

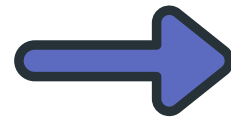
# Prêt **à** dépendre



Implémenter un outil de scoring

# Objectifs du projet

Prêt à  
dépenser

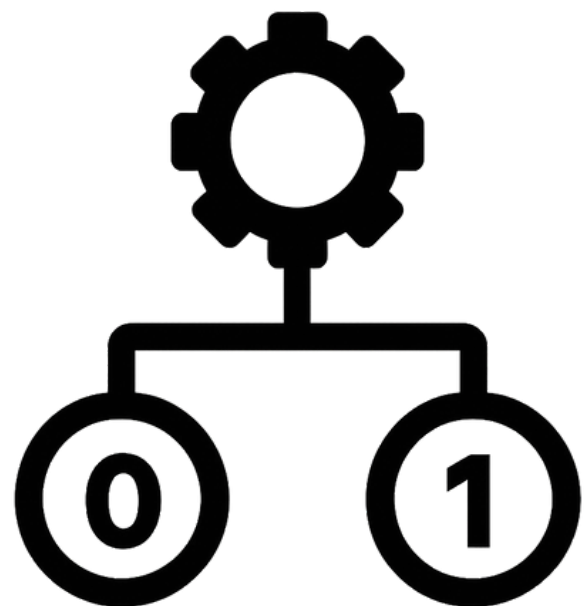


Gestion du  
risque



Outil de  
scoring

Implementer un outil de  
scoring



Suivre l'optimisation via  
MLFlow



Deployer le modèle dans  
le cloud

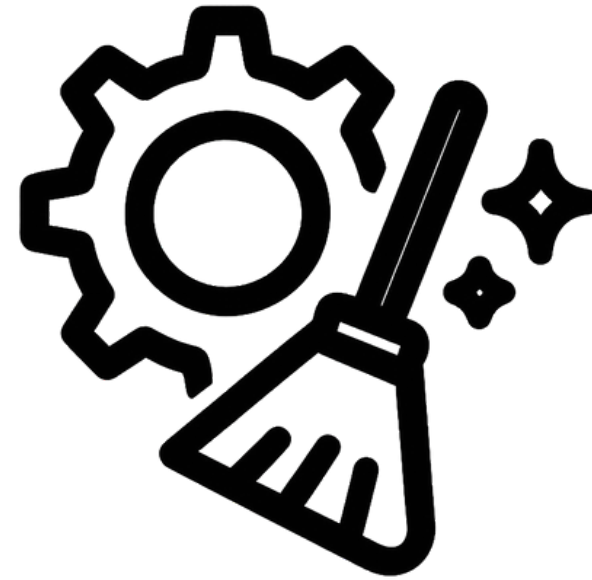




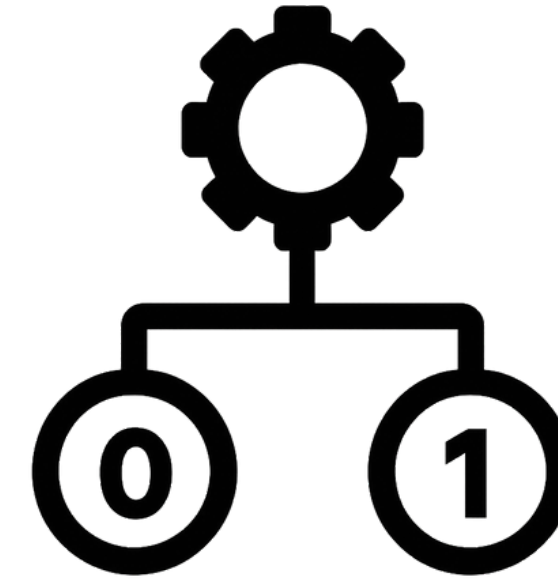
Présentation  
des données



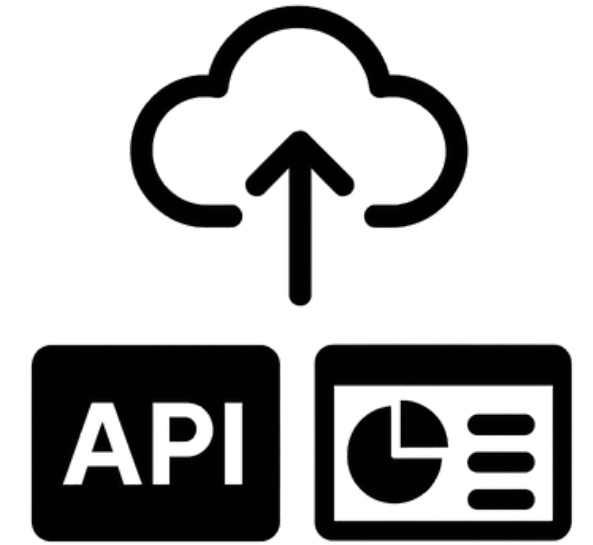
Présentation  
de l'approche



EDA + feature  
engineering



Modélisation



API +  
Dashboard

Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

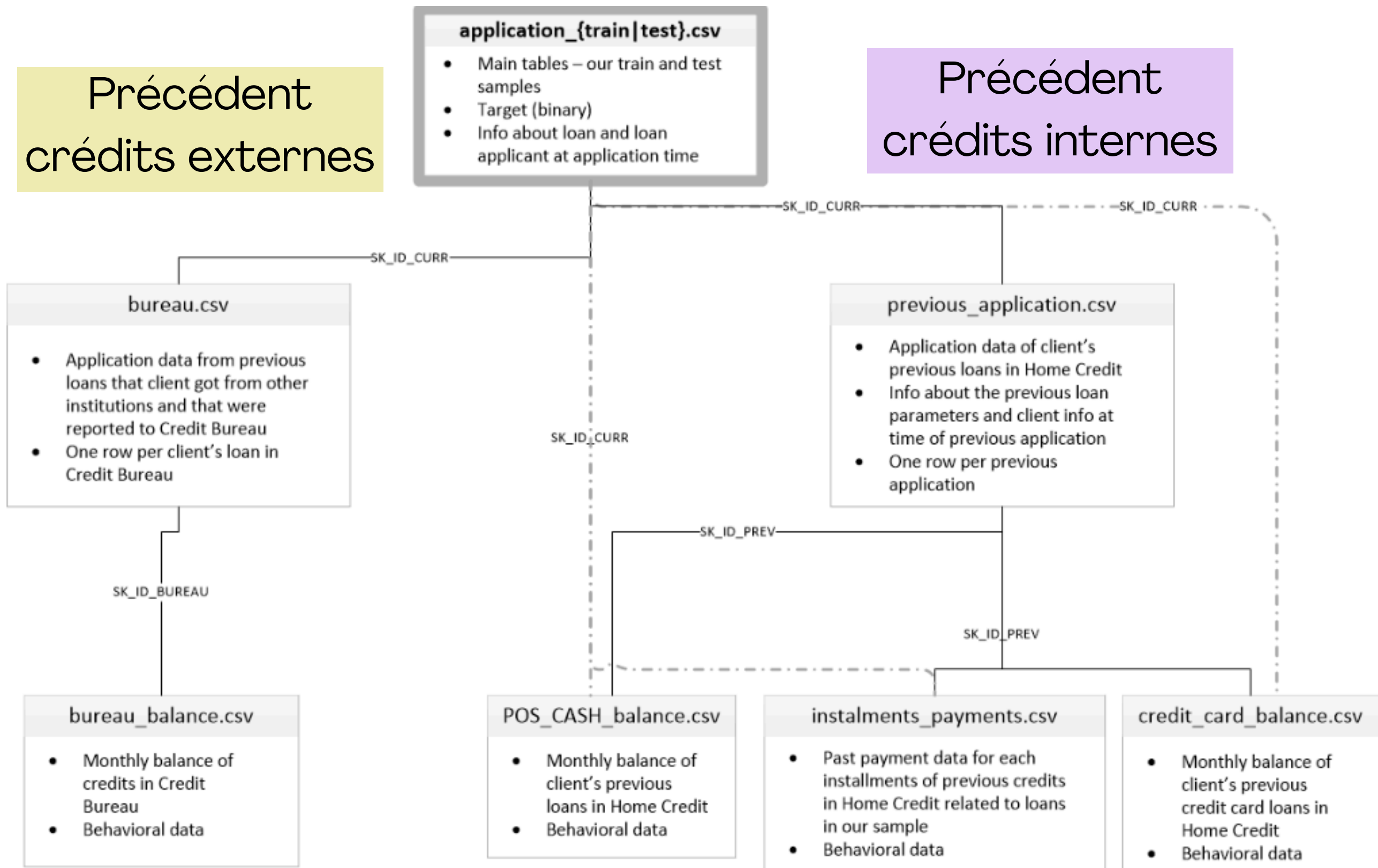
Modélisation

API +  
Dashboard

Précédent  
crédits externes

Table des  
demandes

Précédent  
crédits internes



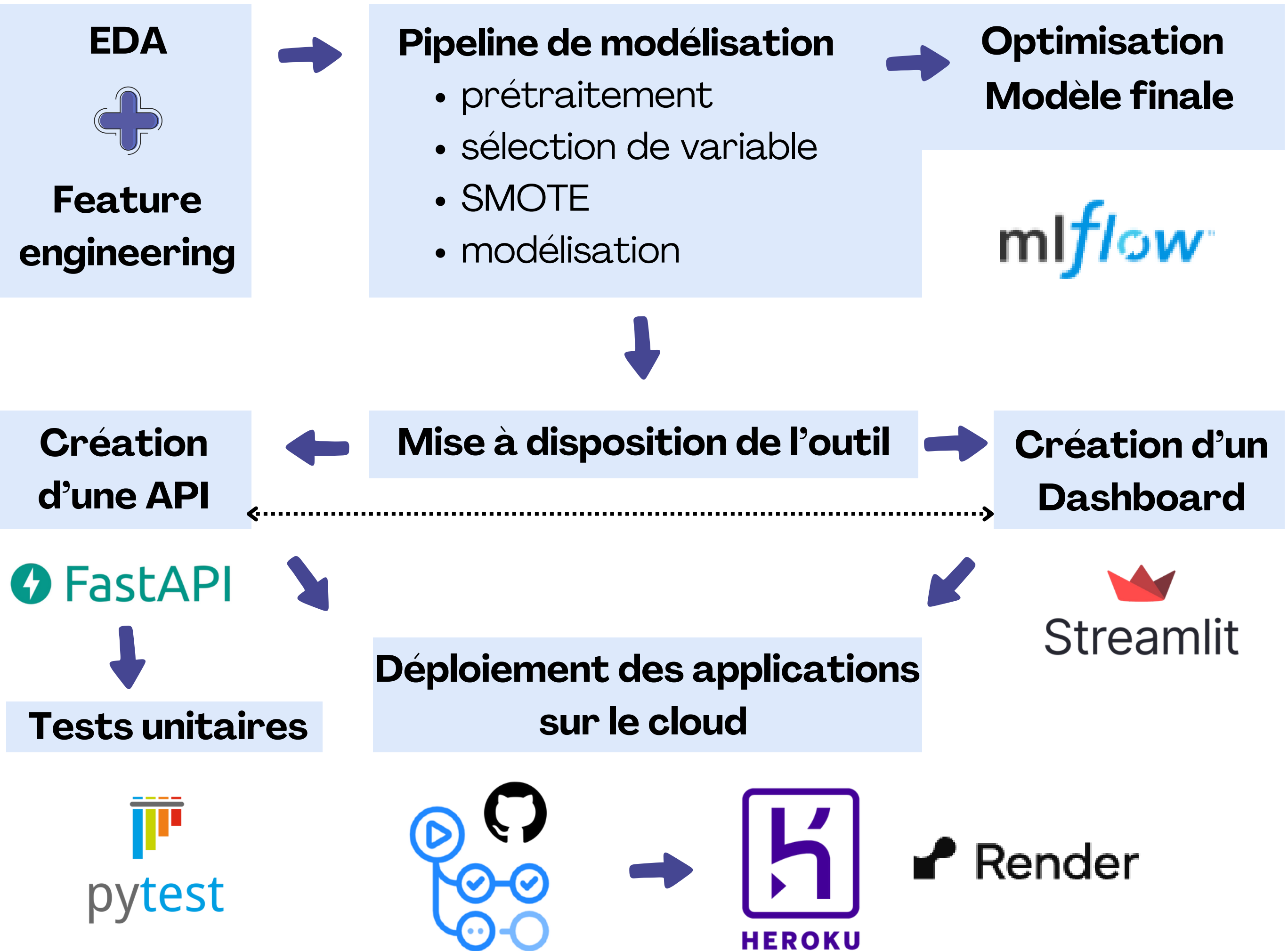
Présentation  
des données

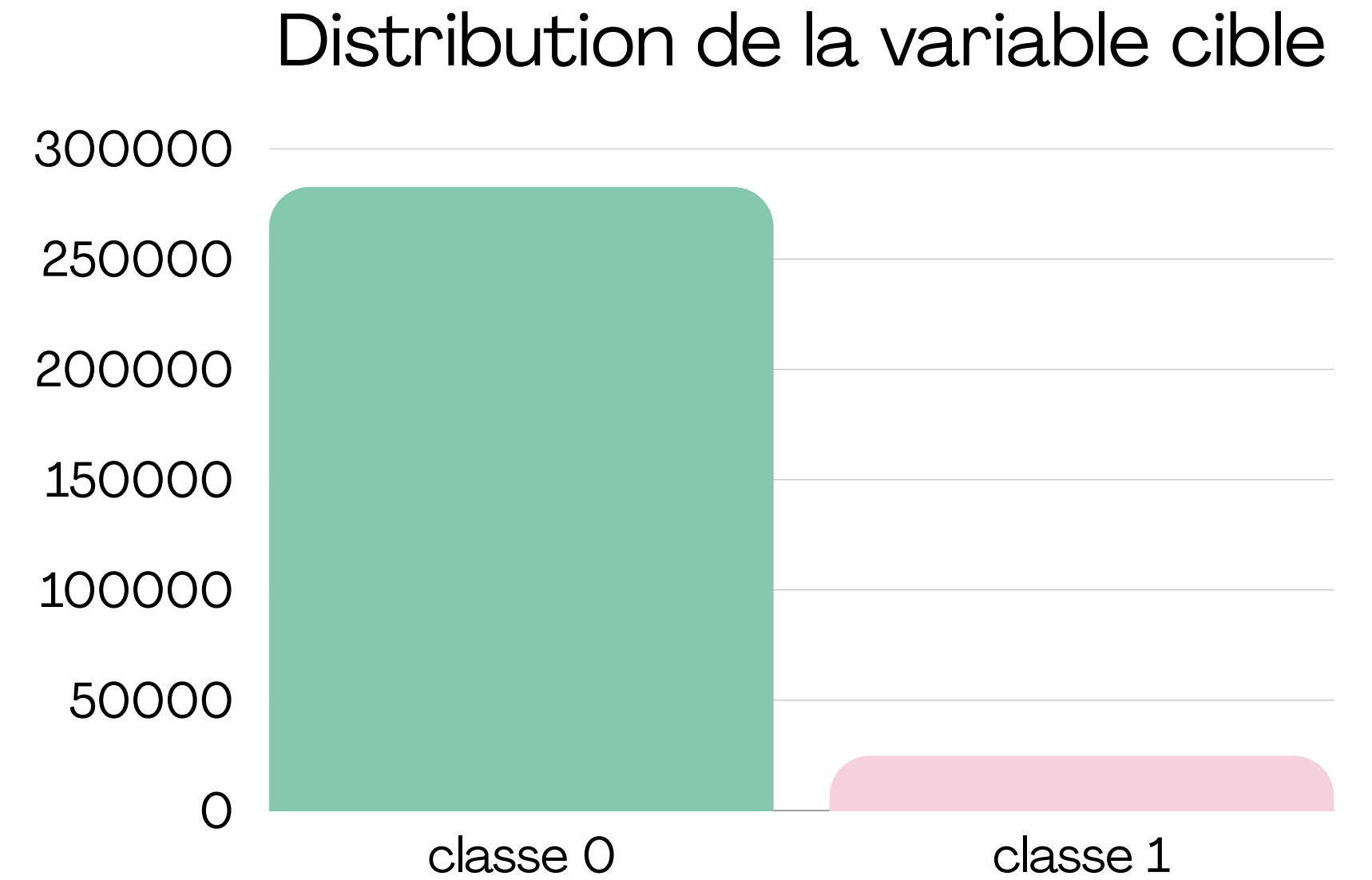
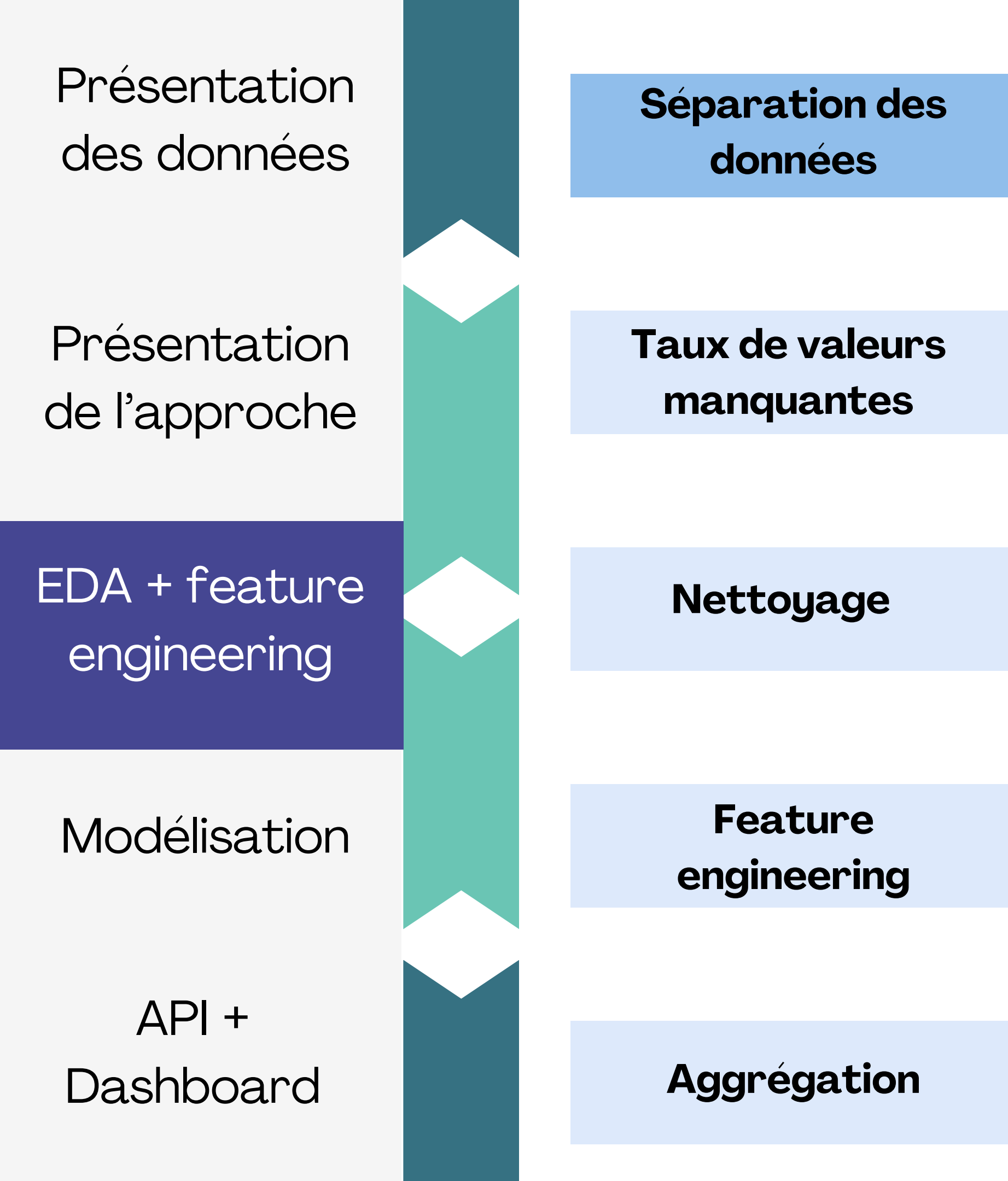
Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



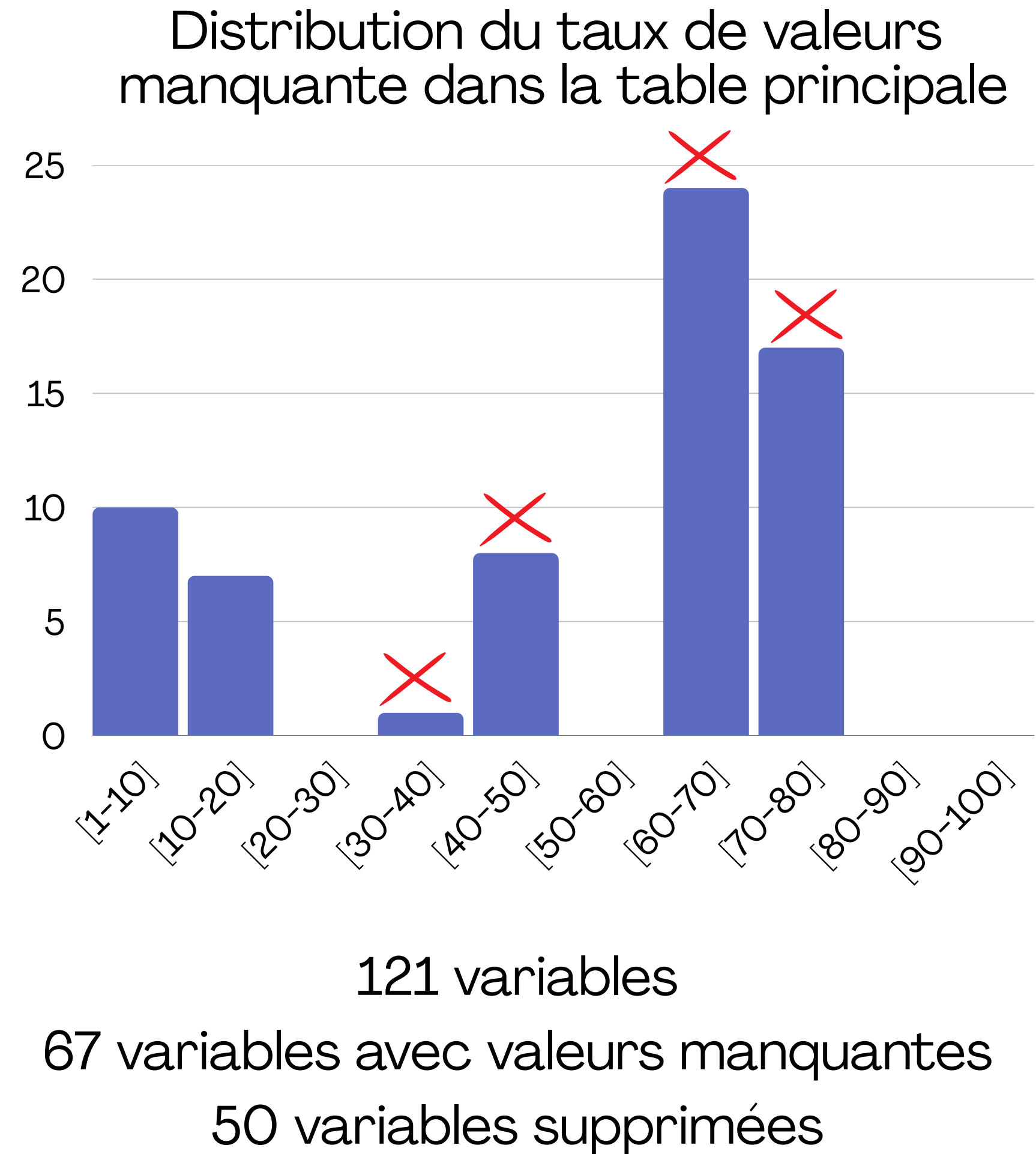
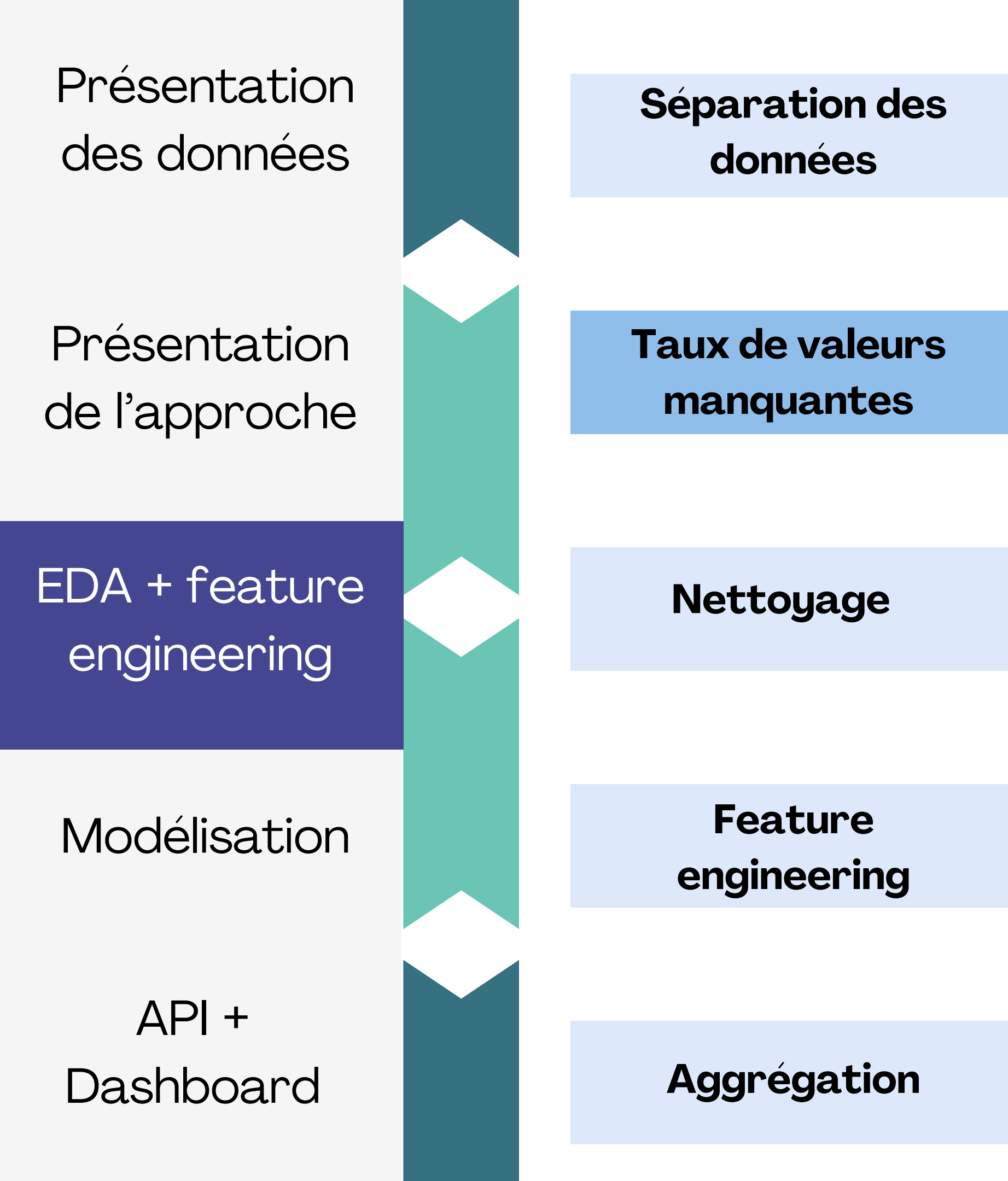


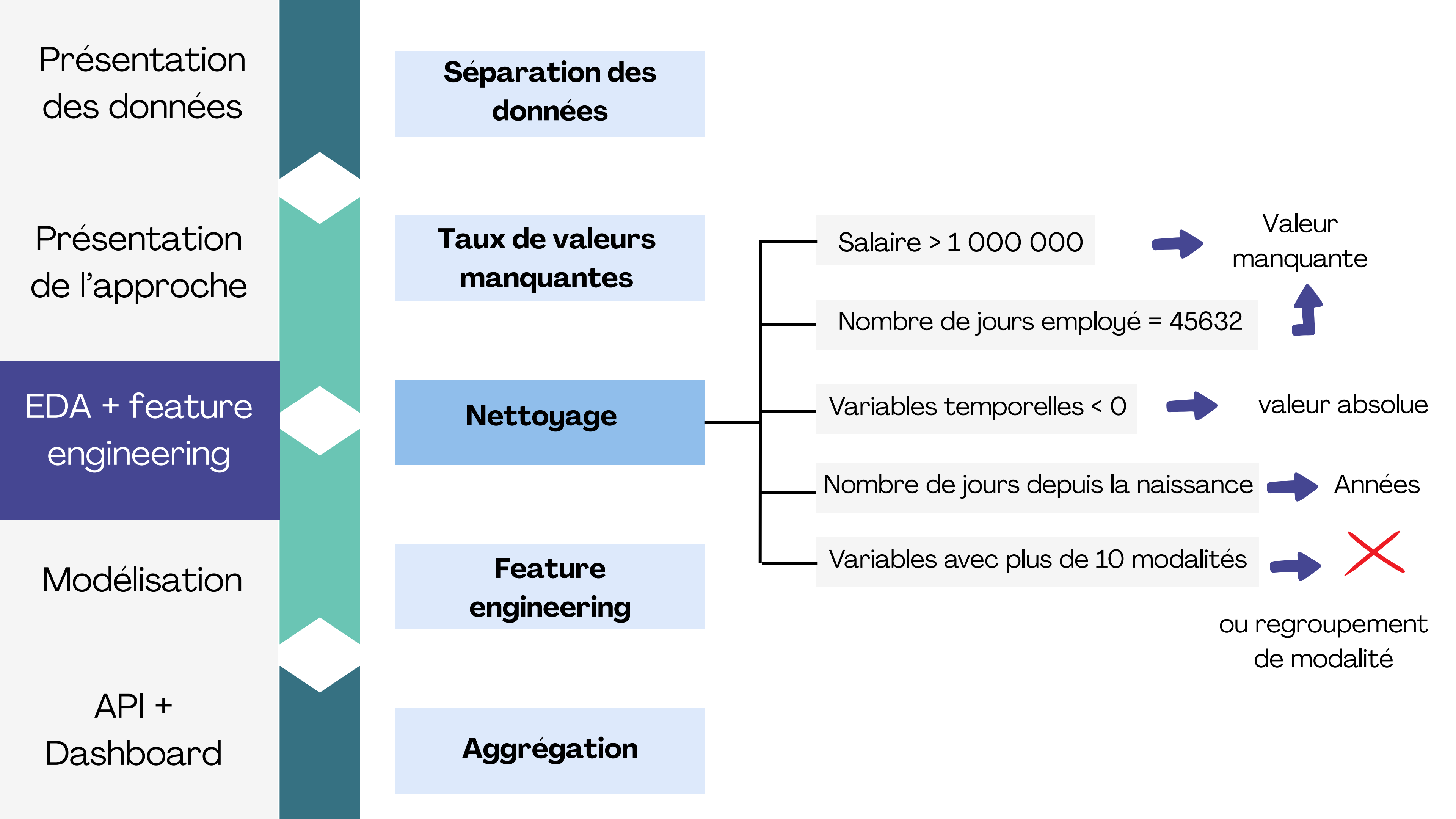
↓

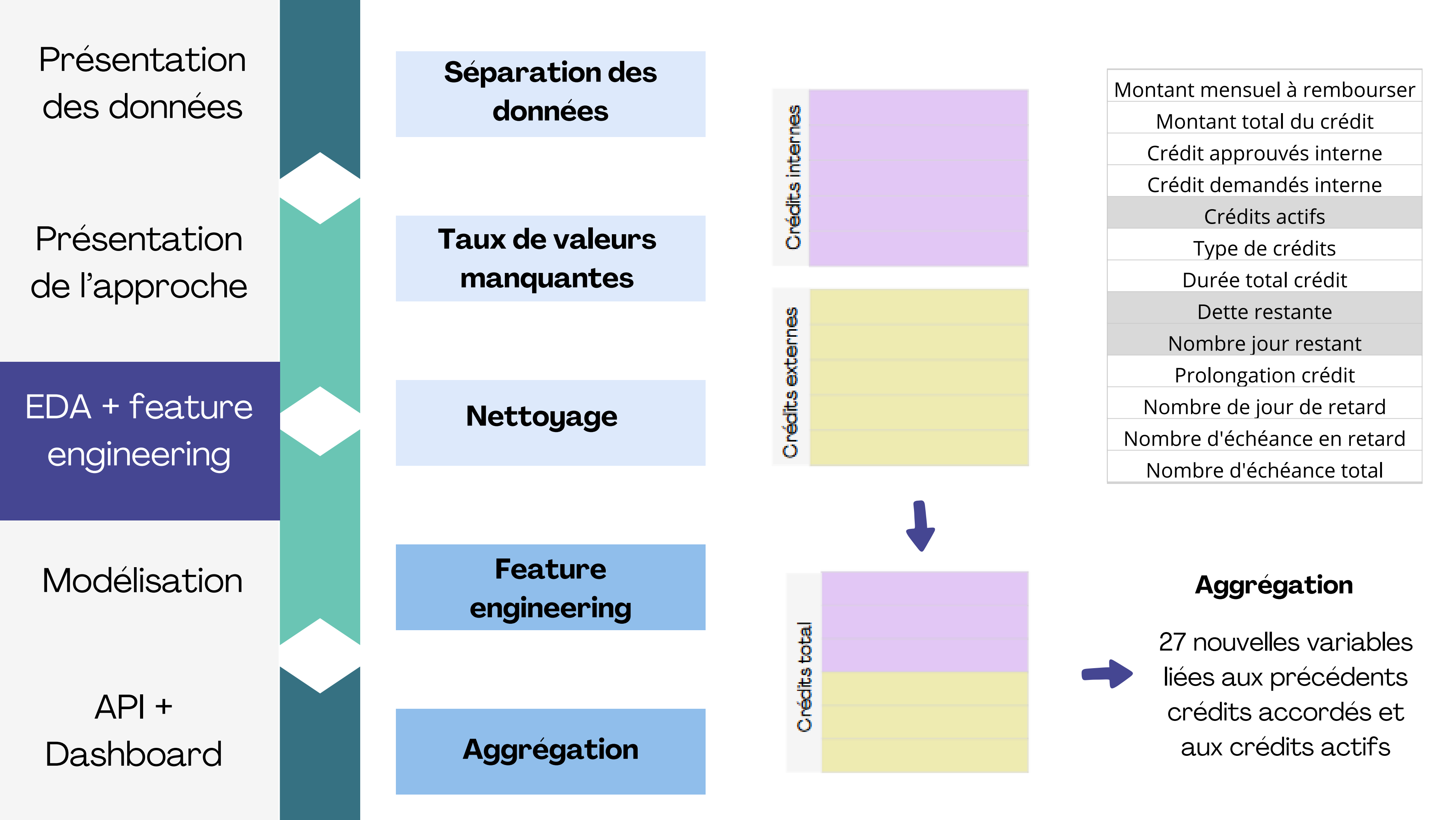
Jeux de données fortement déséquilibré

↓

Stratification







Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard

Prétraitement



2

Stratégie pour gérer  
la classe minoritaire

3

Exploration des  
modèles

4

Optimisation

5

Explicabilité du  
modèle retenu

6

MLflow

Imputation NA

Normalisation

Encodage

Numérique  
Crédits actifs

Imputation  
médiane ou -1

Standard  
Scaler

Numérique  
Crédits accordés

Imputation  
médiane ou -1

Standard  
Scaler

Numérique  
Autre

Imputation  
médiane

Standard  
Scaler

Catégorielle  
cat = 2

Ordinal  
Encodeur

Catégorielle  
cat > 2

OneHot  
Encodeur

Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

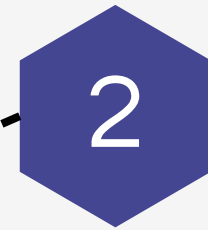
Modélisation

API +  
Dashboard

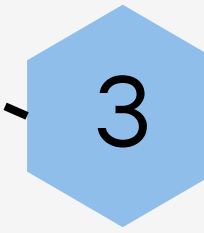
Prétraitement



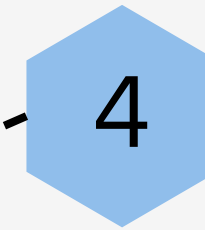
Stratégie pour gérer  
la classe minoritaire



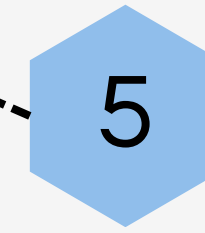
Exploration des  
modèles



Optimisation



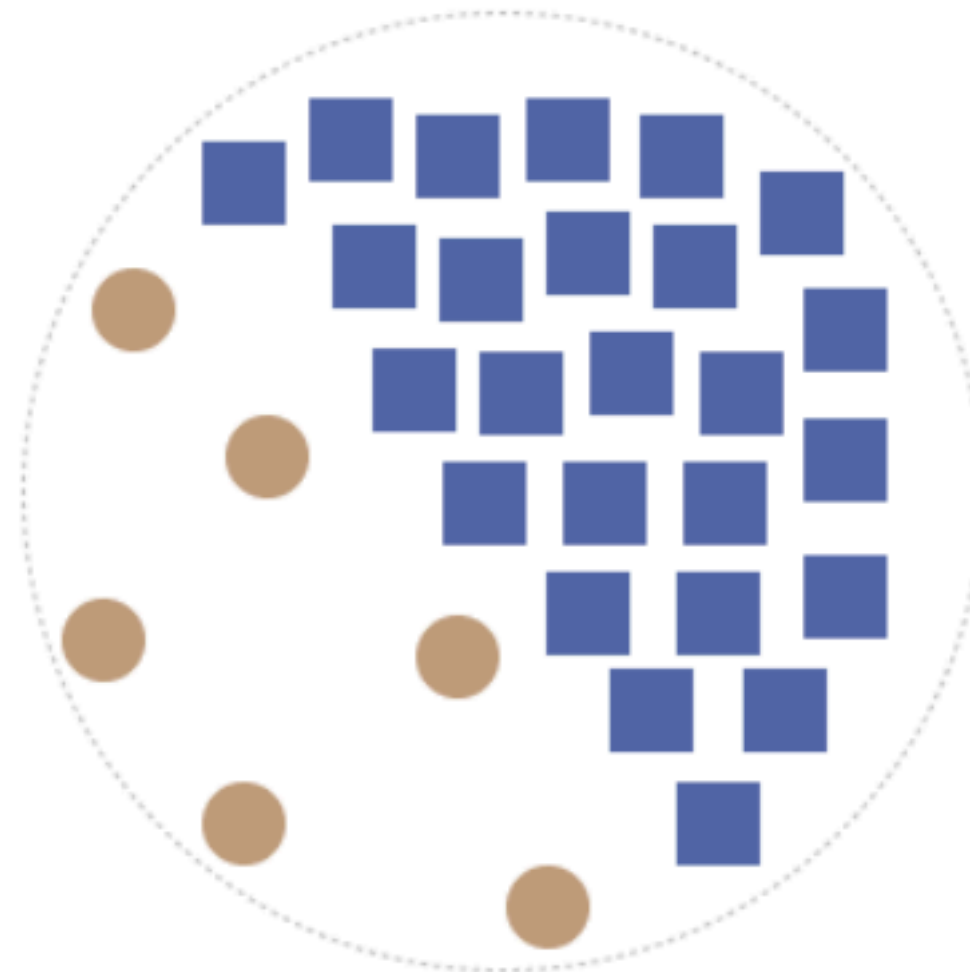
Explicabilité du  
modèle retenu



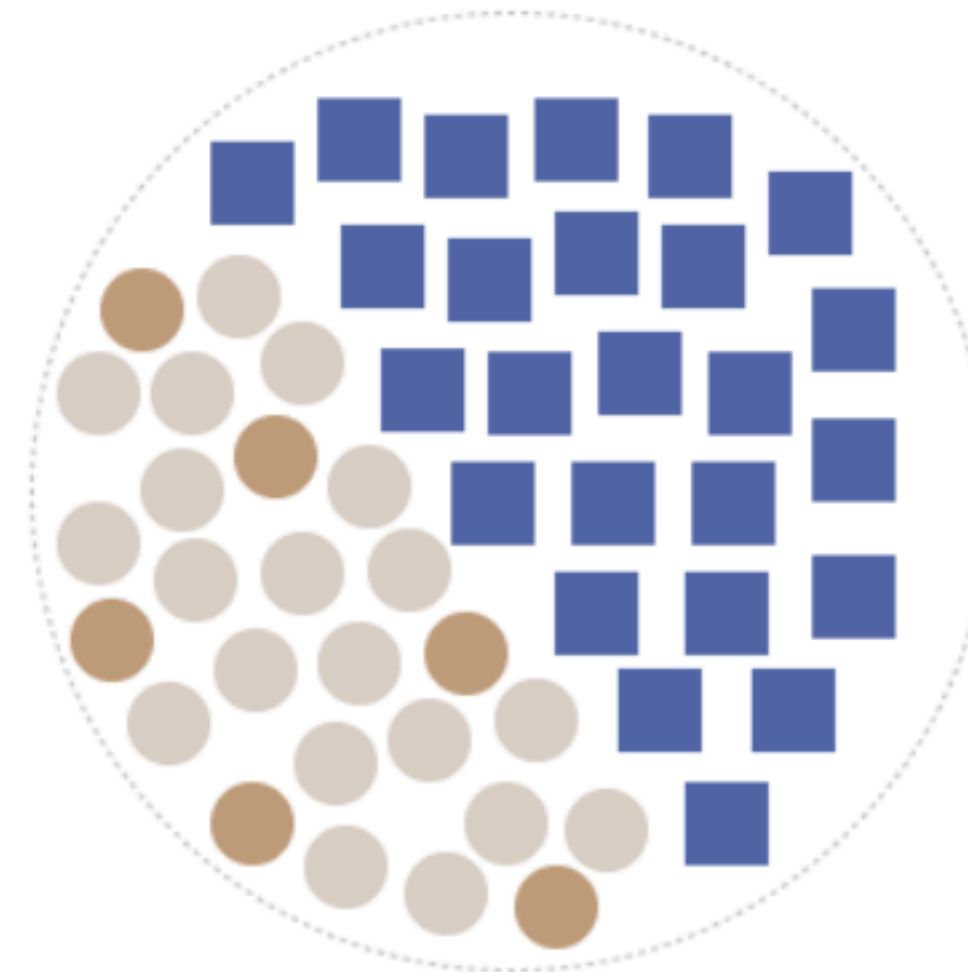
MLflow



**SMOTE**



Original Data



Resampled Data

■ Majority Class  
● Minority Class  
● SMOTE Samples

Ryan Roepke

<https://dataknowsall.com/blog/imbalanced>

**Random undersampling**

**Random oversampling**

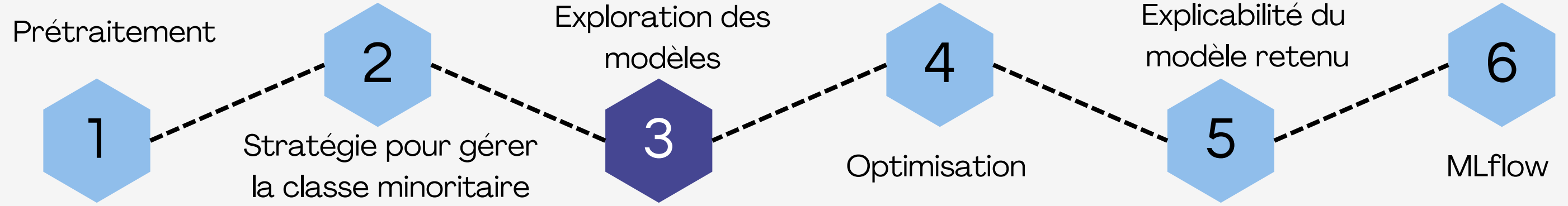
Présentation  
des données

Présentation  
de l'approche

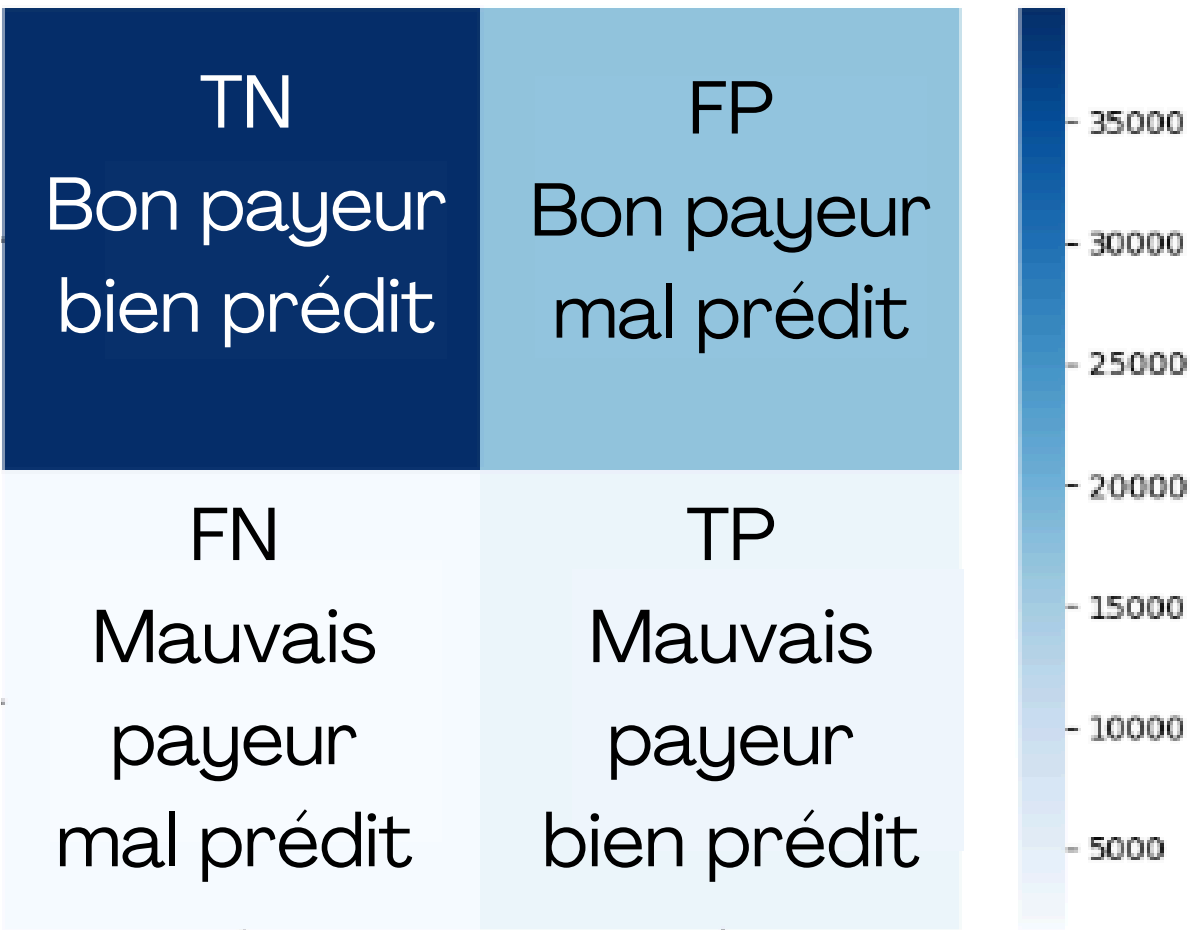
EDA + feature  
engineering

Modélisation

API +  
Dashboard



Pipeline	Modèles testés	Evaluation	
Prétraitement	LogisticRegression	Matrice confusion	Validation croisée sur fold stratifiés
SMOTE	Random Forest	Score de rappel	
Modèle	LigthGBM	Score F2	



Score métier

Prêt: 10 000 - intérêt: 1 000

TN: + 1 000

FP: - 1 000

FN : -1 000

TP: + 10 000

$$\text{Score} = \frac{10 * TP + TN - 10 FN - FP}{TP + TN + FN + FP}$$

Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard

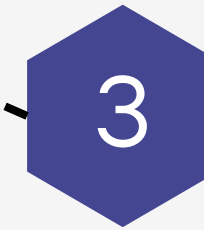
Prétraitement



2

Stratégie pour gérer  
la classe minoritaire

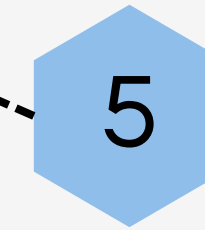
Exploration des  
modèles



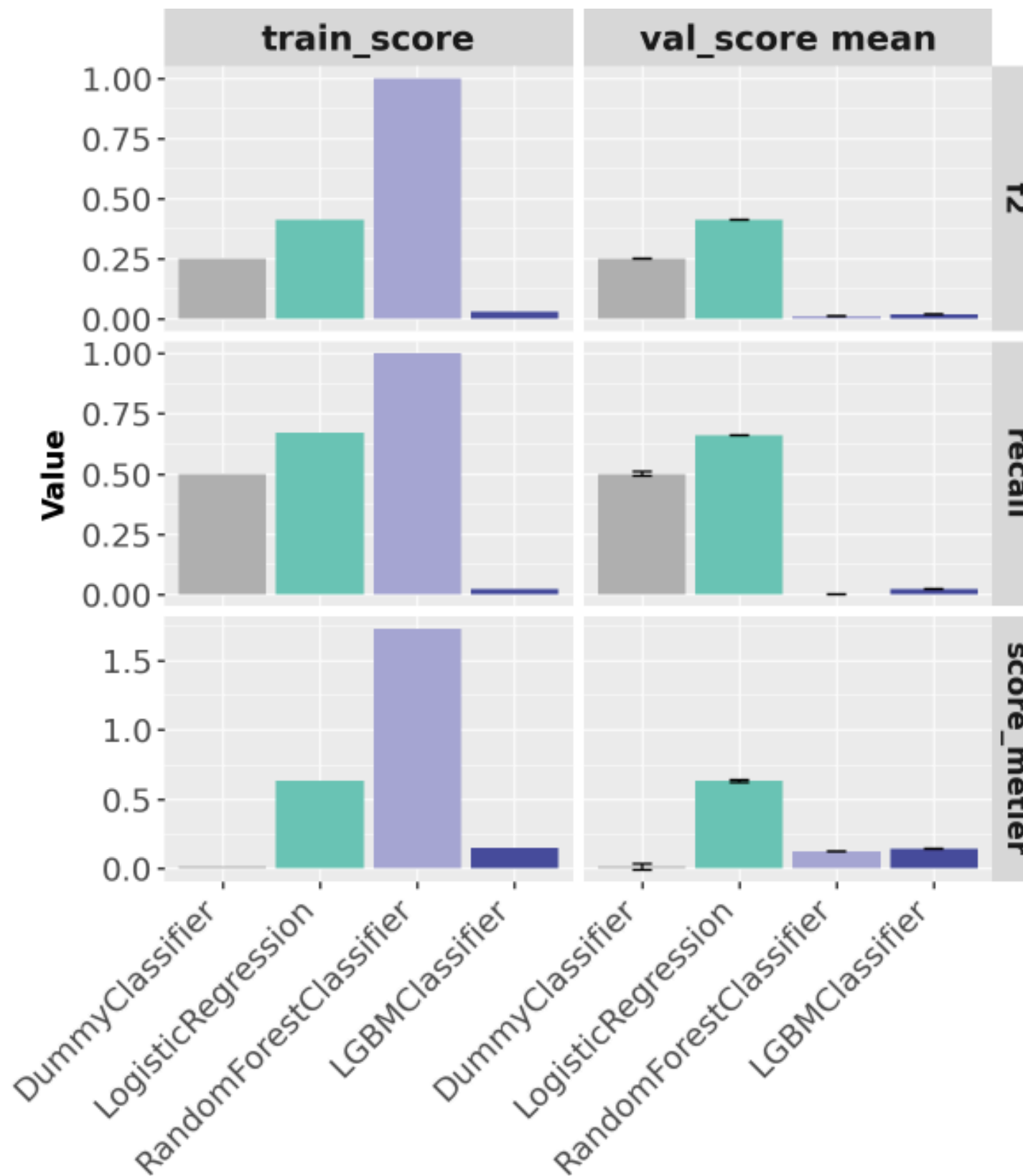
Optimisation



Explicabilité du  
modèle retenu



MLflow



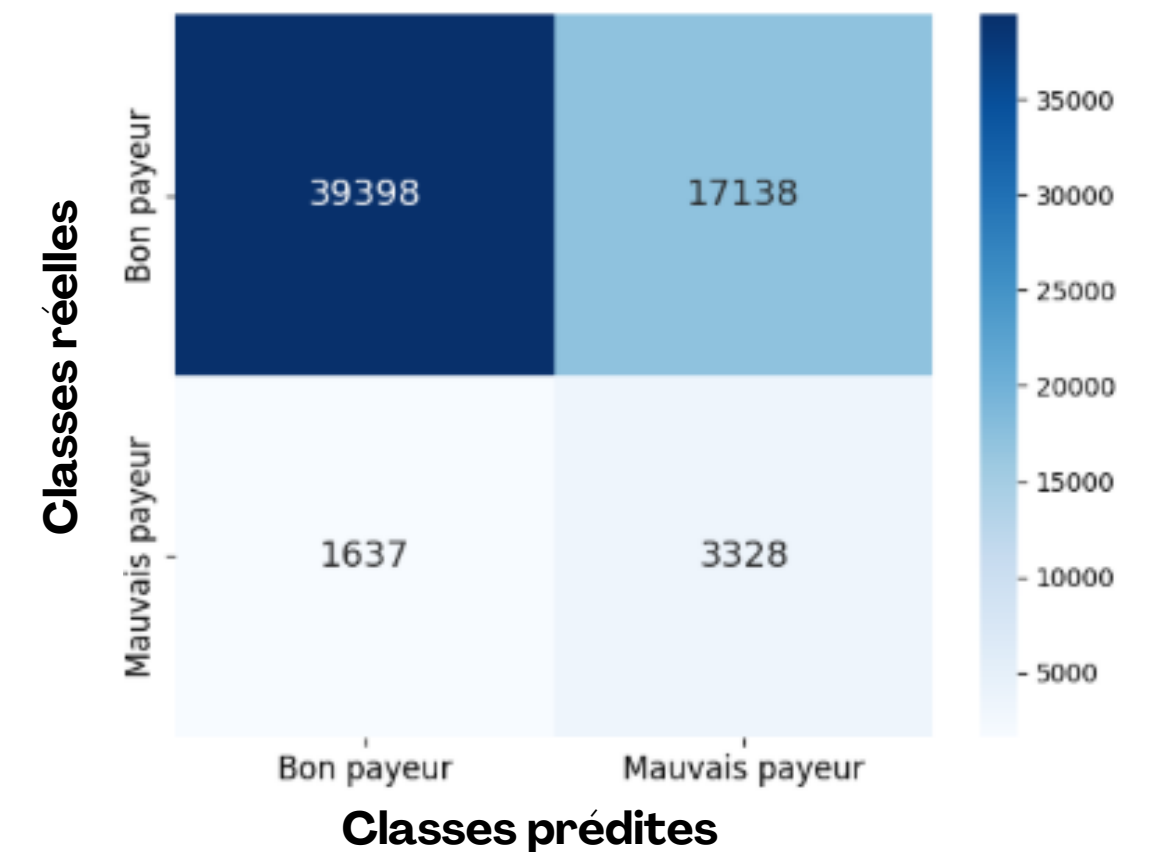
**Dummy** : rappel moyen, score métier faible

**Reg Log** : rappel moyen, score métier > 0

**Random Forest** : gros overfit

**Light GBM** : rappel moyen, score métier faible

Matrice de confusion



Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard

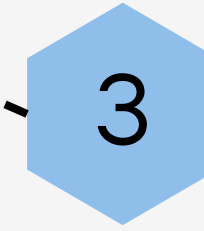
Prétraitement



2

Stratégie pour gérer  
la classe minoritaire

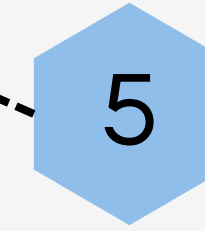
Exploration des  
modèles



4

Optimisation

Explicabilité du  
modèle retenu



6

MLflow

TN Bon payeur bien prédit	FP Bon payeur mal prédit
FN Mauvais payeur mal prédit	TP Mauvais payeur bien prédit

## Score métier

*Prêt: 10 000 - intérêt: 1 000*

TN: + 1 000

FP: - 1 000

FN: - 10 000

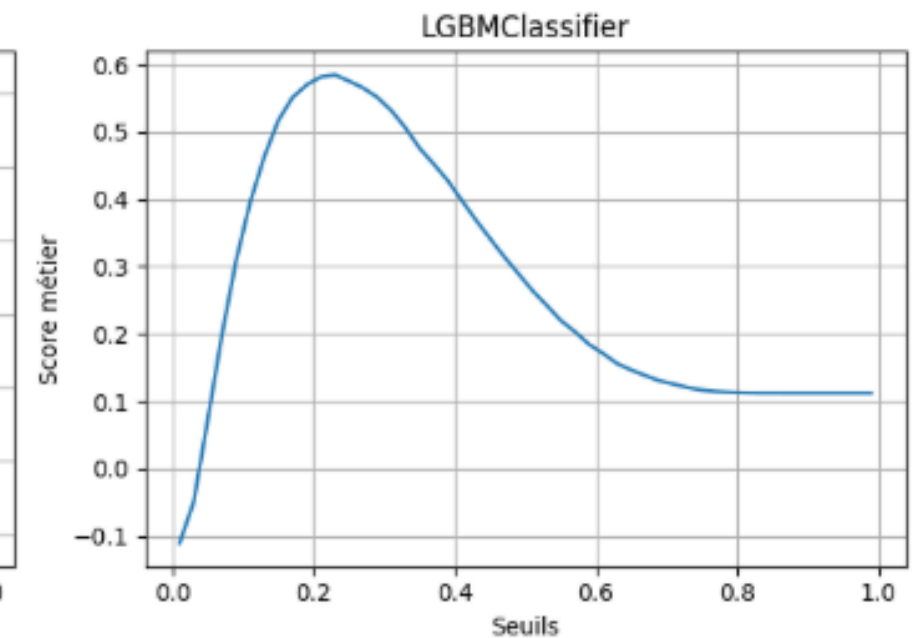
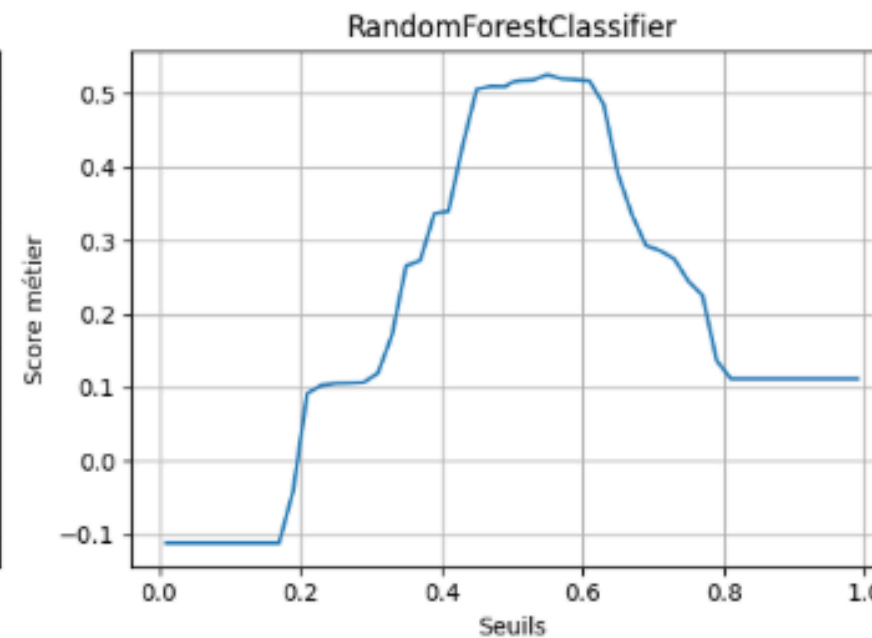
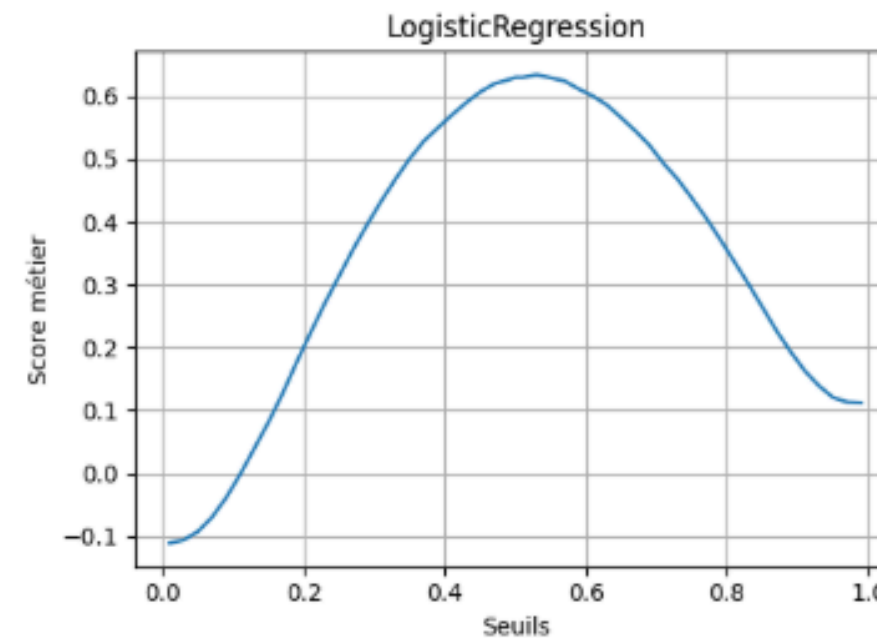
TP: + 10 000

$$\text{Score} = \frac{10 * TP + TN - 10 FN - FP}{TP + TN + FN + FP}$$

## Optimisation des hyperparamètres

- Des modèles
- De la fonction SMOTE

## Optimisation du seuil de classification



Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

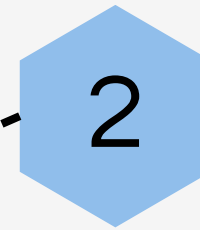
Modélisation

API +  
Dashboard

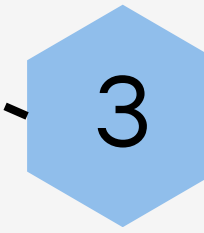
Prétraitement



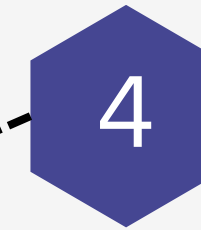
Stratégie pour gérer  
la classe minoritaire



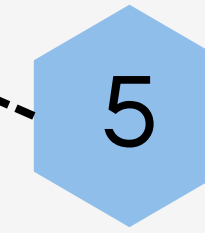
Exploration des  
modèles



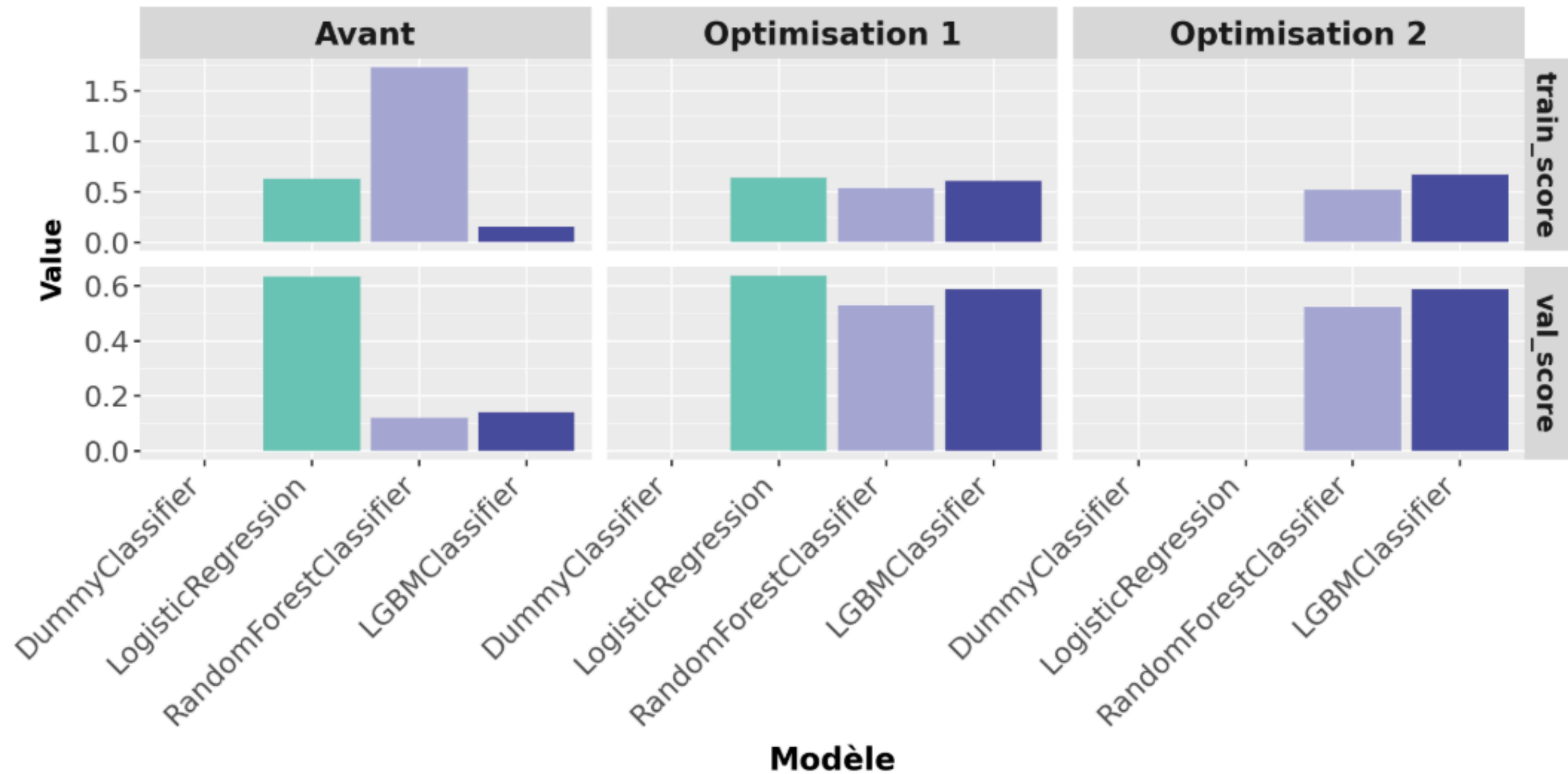
Optimisation



Explicabilité du  
modèle retenu



MLflow



Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard

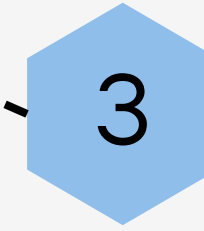
Prétraitement



2

Stratégie pour gérer  
la classe minoritaire

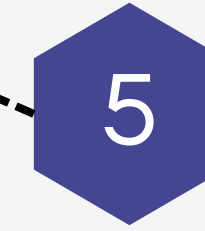
Exploration des  
modèles



4

Optimisation

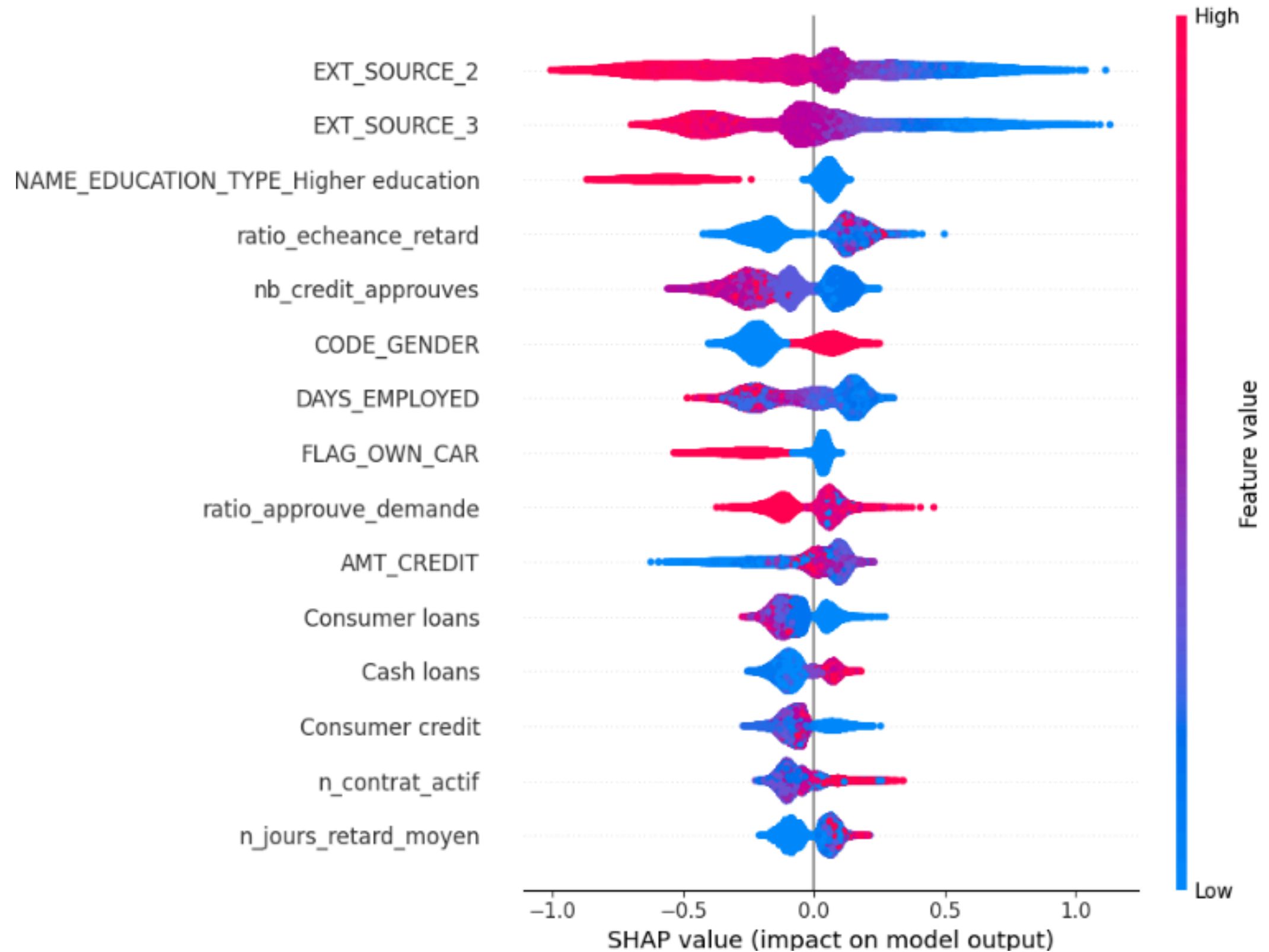
Explicabilité du  
modèle retenu



6

MLflow

Explicativité globale



Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard

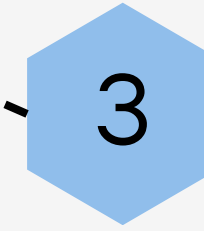
Prétraitement



2

Stratégie pour gérer  
la classe minoritaire

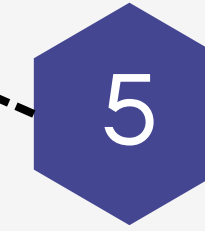
Exploration des  
modèles



4

Optimisation

Explicabilité du  
modèle retenu

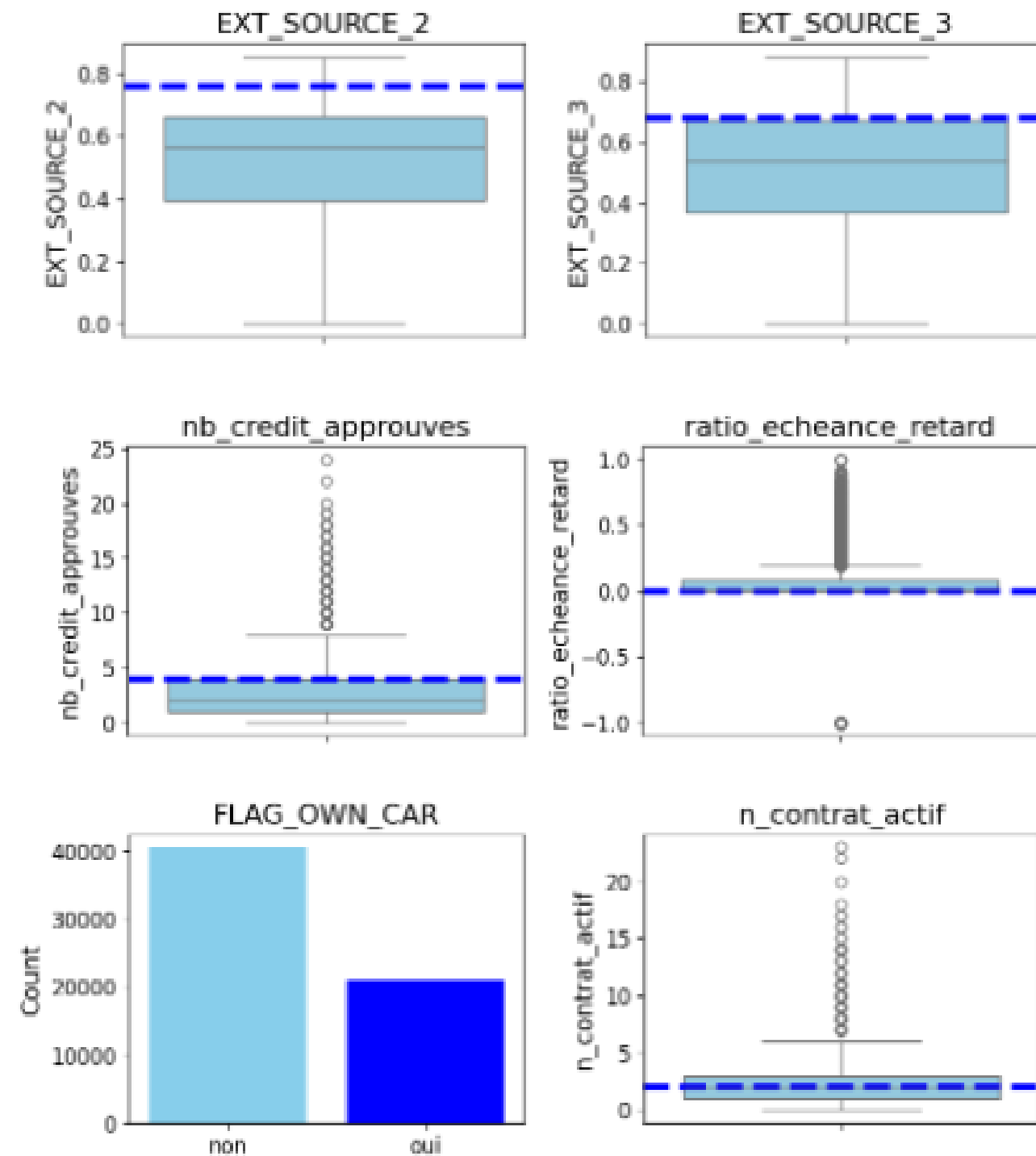
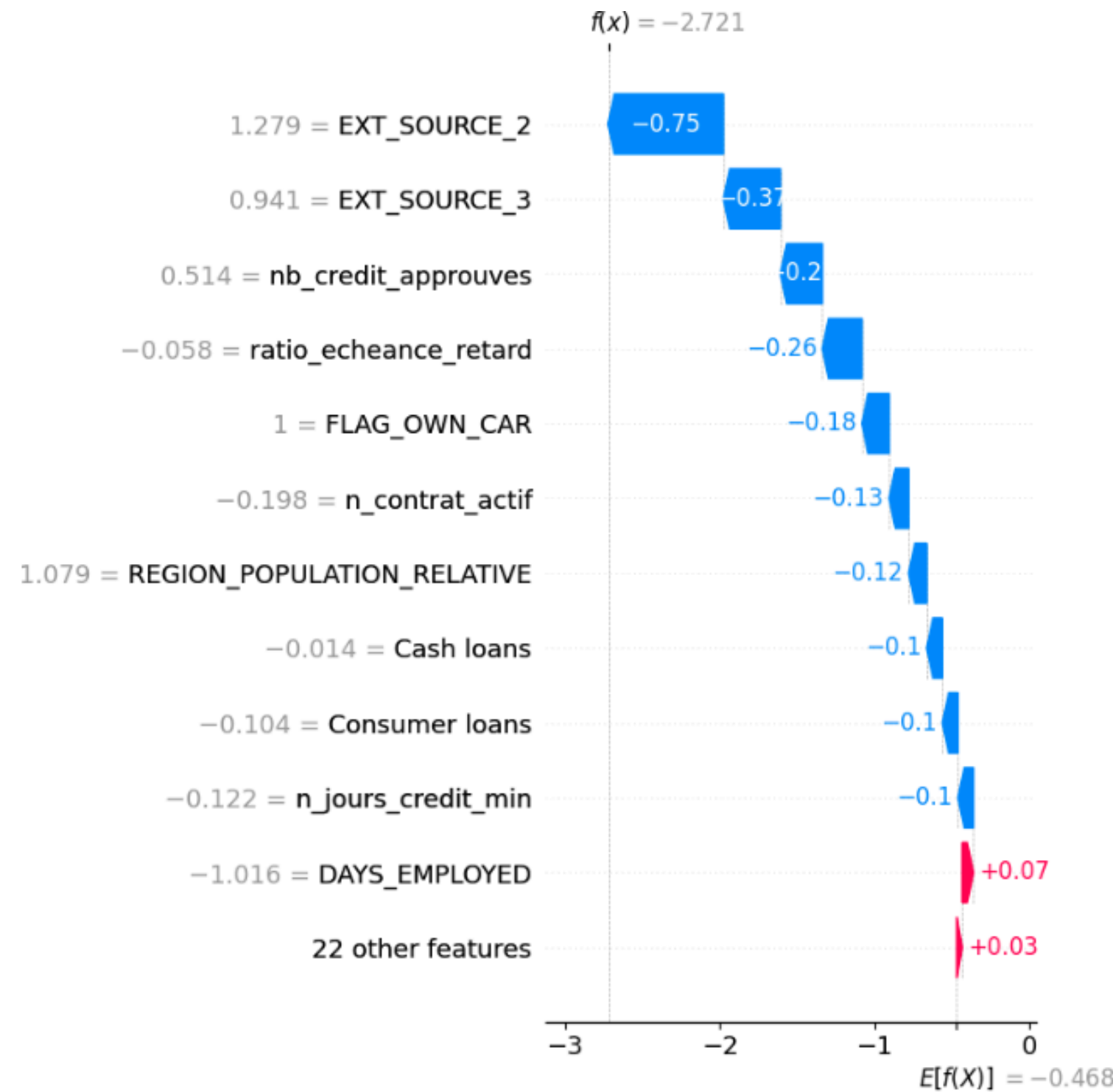


6

MLflow

Explicativité locale

Position de la demande par rapport à l'ensemble



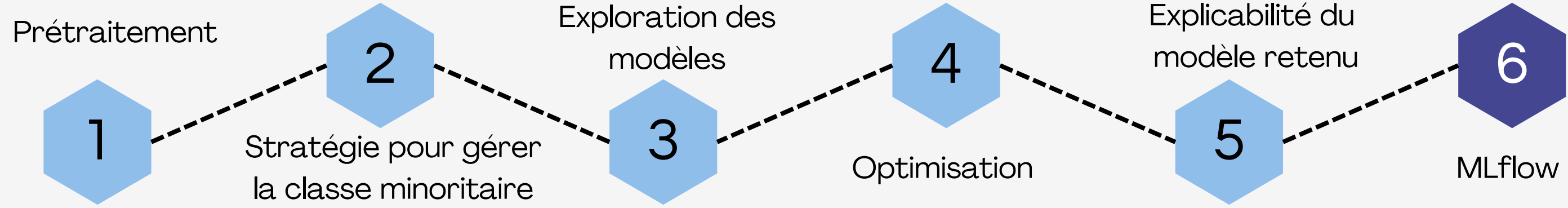
Présentation  
des données

Présentation  
de l'approche

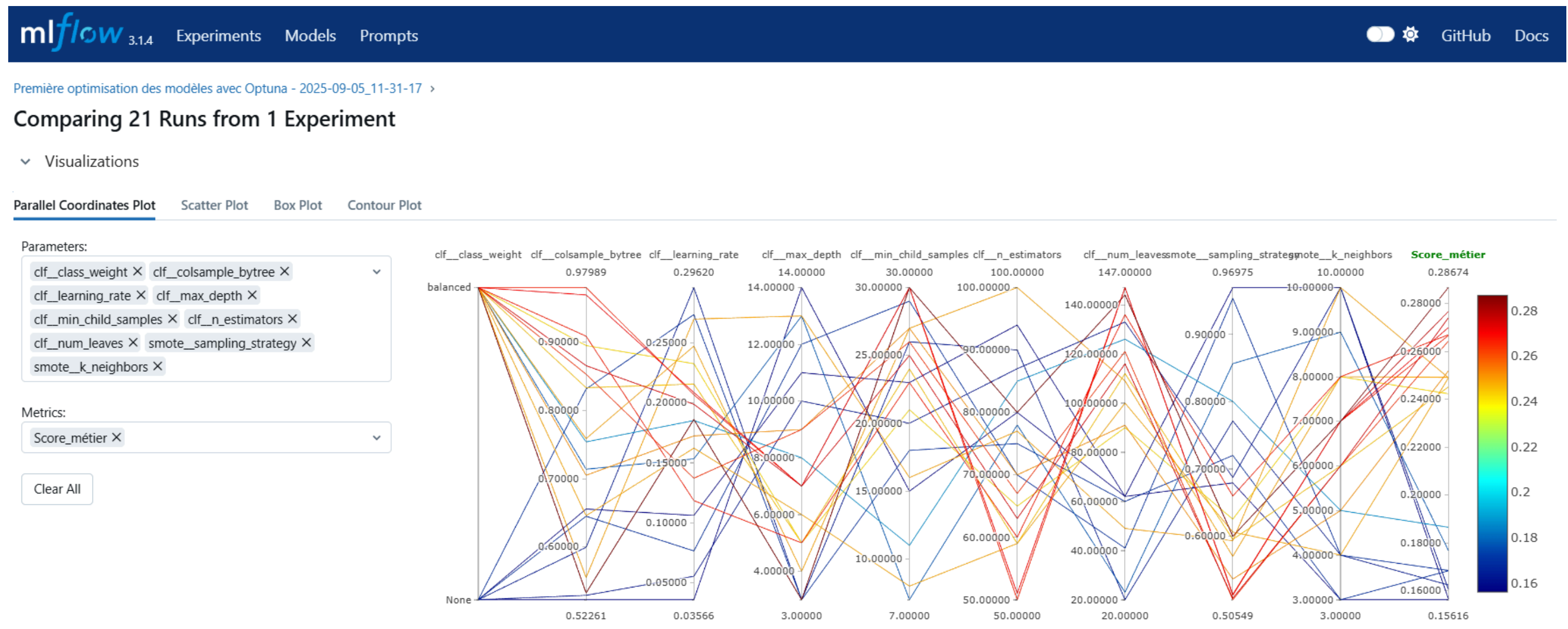
EDA + feature  
engineering

Modélisation

API +  
Dashboard



## Suivi de l'optimisation



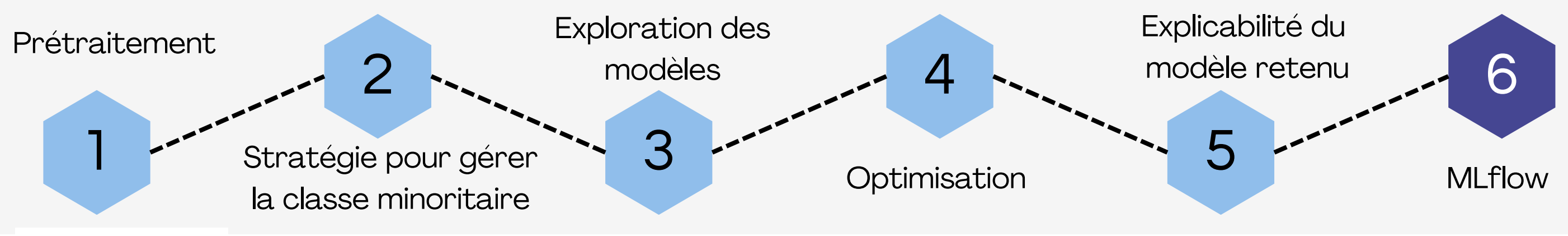
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



Sauvegarde des modèles

mlflow 3.1.4 Experiments Models Prompts GitHub Docs

Première optimisation des modèles avec Optuna - 2025-09-08\_21-05-33 > Models >

LGBMClassifier Register model

Overview Traces Artifacts

MLmodel 7.09KB

Path: file:///C:/Users/33647/Desktop/Openclassroom/Projet7/Nouv...

artifact\_path: file:///C:/Users/33647/Desktop/Openclassroom/Proj...

flavors:

- python\_function:
- env:
  - conda: conda.yaml
  - virtualenv: python\_env.yaml
- loader\_module: mlflow.sklearn
- model\_path: model.pkl
- predict\_fn: predict
- python\_version: 3.12.3
- sklearn:
  - code: null
  - pickled\_model: model.pkl
  - serialization\_format: cloudpickle
  - sklearn\_version: 1.4.2
- is\_signature\_from\_type\_hint: false
- mlflow\_version: 3.1.4
- model\_id: m-fbac621f13d041b39c16d3e3fda04996
- model\_size\_bytes: 6880854
- model\_uuid: m-fbac621f13d041b39c16d3e3fda04996

Sauvegarde d'artéfacts

mlflow 3.1.4 Experiments Models Prompts GitHub Docs

Optimisation modèles avec Optuna test - 2025-08-16\_10-36-40 >

Optimisation: LGBMClassifier(random\_state=42, verbosity=-1)

Overview Model metrics System metrics Traces Artifacts

plots

plots/LightGBM\_threshold\_optimization.png 30.34KB

Path: file:///C:/Users/33647/Desktop/Openclassroom/Projet7/Nouveau dossier/mlflow/mlruns/512...

Score métier en fonction du seuil de décision

Score métier (validation CV)

Seuil de décision

Seuil de décision	Score métier (validation CV)
0.0	-0.1
0.1	0.2
0.2	0.5
0.3	0.4
0.4	0.3
0.5	0.2
0.6	0.1
0.7	0.1
0.8	0.1
0.9	0.1
1.0	0.1

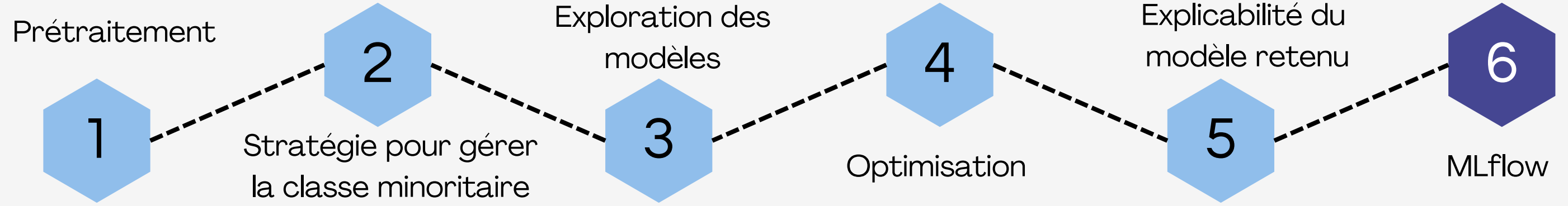
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



Métrique

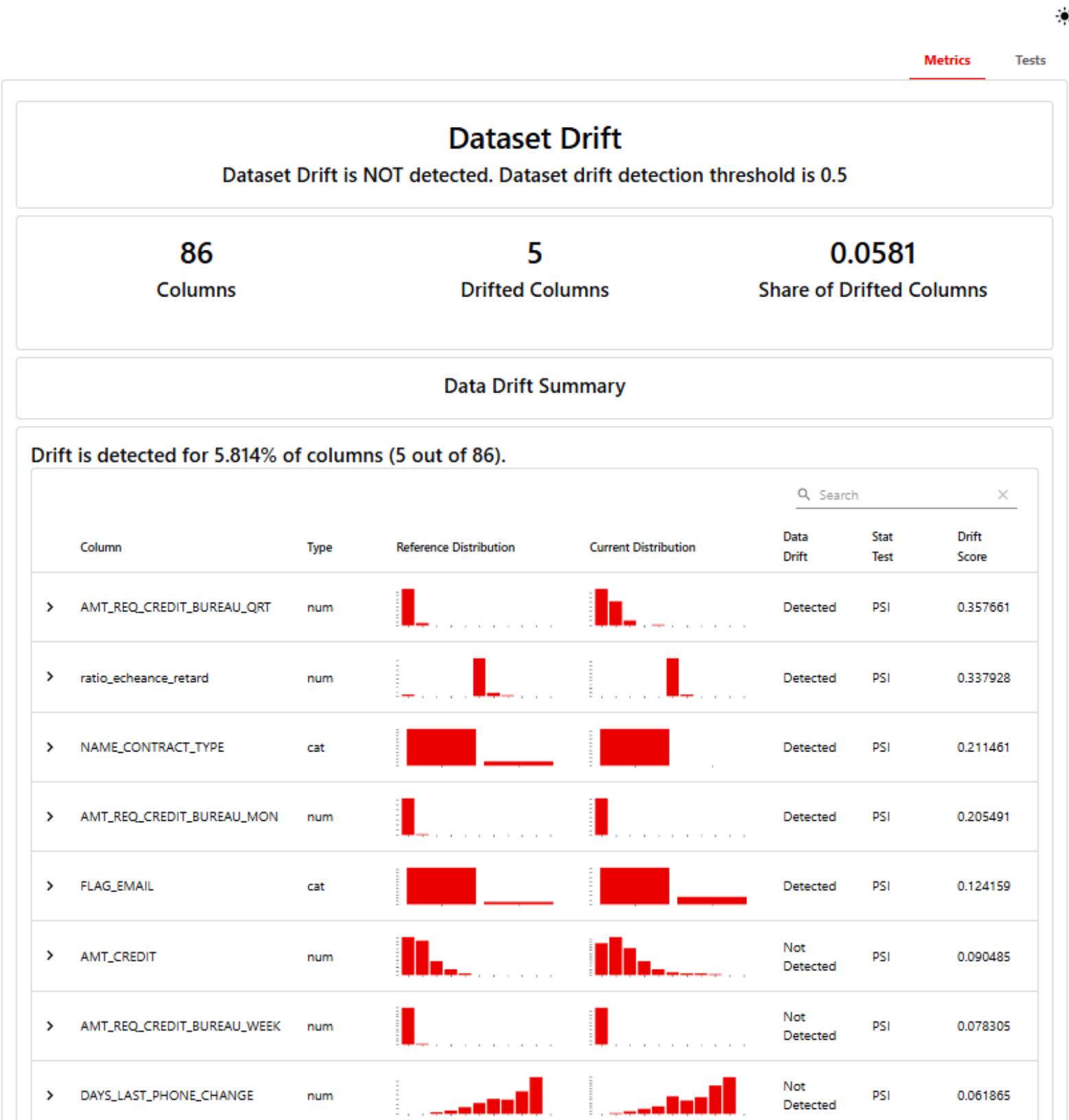
PSI

Résultat

- 94% des variables n'ont pas dérivé
- 6% des variables ont dérivé

Variables globalement stables dans le temps

 ratio\_echeance retard



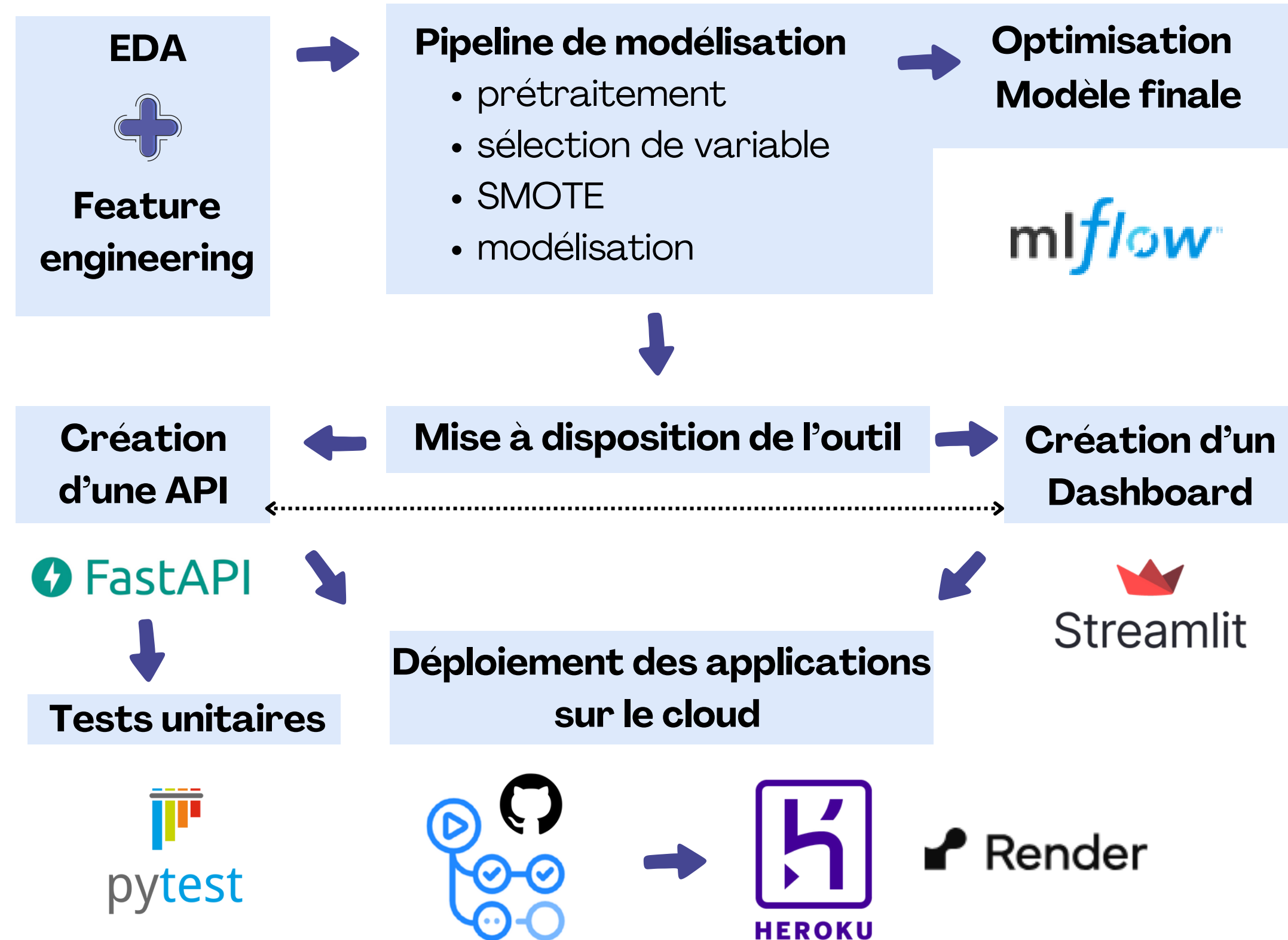
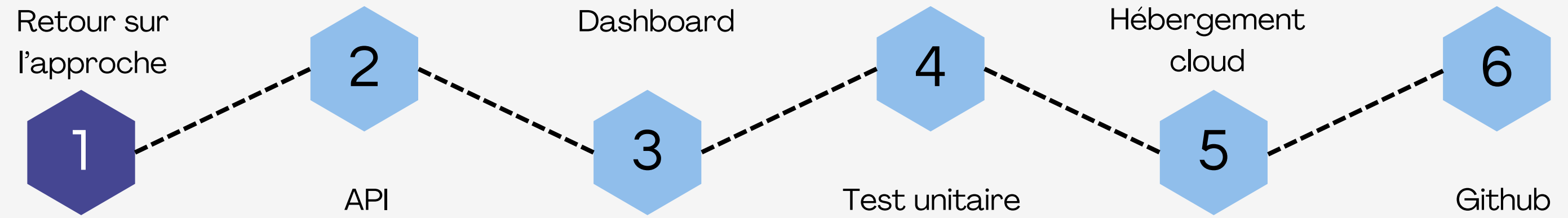
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



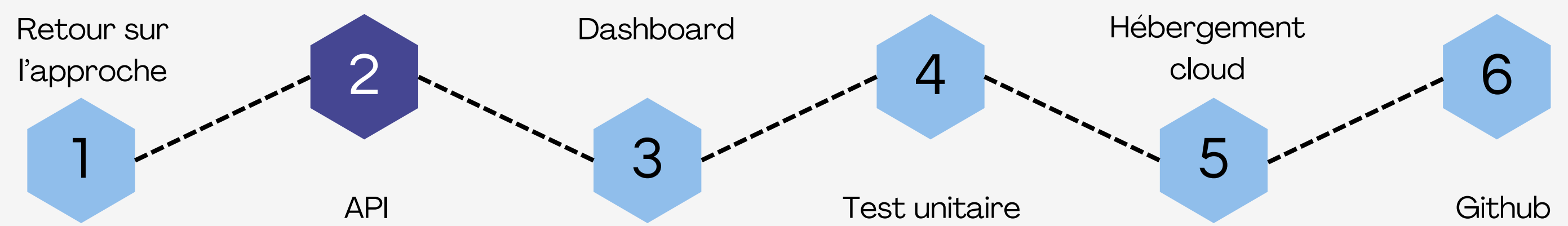
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



<https://api-oc-p7.onrender.com/docs#/>

FastAPI 0.1.0 OAS 3.1  
/openapi.json

default ^

GET / Read Root v

POST /predict Predict ^

Parameters Try it out

No parameters

Request body required application/json v

Example Value | Schema

```
{
  "client_id": 0
}
```

**But**

Rendre l'outil accessible via l'exposition de routes

**Route /**

Affiche un message d'accueil

**Route /predict**

**Entrée**

Identifiant d'une demande

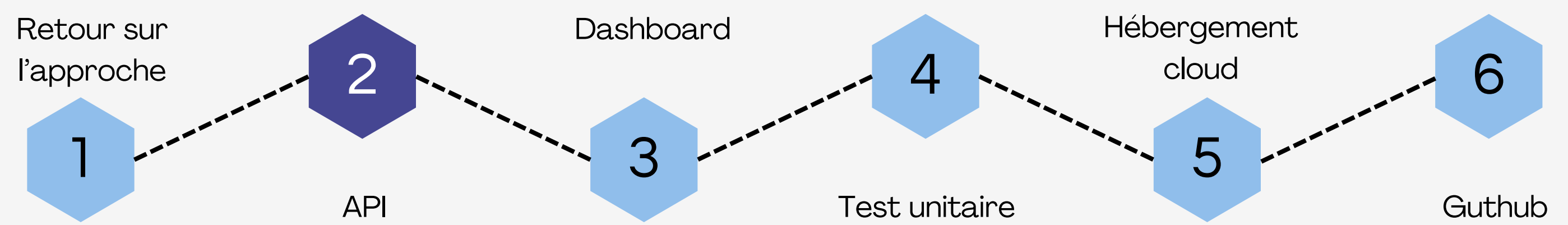
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



<https://api-oc-p7.onrender.com/docs#/>

Responses

Curl

```
curl -X 'POST' \
  'https://api-oc-p7.onrender.com/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "client_id": 411571
  }'
```

Request URL

```
https://api-oc-p7.onrender.com/predict
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "prediction": 1,   "proba": 0.43803789931925585 }</pre> <p>Response headers</p> <pre>cf-cache-status: DYNAMIC cf-ray: 9719b9d8fb0fd086-CDG content-encoding: br content-length: 46 content-type: application/json date: Tue, 19 Aug 2025 12:42:09 GMT rndr-id: 3dc53540-e2e3-4ead server: cloudflare vary: Accept-Encoding x-render-origin-server: uvicorn</pre>

**But**

Rendre l'outil accessible via l'exposition de routes

**Route /**

Affiche un message d'accueil

**Route /predict**

**Entrée**

Identifiant d'une demande

**Retourne**

Probabilité de défaut de  
paiement +  
Classe Bon payeur/Mauvais  
payeur

Utilise une pipeline de modélisation pré entraînée et  
les données de la table client stockées sur github

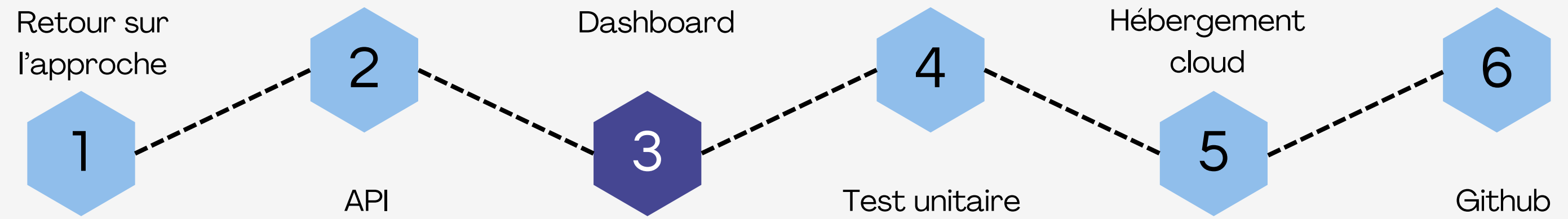
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



Streamlit

<https://oc-p7-cu77.onrender.com>

**But**

- Fournir un interface interactive et plus conviviale
- Afficher les résultats sous la forme de graphiques

**Requête vers l'API**

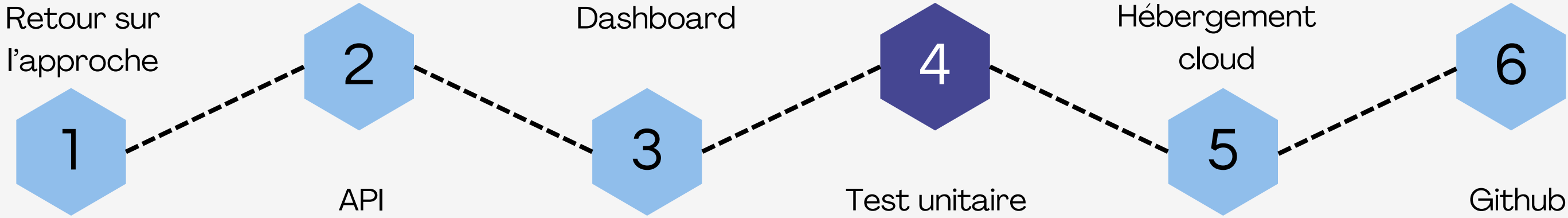
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



## rapport\_tests.html

Report generated on 19-Aug-2025 at 09:14:49 by [pytest-html](#) v4.1.1

### Environment

Python	3.12.3
Platform	Linux-6.11.0-1018-azure-x86_64-with-glibc2.39
Packages	<ul style="list-style-type: none"><li>pytest: 7.4.4</li><li>pluggy: 1.6.0</li></ul>
Plugins	<ul style="list-style-type: none"><li>metadata: 3.1.1</li><li>anyio: 4.10.0</li><li>html: 4.1.1</li></ul>
CI	true
JAVA_HOME	/usr/lib/jvm/temurin-17-jdk-amd64

### Summary

4 tests took 75 ms.

(Un)check the boxes to filter the results.

<input type="checkbox"/> 0 Failed,	<input checked="" type="checkbox"/> 4 Passed,	<input type="checkbox"/> 0 Skipped,	<input type="checkbox"/> 0 Expected failures,	<input type="checkbox"/> 0 Unexpected passes,	<input type="checkbox"/> 0 Errors,	<input type="checkbox"/> 0 Reruns	<a href="#">Show all details</a>
Result	Test	Duration					
Passed	Test/test_api.py::test_read_root	55 ms					
Passed	Test/test_api.py::test_predict_success	15 ms					
Passed	Test/test_api.py::test_predict_client_not_found	2 ms					
Passed	Test/test_api.py::test_predict_validation_error	3 ms					

But

Vérifier que l'API  
fonctionne comme  
attendue



Tests

Route d'entrée

Vérification du  
message d'accueil

Route prédicit

Vérification

- Modèle d'entrée
- Client non valide
- Client valide
  - réponse
  - type probas

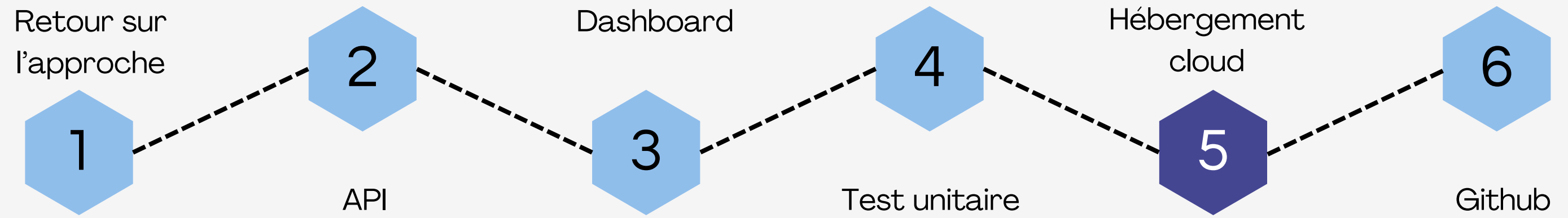
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



The screenshot shows the Render dashboard for a project named 'API\_OC\_p7'. The left sidebar contains navigation links: Dashboard, API\_OC\_p7, Events, Settings, MONITOR, Logs, and Metrics. The main content area displays the project details, including the service ID 'srv-d2gcv5ggjchc73b18u30', the repository 'AVit65 / OC\_P7\_Implementer\_un\_outil\_de\_scoring', and the deployment URL 'https://api-oc-p7.onrender.com'. A 'Manual Deploy' button is visible. A notification at the bottom states: 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. Upgrade now'. A deployment log entry shows a successful deployment for commit '2ca3b59' on August 27, 2025 at 10:00 AM.



**But**

Héberger l'API et le dashboard  
sur le cloud

→ deux services à configurer

**Configuration**

- Relier au repo Github
- déploiement a chaque commit
- Commande pour lancer l'app
- Commande pour construire l'environnement

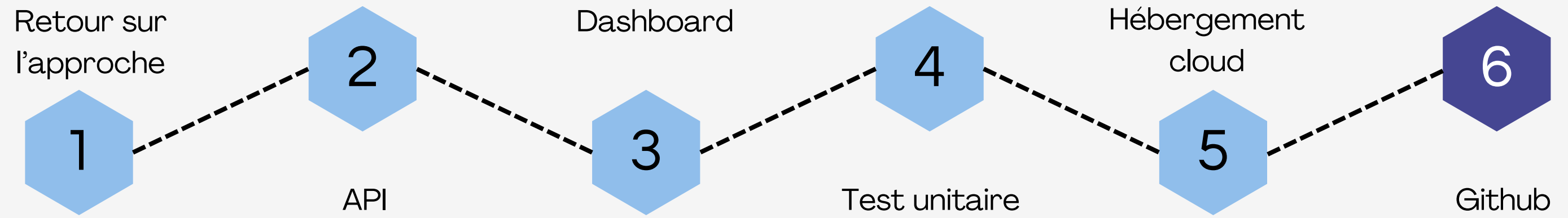
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



[https://github.com/AVit65/OC\\_P7\\_Implementer\\_un\\_outil\\_de\\_scoring](https://github.com/AVit65/OC_P7_Implementer_un_outil_de_scoring)

Platform Solutions Resources Open Source Enterprise Pricing Sign in Sign up

AVit65 / OC\_P7\_Implementer\_un\_outil\_de\_scoring Public Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main Go to file Code

AVit65 Ajout Notebook/Images + modif... e917da2 · 37 minutes ago 6 Commits

File	Commit	Time
.github/workflows	Modif workflows	4 days ago
API	Ajout Notebook/Images + modif...	37 minutes ago
Config	Config	last week
Images	Ajout Notebook/Images + modif...	37 minutes ago
Notebook	Ajout Notebook/Images + modif...	37 minutes ago
Output	Ajout Notebook/Images + modif...	37 minutes ago
Streamlit	Ajout Notebook/Images + modif...	37 minutes ago
Test	Premier Dépôt	last week
Utils	Premier Dépôt	last week
.gitignore	Premier Dépôt	last week
.python-version	Premier Dépôt	last week
Readme.md	Ajout Notebook/Images + modif...	37 minutes ago

About: No description, website, or topics provided.

Readme Activity 0 stars 0 watching 0 forks Report repository

Releases: No releases published

Packages: No packages published

Languages: Jupyter Notebook 60.9%, HTML 39.0%, Python 0.1%



Guthub  
Action



But

- Héberger le code source et les données
- Gérer les versions du code
- Automatiser des workflows

Doit contenir

- Le code et les data
- Un fichier Requirements
- Un fichier Readme

Nécessite une architecture  
organisée

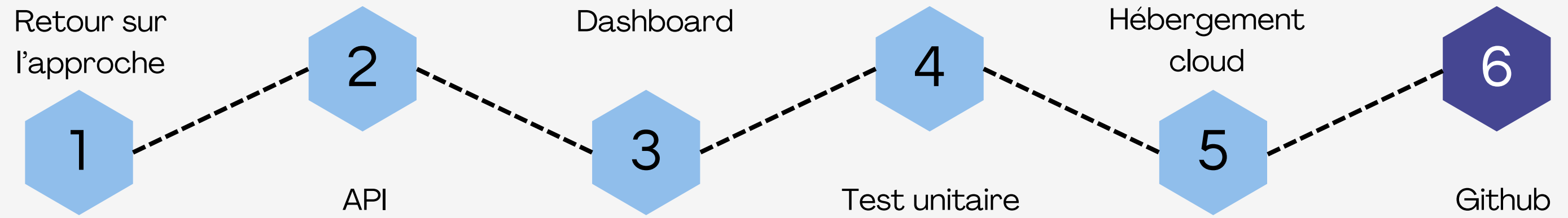
Présentation  
des données

Présentation  
de l'approche

EDA + feature  
engineering

Modélisation

API +  
Dashboard



Event	Status	Branch	Actor	Run Name	Time
✓	Completed	main	AVit65	Ajout Notebook/Images + modif API/...	40 minutes ago
✓	Completed	main	AVit65	Modif workflows	4 days ago
✓	Completed	main	AVit65	Modif API	last week
✓	Completed	main	AVit65	Modif API	last week
✓	Completed	main	AVit65	Premier Dépôt	last week



**Guthub  
Action**



**But**

- Héberger le code source et les données
- Gérer les versions du code
- Automatiser des workflows

**Workflows**

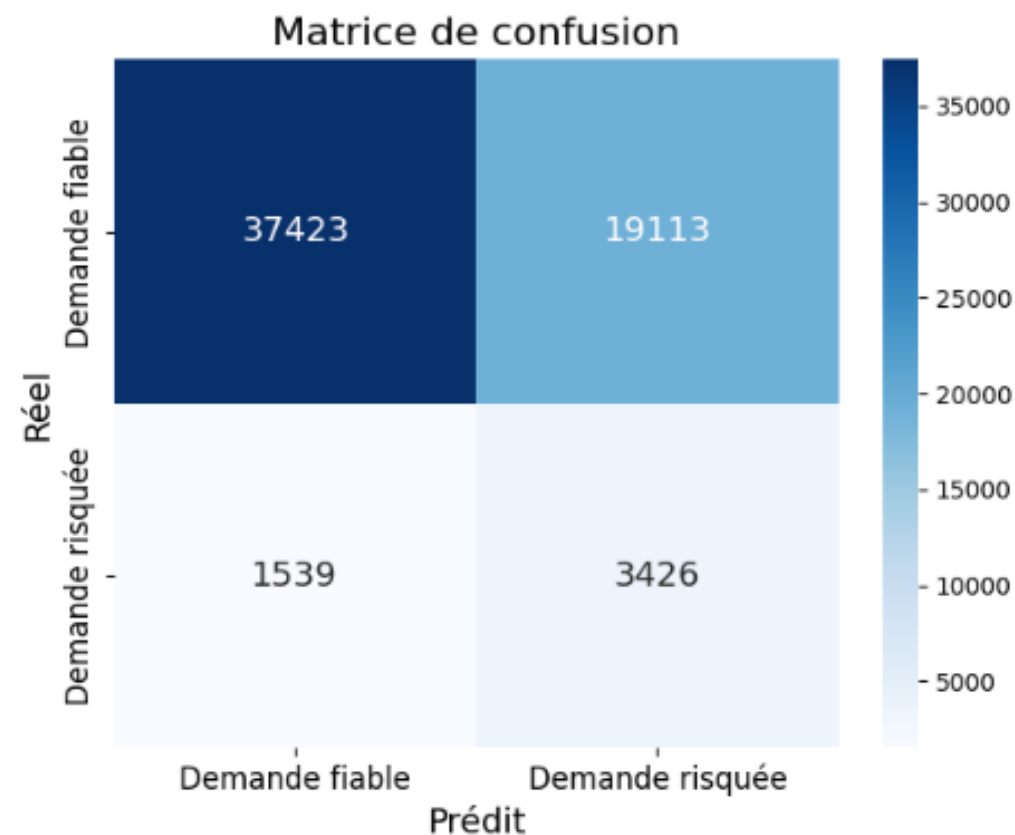
- Test automatique de l'API à chaque push
- Déploiement de l'API et du dashboard automatiquement à chaque push

# Conclusions

Implementer un outil de scoring

Suivre l'optimisation via MLFlow

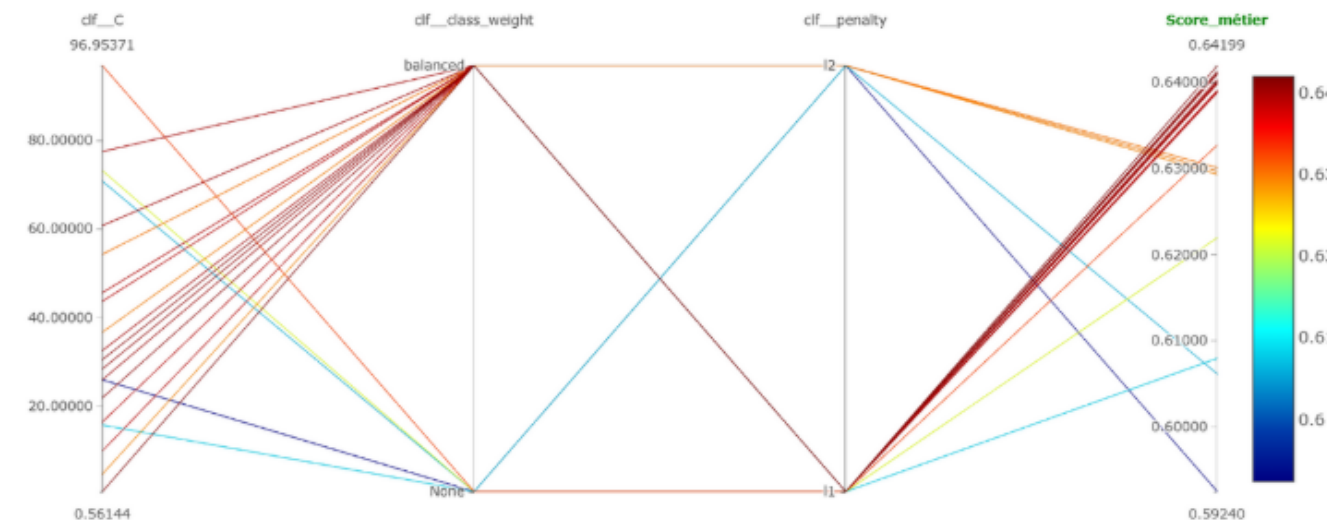
Deployer le modèle dans le cloud



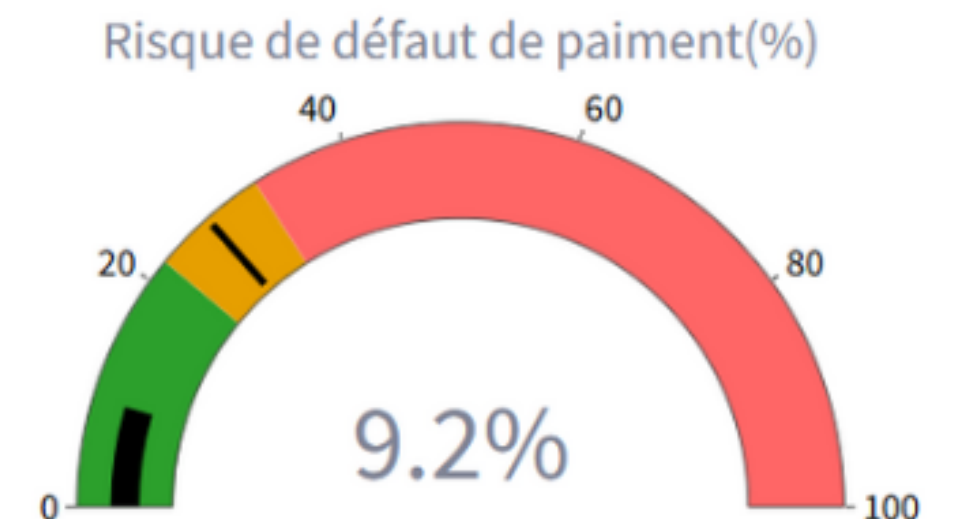
- Light GBM
- 65 % des mauvais payeurs
- 70 % des bons payeurs

mlflow

- Suivre de l'optimisation
- Sauvegarde des modèles



- Développement d'un API
- Développement d'un dashboard



**Merci pour votre attention**