IMPLÉMENTATION DU MODÈLE DE SCORING

MISE EN PRODUCTION DE L'API ET DU DASHBOARD INTERACTIF

SOMMAIRE

Prêt à dépenser

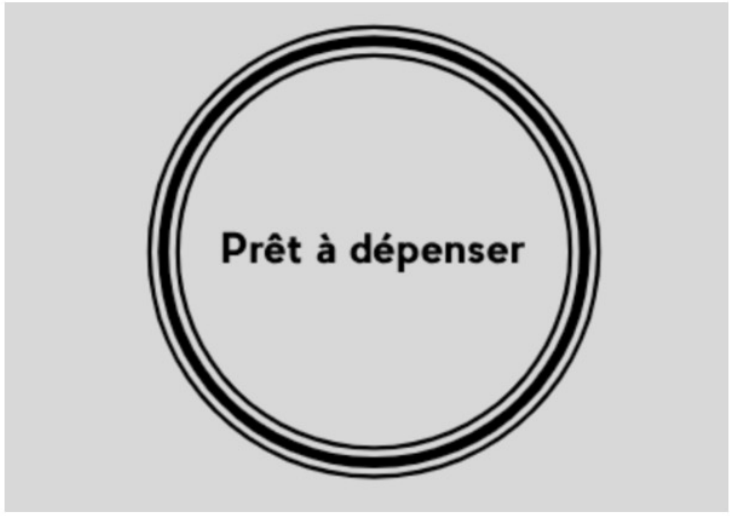
- Présentation
  - Projet
  - Jeu de données
- Modélisation
  - Métriques
  - Démarche
  - Tracking des modèles
  - Synthèse des résultats
- Pipeline de déploiement
  - Architecture
  - Tests unitaires
- DataDrift
- Dashboard

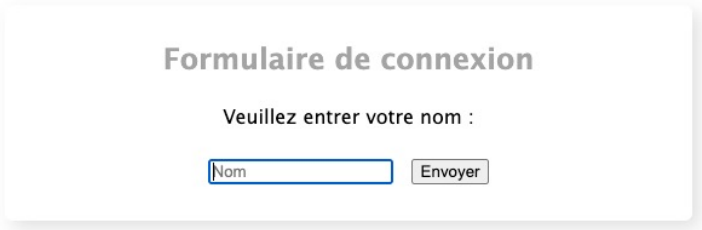
- Notes :
  - Données : https://www.kaggle.com/c/home-credit-default-risk/data
  - Kernel de départ adapté ensuite : https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script
  - GitHub : https://github.com/elisepoupi/P7_scoring
  - Dashboard : https://epoupi-pret-a-depenser.herokuapp.com/ noms à utiliser pour tests : personnages De Game of Thrones (ex: Robert, Ellaria…)

PRÉSENTATION PROJET

Prêt à dépenser

- Mettre en œuvre un outil de scoring crédit pour calculer la probabilité qu'un client rembourse son crédit et déterminer l'octroi ou non du crédit
  - **Maximisation des bénéfices de l'entreprise**
- Communiquer les résultats sur un dashboard à destination des chargés de relation client et du client.
  - **Rapidité de l'API**
  - **Clarté de l'analyse**
  - **Marge de négociation pour le chargé de relation client**
- S'appuyer sur des sources variées : données d'autres institutions financières, données comportementales…
  - **Utilisation d'un maximum de variables**





PRÉSENTATION
JEU DE DONNÉES

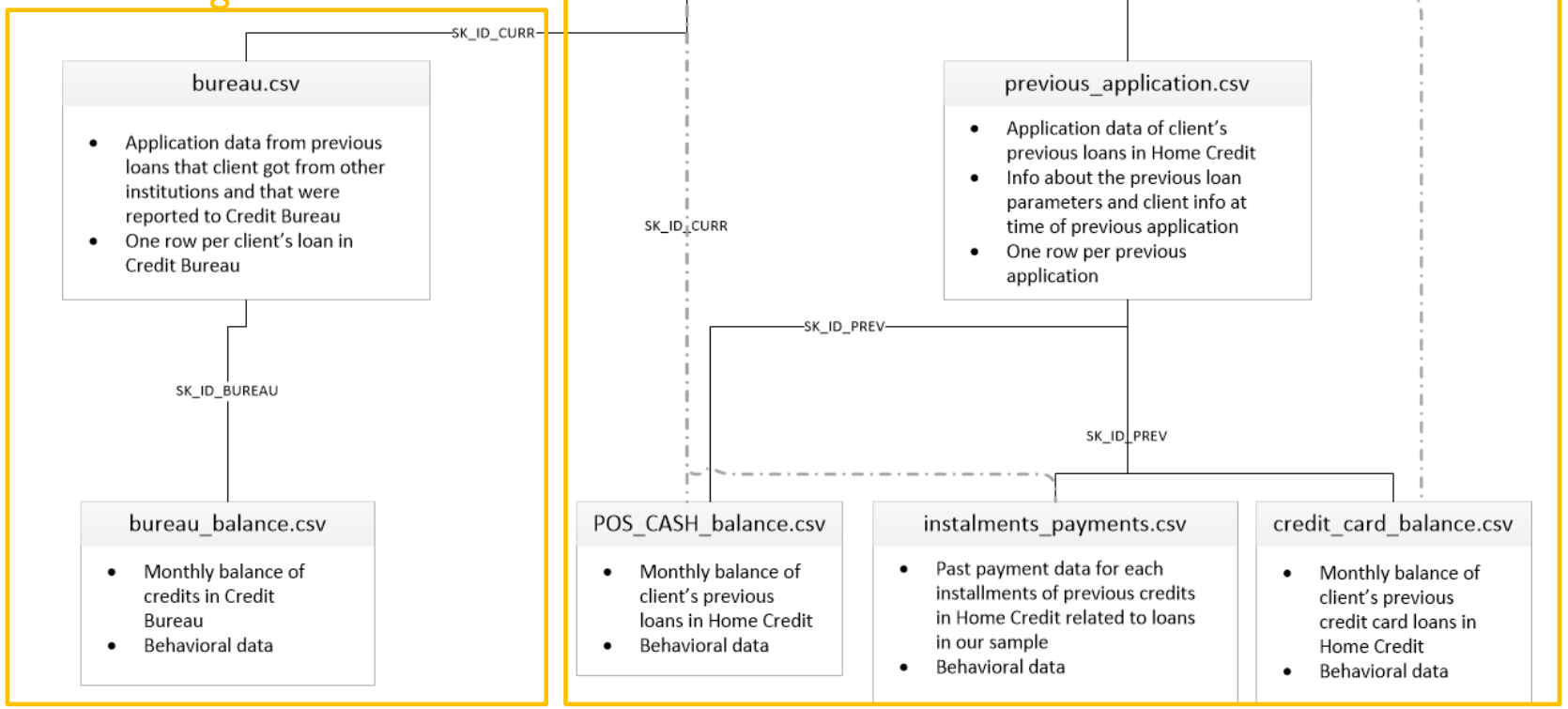
Prêt à dépenser

application_{train|test}.csv

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

120 variables
356 000 individus

Anciens prêts dans d'autres organismes bancaires



Anciens prêts chez prêt à dépenser

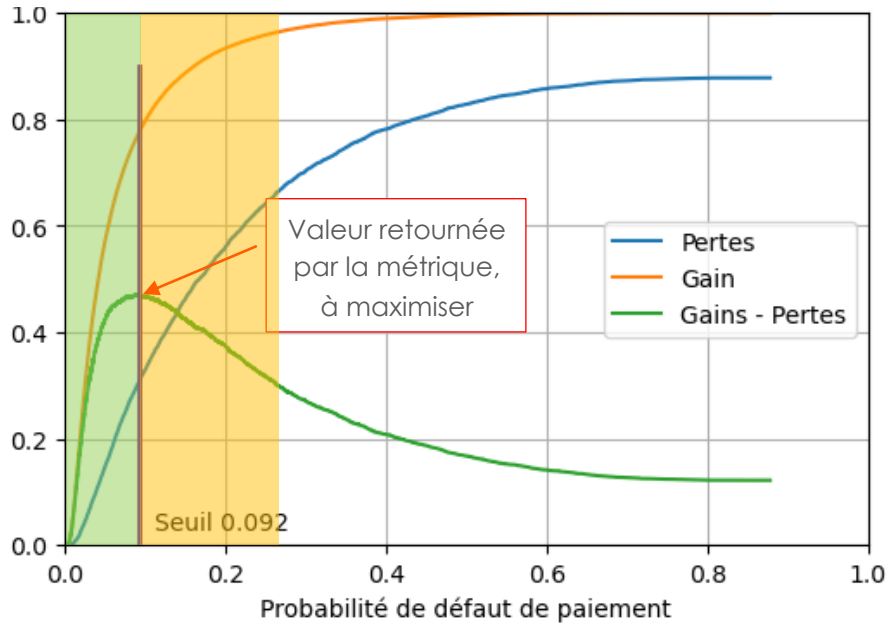
- Nombreuses valeurs manquantes
- 767 variables après regroupements

MODÉLISATION
MÉTRIQUES

Prêt à dépenser

- Métriques standard suivies :
  - ROC AUC et Accuracy
- Métrique métier : maximiser les bénéfices de l'entreprise
  - Hypothèses :
    - Bénéfice d'un dossier réalisé sans défaut de paiement : 1
    - Perte d'un dossier réalisé avec défaut de paiement : 10
  - Formule : bénéfice potentiel maximum / bénéfice total des dossiers sans défaut de paiement

- Métrique métier choisie :



Prêts accordés
Marge de négociation laissée au chargé de relation client
Avec gains potentiels > 30% des gains possibles

MODELISATION
DÉMARCHE 1/2

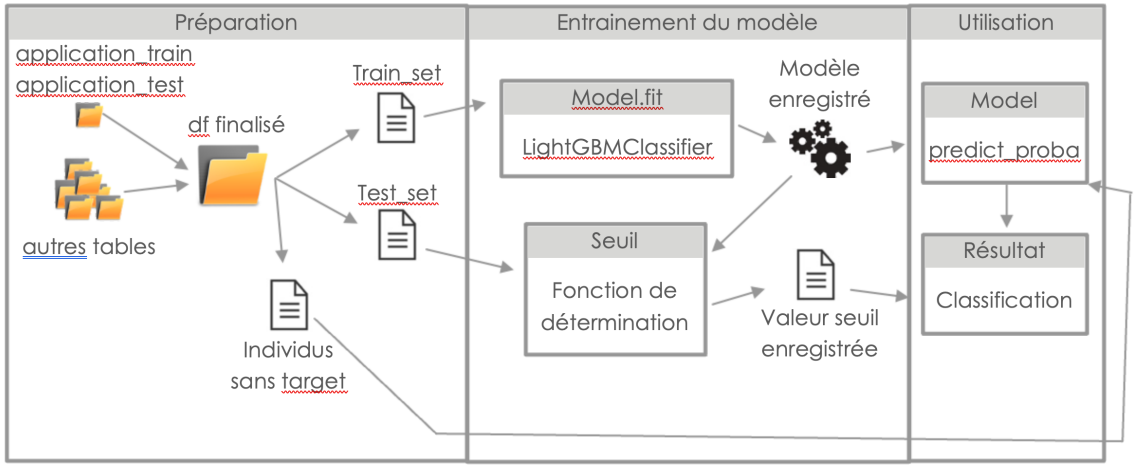
Prêt à dépenser

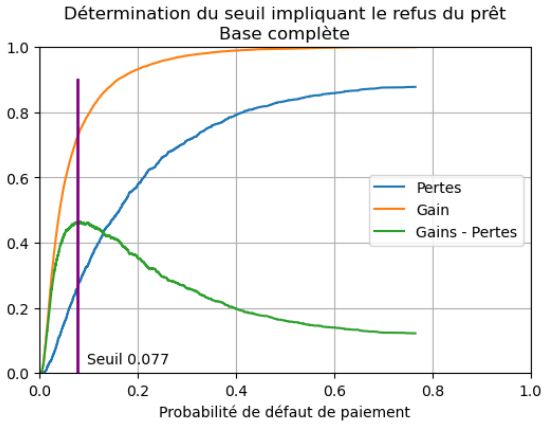
	Essai 1	Essai 2	Model 1	Essai 4	Model 2
Objectif	1er résultat rapide	Pouvoir utiliser Lime ou une régression linéaire ou de l'over sampling	Améliorer le score métier	Réduire le nombre de variables pour explications claires	Négociation : marge plus importante et possibilité de modifier des données
Préprocessing		KNNImputer (trop de variables) ACP (NaN) ANOVA KNNImputer 200	Imputation manuelle des valeurs manquantes	Imputation manuelle des valeurs manquantes Sélection ANOVA	Imputation manuelle des valeurs manquantes Pipeline intégrant la transformation des données Under sampling
Valeurs NaN	Non traitées	Aucune	Aucune	Aucune	Aucune
Réduc. dimens.	Aucune	200 features ANOVA	Aucune	200 feat. ANOVA	Aucune
Modèle	LightBGMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
Explication locale	Lime (NaN)	Lime imparfait (catégoriel : 1variable par label existant)	Lime imparfait	Lime imparfait	Lime plus clair (variable catégorielle = 1 variable)
Score	0.470	0.439	0.470	0.451	0.404

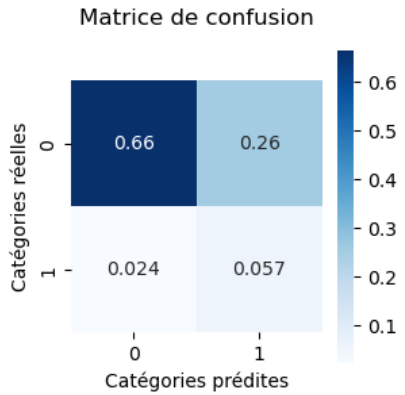
MODELISATION
DÉMARCHE 2/2

Prêt à dépenser

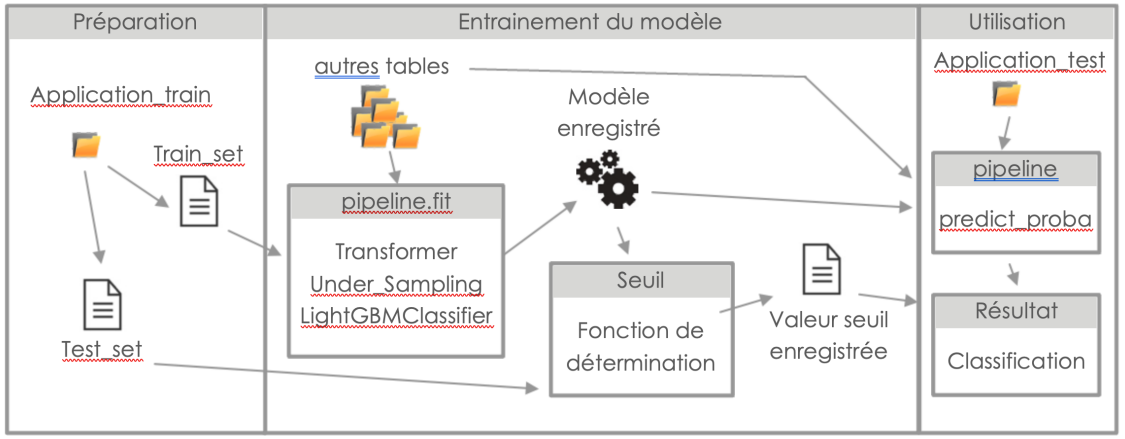
MODELE N° 1

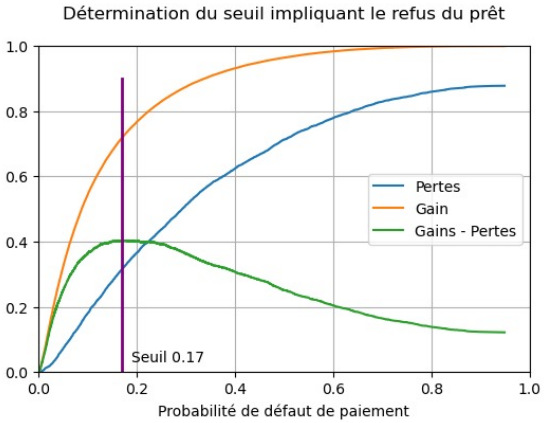


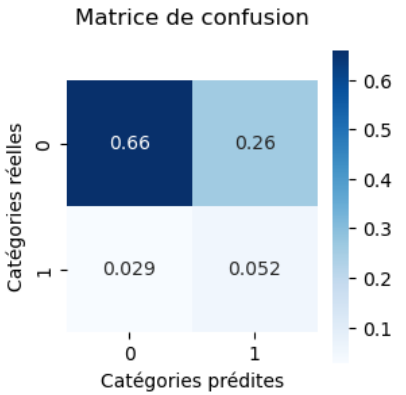




MODELE N° 2



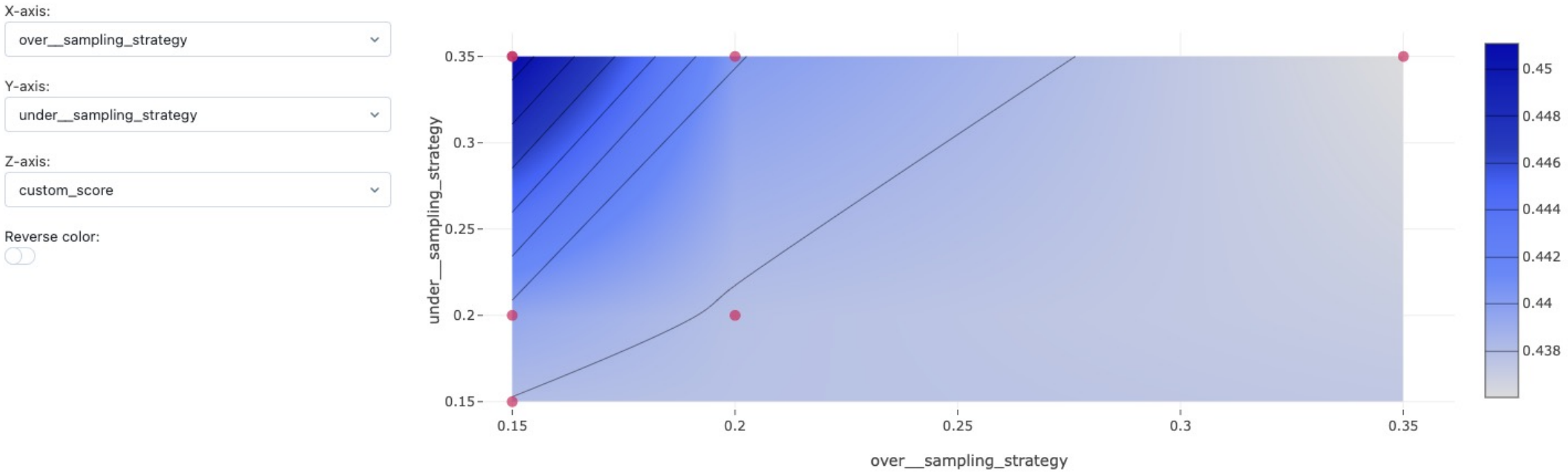




MODELISATION
TRACKING DES MODÈLES

Prêt à dépenser

Exemple de visualisation MLFlow



PIPELINE DE DÉPLOIEMENT
ARCHITECTURE

Prêt à dépenser

Projet

- P7_scoring_analyse.ipynb
- note_methodo
- P7_Présentation.pptx
- note_methodo.pdf
- .gitignore
- requirements.txt
- ~$P7_Présentation.pptx
- README.md
- .git
- .github
- .ipynb_checkpoints
- .pytest_cache
- api_scoring
- dashboard
- data
  - installments_payments.csv
  - data_drift.html
  - df_200varANOV...putationKNN.csv
  - credit_card_balance.csv
  - previous_application.csv
  - POS_CASH_balance.csv
  - bureau_balance.csv
  - bureau.csv
  - application_train.csv
  - data_drift_partiel.html
  - application_test.csv
  - data_stability.html
  - data_stability_partiel.html
  - Features_ANOVA.csv
  - HomeCredit_colu..._description.csv
- static

Git Hub

elisepoupi test après maj du fichier yml — d106f6b 26 minutes ago — **40** commits

.github/workflows	Update python-app.yml	3 hours ago
api_scoring	ajout test dashboard	4 hours ago
dashboard	ajout test dashboard	4 hours ago
data	maj du 01-03-23	3 weeks ago
static	gross maj du 16-03-23	4 days ago
.gitignore	gross maj du 16-03-23	4 days ago
README.md	Initial commit	last month
note_methodo.docx	maj des modèles	5 hours ago
note_methodo.pdf	maj des modèles	5 hours ago
requirements.txt	modification des requirements où trop d'éléments apparaissaient	2 weeks ago
scoring_analyse-Copy1.ipynb	test après maj du fichier yml	26 minutes ago
scoring_analyse.ipynb	maj du 9-3-23	2 weeks ago

api_scoring

- .git Vers HEROKU 1
- .gitignore
- .pytest_cache
- documentation
- Procfile
- requirements.txt
- runtime.txt
- scoring.py
- static
  - application.csv
  - base_client.csv
  - bureau_balance.csv
  - bureau.csv
  - col_to_delete.csv
  - credit_card_balance.csv
  - imputation.csv
  - imputation2.csv
  - installments_payments.csv
  - list_OHE.json
  - mlflow_model
  - mlflow_pipeline
  - pos_cash_balance.csv
  - previous_application.csv
  - type_columns.json

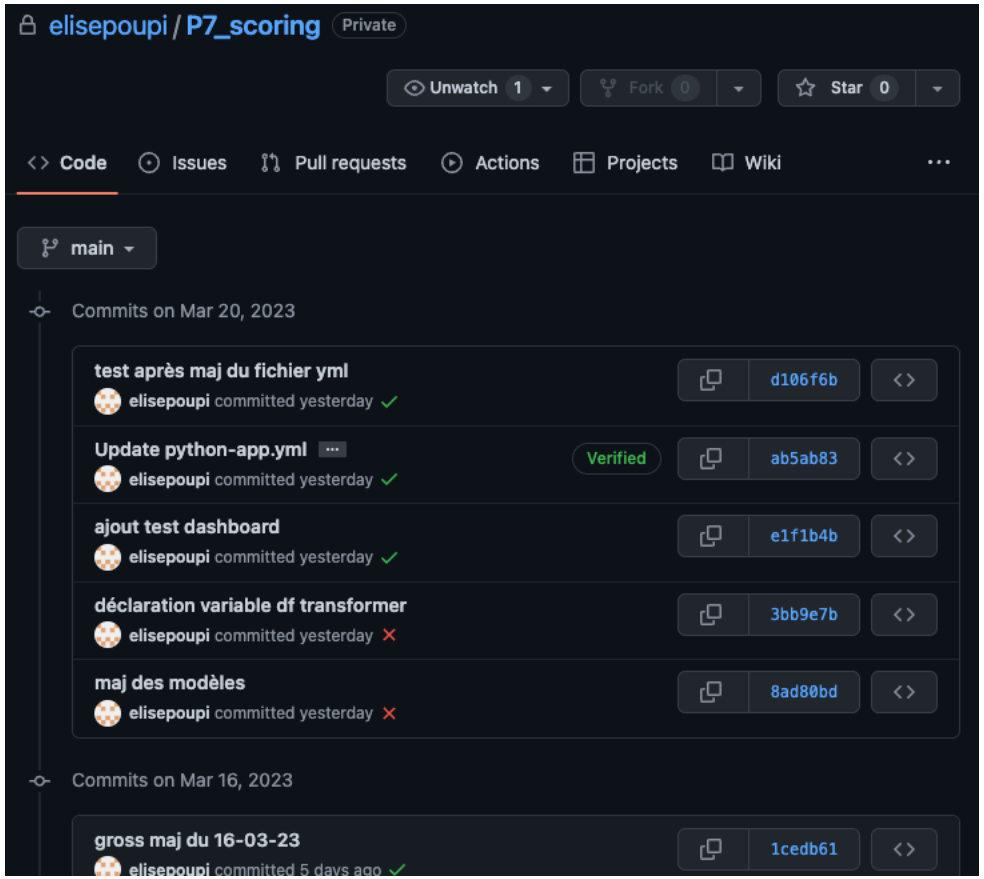
dashboard

- __pycache__
- .git Vers HEROKU 2
- .gitignore
- .ipynb_checkpoints
- config.py
- main.py
- Procfile
- requirements.txt
- runtime.txt
- static
  - application.csv
  - base_client.csv
  - data_boxplot.csv
  - feature_importance.csv
  - graph_metric_model.png
  - graph_metric_pipeline.png
  - image_refus.png
  - image_valide.png
  - list_OHE.json
  - logo.png
  - seuil_pipeline.txt
  - seuil.txt
  - style.css
  - type_columns.json
- templates
  - connexion.html
  - donnees.html
  - score.html

PIPELINE DE DÉPLOIEMENT
GIT

Prêt à dépenser

Visualisation des commits sur GitHub



Commandes git

Pull

Status

Add

Commit

Push

```
elisepoupinet@MacBook-Air-de-Elise Projet % git pull
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 827 bytes | 118.00 KiB/s, done.
From https://github.com/elisepoupi/P7_scoring
   e1f1b4b..ab5ab83  main       -> origin/main
Updating e1f1b4b..ab5ab83
Fast-forward
 .github/workflows/python-app.yml | 1 +
 1 file changed, 1 insertion(+)
elisepoupinet@MacBook-Air-de-Elise Projet % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   scoring_analyse-Copy1.ipynb

no changes added to commit (use "git add" and/or "git commit -a")
elisepoupinet@MacBook-Air-de-Elise Projet % git add .
elisepoupinet@MacBook-Air-de-Elise Projet % git commit -m "test après maj du fichier yml"
[main d106f6b] test après maj du fichier yml
 1 file changed, 43 insertions(+), 5 deletions(-)
elisepoupinet@MacBook-Air-de-Elise Projet % git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 39.98 KiB | 602.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/elisepoupi/P7_scoring.git
   ab5ab83..d106f6b  main -> main
branch 'main' set up to track 'origin/main'.
elisepoupinet@MacBook-Air-de-Elise Projet %
```

PIPELINE DE DÉPLOIEMENT
PYTEST

Prêt à dépenser

En local

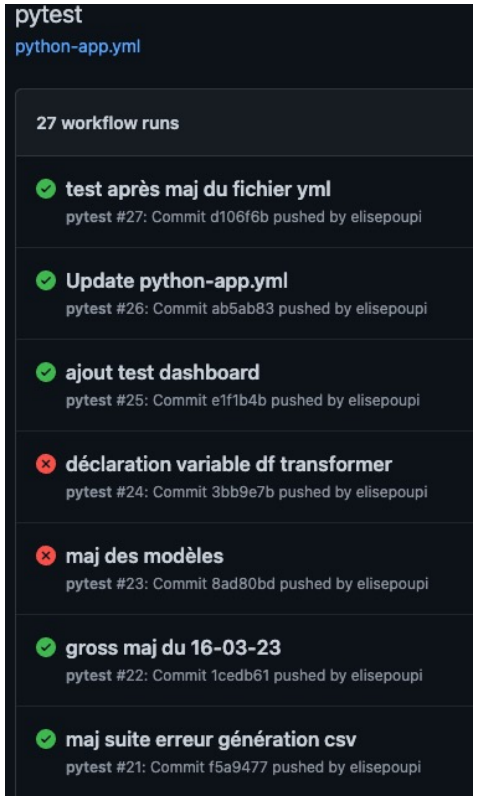
```
(LightGBM) elisepoupinet@MacBook-Air-de-Elise api_scoring % pytest scoring.py
============================= test session starts ==============================
platform darwin -- Python 3.9.13, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/elisepoupinet/Documents/Data_Scientist/P7 scoring/Projet/api_scoring
plugins: anyio-3.5.0
collected 1 item

scoring.py .                                                             [100%]

============================== 1 passed in 2.70s ===============================
(LightGBM) elisepoupinet@MacBook-Air-de-Elise api_scoring %
```

- Tests réalisés :
  - Api : test de la fonction de prédiction du score à partir d'un individu du df complet
  - Dashboard : test de la fonction renvoyant la variable « SK_ID_CURR » à partir du nom du client

Automatisé sur push Git

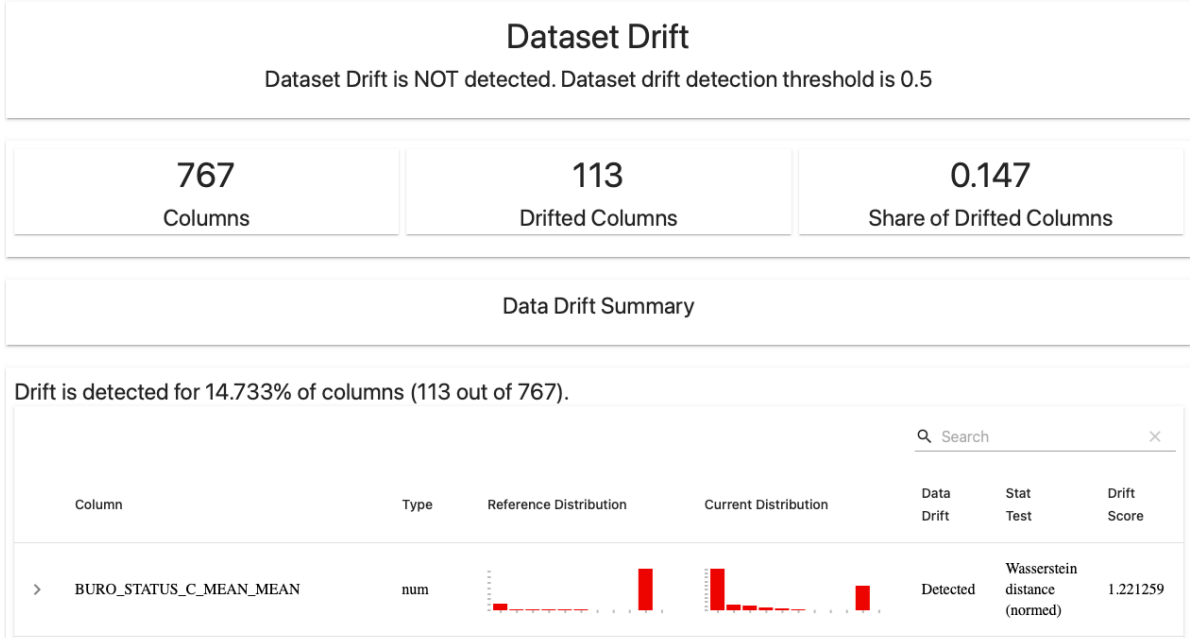


Résultat test automatisé sur GitHub

DATA DRIFT

Prêt à dépenser

- Data Drift recherché sur le df final :
  - Toutes les tables apparaissent 🙂
  - Les variables catégorielles ont été encodées 😕
  - Nombreuses variables _sum, _mean… diluent les résultats 😕



- Data Drift détecté si Drift_score > 0.10 sur 50% des variables. Ici drift sur 0.147% des variables. Parmi elles :
  - Data Drift sur les tables Bureau et Bureau_balance (provenance d'autres organismes bancaires) > 0.6
  - Data Drift sur Payment_Rate (var. importante) : 0.58
  - Data Drift sur EXT_SOURCE_1 : 0.16

- Dans l'idéal il faudrait refaire une analyse avec un df finalisé où les tables sont regroupées, sans encodage des variables catégorielles, et sans création de toutes les variables _mean, _sum, _var… pour vérifier la validité des résultats.

DASHBOARD

Prêt à dépenser

- Connexion

- Visualisation des données

- Visualisation des résultats

MERCI POUR VOTRE ATTENTION

PLACE AUX QUESTIONS