 jeux de données splittés seront utilisés et comparés pour connaître la meilleure stratégie : SMOTE, RUS et un split désequilibrer pour comparer

Random Under Sampling

Preprocessing +
sampling_strategy= majority'

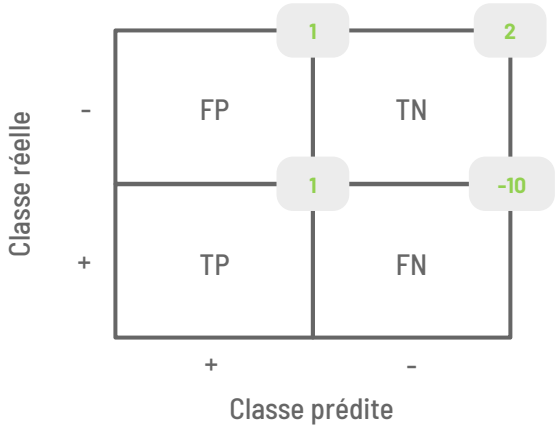
FONCTION COÛT

Principes et tuning

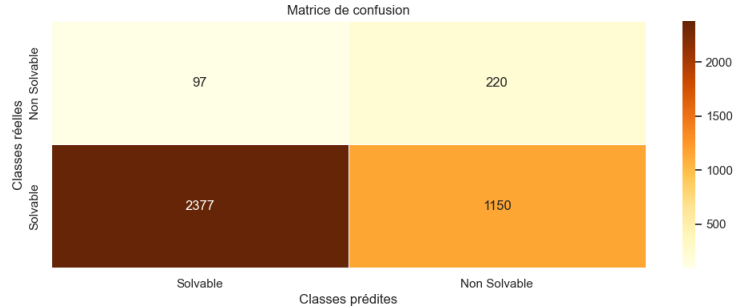
RAPPEL :

Classe 0 négative : NON DÉFAILLANT

Classe 1 positive : DÉFAILLANT



MATRICE DE CONFUSION



FONCTION D'ÉVALUATION

```
# Gain total
J = tp*tp_value + tn*tn_value + fp*fp_value + fn*fn_value

# Gain maximum
max_J = (fp + tn)*tn_value + (fn + tp)*tp_value

# Gain minimum
min_J = (fp + tn)*fp_value + (fn + tp)*fn_value

# Gain normalisé entre 0 et 1
J_normalized = (J - min_J)/(max_J - min_J)
```

MÉTRIQUES

ACCURACY

$$\frac{\text{VRAI}}{\text{ENSEMBLE}}$$

ROC AUC

F-beta SCORE

$$\frac{2*\text{precision} \,.\, \text{recall}}{\text{precision} + \text{recall}}$$

RECALL

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

PRECISION

$$\frac{TF}{TF + FP}$$

F1 SCORE

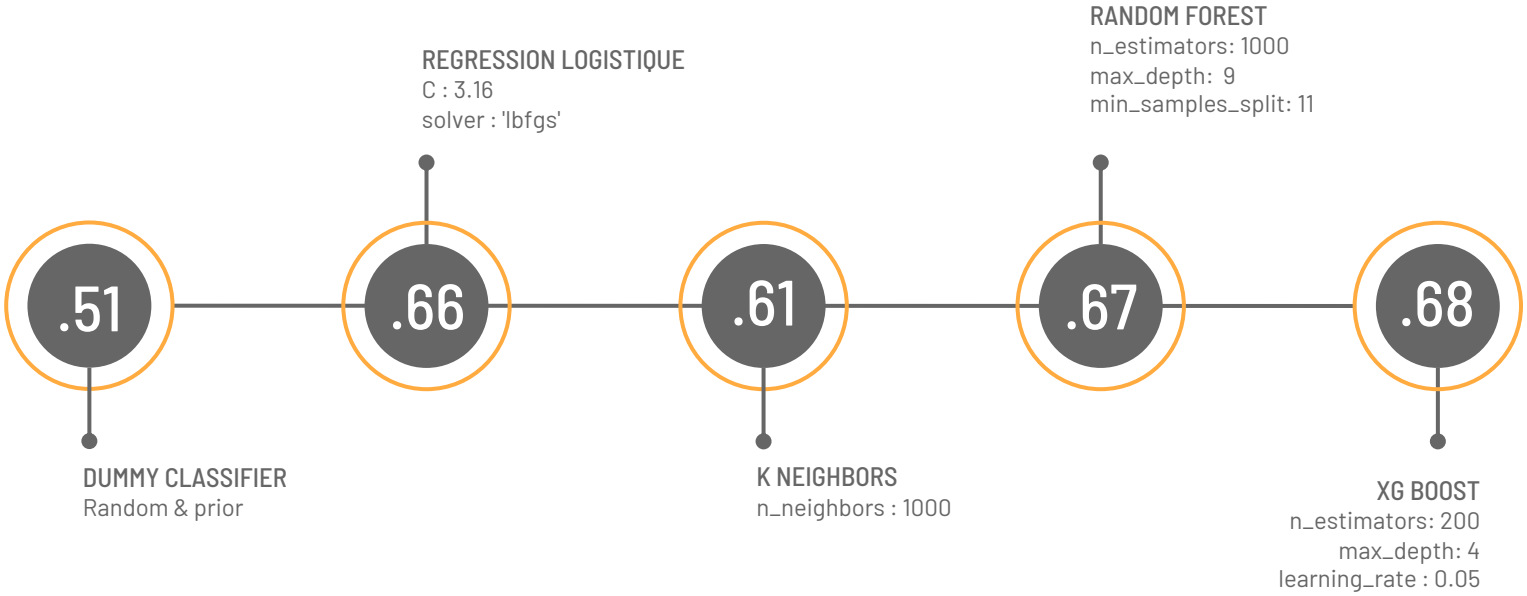
$$\frac{\text{precision} \,.\, \text{recall}}{\text{precision} + \text{recall}}$$

BALANCE ACCURACY

$$\frac{1}{2}\frac{TP}{TP + FN} + \frac{1}{2}\frac{TN}{TN + FP} = 50\%$$

GRIDSEARCHCV POUR DÉTERMINER L'ALGORITHME

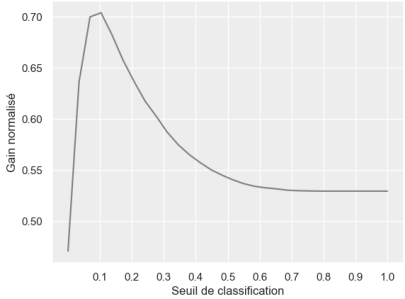
METRIC : rocauc



PARAMÈTRES FINAUX

metrics & params

THRESHOLD

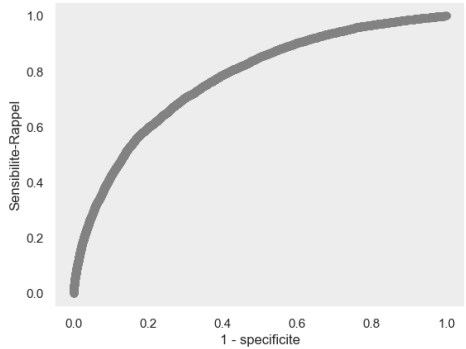


.70

+0.016

XG BOOST
n_estimators: 500
max_depth: 7
learning_rate : 0.14

AUROC



métier	fbeta	f1	threshold
0.703	0.522	0.292	0.069
+0.02	+0.0001	+0.0031	

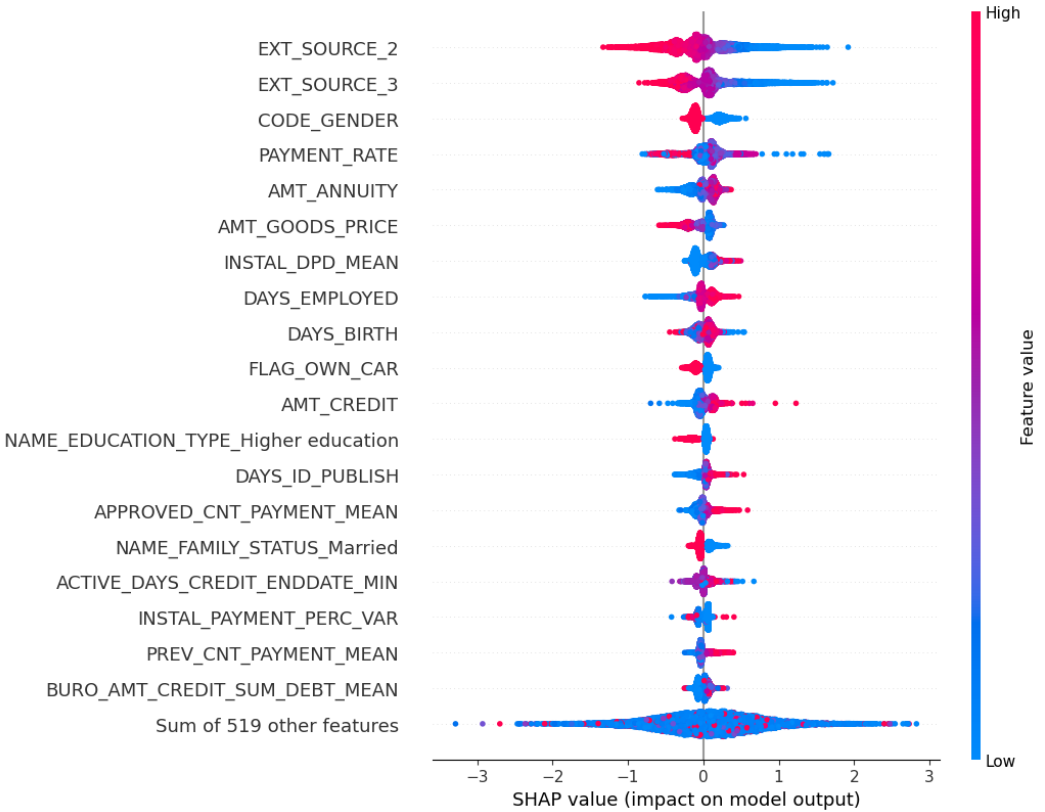
balance	accuracy
0.7	0.742
+0.0026	+0.066

```
y_pred = pipeline.predict_proba(X_test)
y_pred = (y_pred[:,1] >= threshold )*1
```

TABLEAU DE SYNTHÈSE DES RÉSULTATS

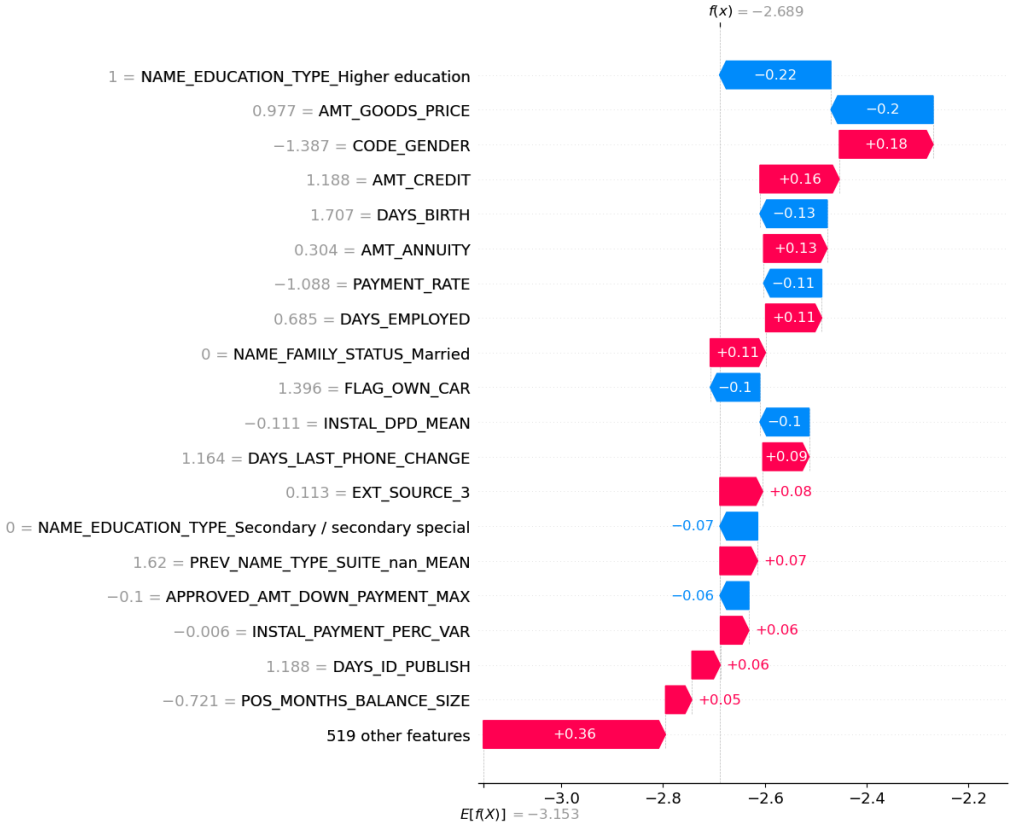
					Metrics				
	Run Name	Created	Duration	Experiment Name	accuracy_score	balance_accuracy	f1	fbeta_score	roc_auc
	xgb_threshold/0.10344827586206896	4 days ago	8.8s	XGBoost Models	0.233	0.567	0.171	0.5	0.567
	xgb_threshold/0.06896551724137931	6 hours ago	6.4s	XGBoost Models	0.742	0.7	0.292	0.522	0.7
	xgb_final/rus/2	10 hours ago	2.0h	XGBoost Models	0.676	0.684	0.261	0.521	0.684
	xgb_final/rus/1	4 days ago	1.2h	XGBoost Models	0.675	0.672	0.251	0.502	0.672
	xgb/smote	4 days ago	49.2min	XGBoost Models	0.917	0.505	0.024	0.014	0.505
	xgb/rus	4 days ago	6.6min	XGBoost Models	0.678	0.678	0.256	0.51	0.678
	xgb/imbalanced	4 days ago	22.4min	XGBoost Models	0.919	0.516	0.066	0.039	0.516
	rf/smote	4 days ago	58.0min	RandomF Models	0.89	0.55	0.176	0.149	0.55
	rf/rus	4 days ago	4.5min	RandomF Models	0.68	0.676	0.256	0.507	0.676
	rf/imbalanced	5 days ago	41.1min	RandomF Models	0.918	0.5	0	0	0.5
	logistic_regression/smote	5 days ago	1.4min	LogReg Models	0.73	0.652	0.253	0.449	0.652
	logistic_regression/rus	5 days ago	17.5s	LogReg Models	0.663	0.658	0.24	0.486	0.658
	logistic_regression/imbalanced	5 days ago	41.6s	LogReg Models	0.918	0.541	0.151	0.097	0.541
	knn/smote	5 days ago	21.0min	KNeighbors Models	0.093	0.505	0.152	0.473	0.505
	knn/rus	5 days ago	38.7s	KNeighbors Models	0.773	0.607	0.227	0.352	0.607
	knn/imbalanced	5 days ago	6.7min	KNeighbors Models	0.918	0.5	0	0	0.5
	dummy/uniform	4 days ago	4.7s	Dummy Models	0.507	0.507	0.144	0.337	0.507
	dummy/uniform	4 days ago	9.6s	Dummy Models	0.507	0.507	0.144	0.337	0.507
	dummy/uniform	4 days ago	7.5s	Dummy Models	0.507	0.507	0.144	0.337	0.507
	dummy/prior/smote	5 days ago	4.7s	Dummy Models	0.918	0.5	0	0	0.5
	dummy/prior/rus	5 days ago	4.7s	Dummy Models	0.918	0.5	0	0	0.5
	dummy/prior/imbalanced	5 days ago	5.5s	Dummy Models	0.918	0.5	0	0	0.5

INTERPRÉTABILITÉ GLOBALE ET LOCALE DU MODÈLE



Interprétabilité locale

INTERPRÉTABILITÉ GLOBALE ET LOCALE DU MODÈLE



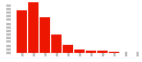
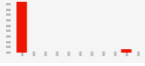
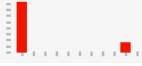
Client n°162308

Interprétabilité globale

ANALYSE DU DATA DRIFT

Dataset Drift is NOT detected

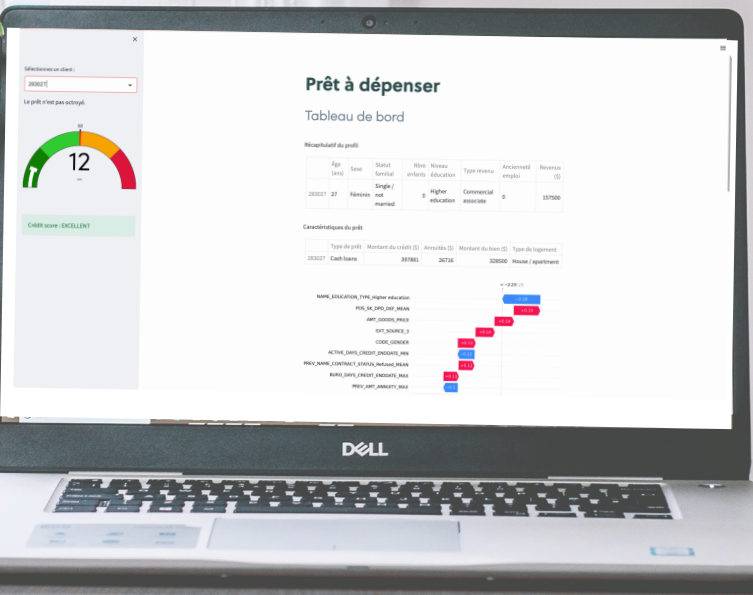
TESTS	SUCCESS	WARNING	FAIL	ERROR
347	326	0	21	0

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
AMT_REQ_CREDIT_BUREAU_QRT	num			Detected	Wasserstein distance (normed)	0.359052
AMT_REQ_CREDIT_BUREAU_MON	num			Detected	Wasserstein distance (normed)	0.281765
AMT_GOODS_PRICE	num			Detected	Wasserstein distance (normed)	0.210785
AMT_CREDIT	num			Detected	Wasserstein distance (normed)	0.207334
AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.161102
AMT_REQ_CREDIT_BUREAU_WEEK	num			Detected	Wasserstein distance (normed)	0.15426
NAME_CONTRACT_TYPE	cat			Detected	Jensen-Shannon distance	0.14755
DAYS_LAST_PHONE_CHANGE	num			Detected	Wasserstein distance (normed)	0.138977
FLAG_EMAIL	num			Detected	Jensen-Shannon distance	0.122121

DATADRIFT THRESHOLD	0.5
DRIFTED COLUMNS	7.5%

DASHBOARD

Lien http



PREUVE DE DÉPLOIEMENT PYTEST

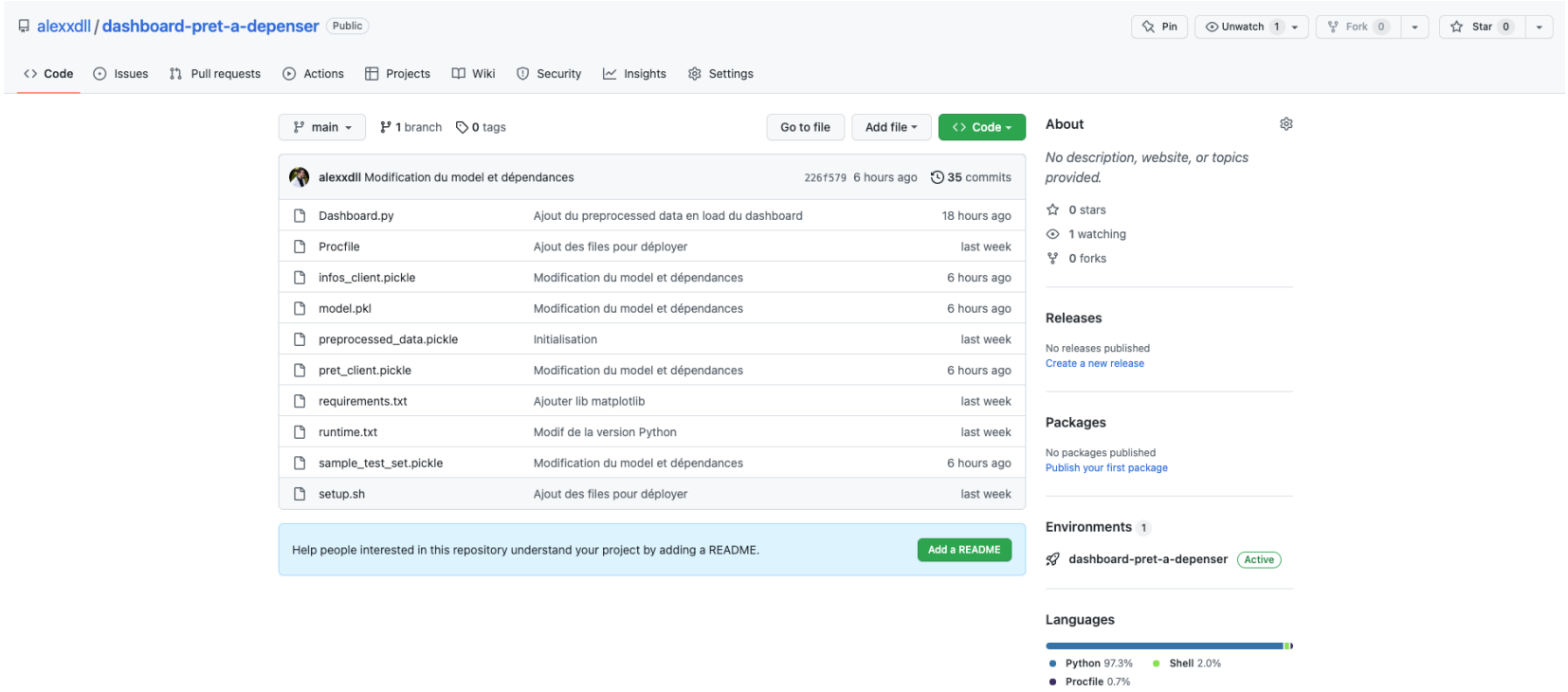
CODE SOURCE

```python
# Import des librairies
import pytest
import asyncio
import requests as re
import joblib
import numpy as np
from main import predict, shap_client

# Préparation du test
data = joblib.load('sample_test_set.pickle')
profile_ID = data.index.tolist()[0]
API_PRED = "https://api-creditscore.herokuapp.com/predict/"
API_SHAP = "https://api-creditscore.herokuapp.com/shap_client/"

# Recueil des return de l'API déployée
API_GET = API_PRED+(str(profile_ID))
score_client = re.get(API_GET).json()
API_GET = API_SHAP + (str(profile_ID))
shap_values = re.get(API_GET).json()

# Initialisation des tests (pour chaque fonction)
class Test:
    def test_return_predict(self):
        assert asyncio.run(predict(profile_ID)) == score_client

    def test_return_shap_client(self):
        assert asyncio.run(shap_client(profile_ID)) == shap_values
```

SUR TERMINAL

```
(base) alexandredelaguillaumie@MacBook-Pro-de-Alexandre-2 fastapi % pytest -q
..                                                                       [100%]
============================== warnings summary ==============================
test_unitaire.py: 502 warnings
  `np.int` is a deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you wish to review your current use, check the release note link for additional information.
  Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations

test_unitaire.py: 500 warnings
  NPY_ARRAY_UPDATEIFCOPY, NPY_ARRAY_INOUT_ARRAY, and NPY_ARRAY_INOUT_FARRAY are deprecated, use NPY_WRITEBACKIFCOPY, NPY_ARRAY_INOUT_ARRAY2, or NPY_ARRAY_INOUT_FARRAY2 respectively instead, and call PyArray_ResolveWritebackIfCopy before the array is deallocated, i.e. before the last call to Py_DECREF.

test_unitaire.py: 500 warnings
  UPDATEIFCOPY detected in array_dealloc.  Required call to PyArray_ResolveWritebackIfCopy or PyArray_DiscardWritebackIfCopy is missing

test_unitaire.py::Test::test_return_shap_client
  ntree_limit is deprecated, use `iteration_range` or model slicing instead.

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
2 passed, 1503 warnings in 53.25s
```

PREUVES DE DÉPLOIEMENT VIA GIT



alexxdll / dashboard-pret-a-depenser Public

Pin | Unwatch 1 | Fork 0 | Star 0

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

main | 1 branch | 0 tags | Go to file | Add file | Code

alexxdll Modification du model et dépendances 226f579 6 hours ago 35 commits

Dashboard.py	Ajout du preprocessed data en load du dashboard	18 hours ago
Procfile	Ajout des files pour déployer	last week
infos_client.pickle	Modification du model et dépendances	6 hours ago
model.pkl	Modification du model et dépendances	6 hours ago
preprocessed_data.pickle	Initialisation	last week
pret_client.pickle	Modification du model et dépendances	6 hours ago
requirements.txt	Ajouter lib matplotlib	last week
runtime.txt	Modif de la version Python	last week
sample_test_set.pickle	Modification du model et dépendances	6 hours ago
setup.sh	Ajout des files pour déployer	last week

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

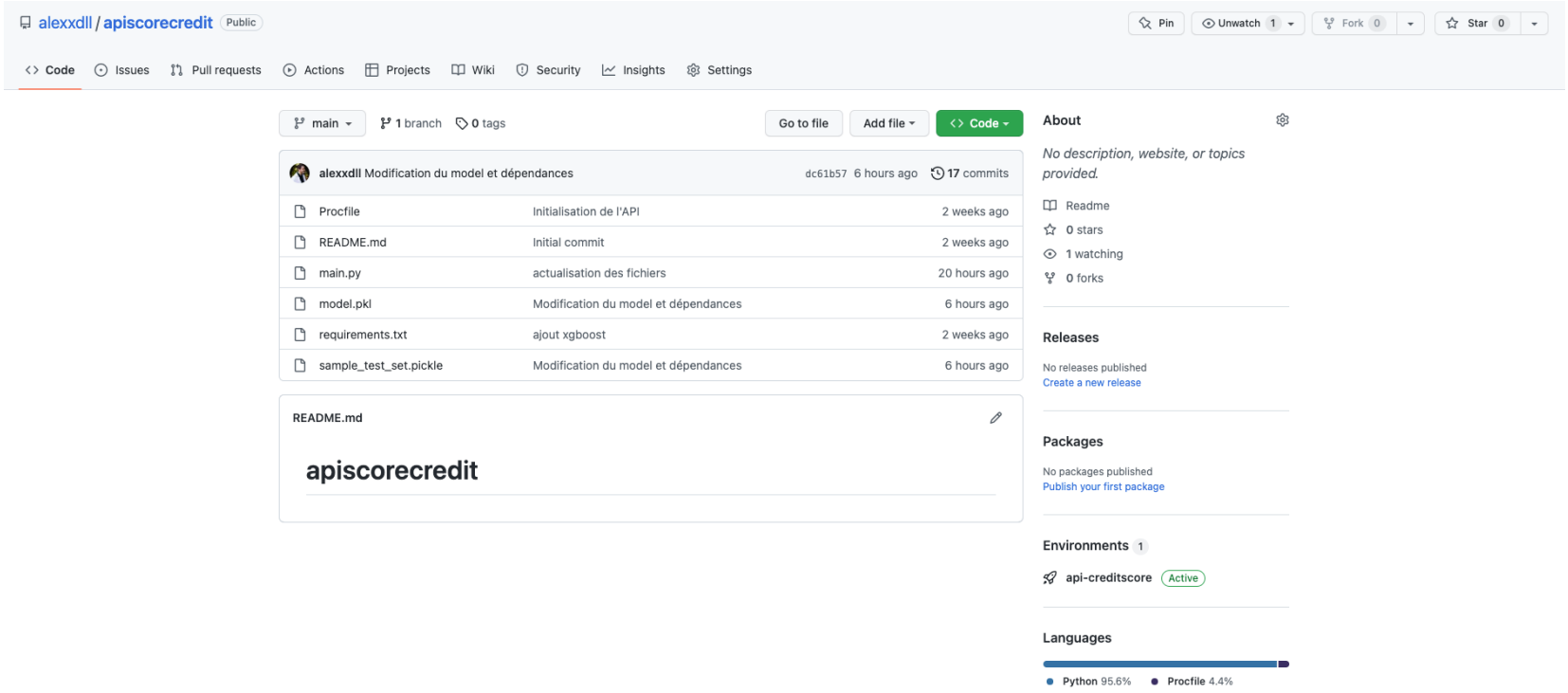
Environments 1

dashboard-pret-a-depenser Active

Languages

Python 97.3% Shell 2.0%
Procfile 0.7%

PREUVES DE DÉPLOIEMENT VIA GIT



LIMITES ET LES AMÉLIORATIONS POSSIBLES

+

- Il faudrait communiquer un brief plus précis afin d'évaluer précisément la loss function
- Réaliser séparément une black box pour mieux séparer les classes serait intéressant
- Éventuellement réaliser le projet avec Pyspark pourrait améliorer les temps de calcul

-

- La taille du dataset entraine des problème de stockage et de mémoire conséquents.
- L'optimisation des hyperparamètres est très longues et nécéssite une courte liste d'hyperparams
- Les poids de pénalités de la fonction coût sont fixés arbitrairement et empêche l'optimisation
- La demande d'interpretabilité empêche de pouvoir réduire la dimension et améliorer les performances

MERCI POUR VOTRE ÉCOUTE