

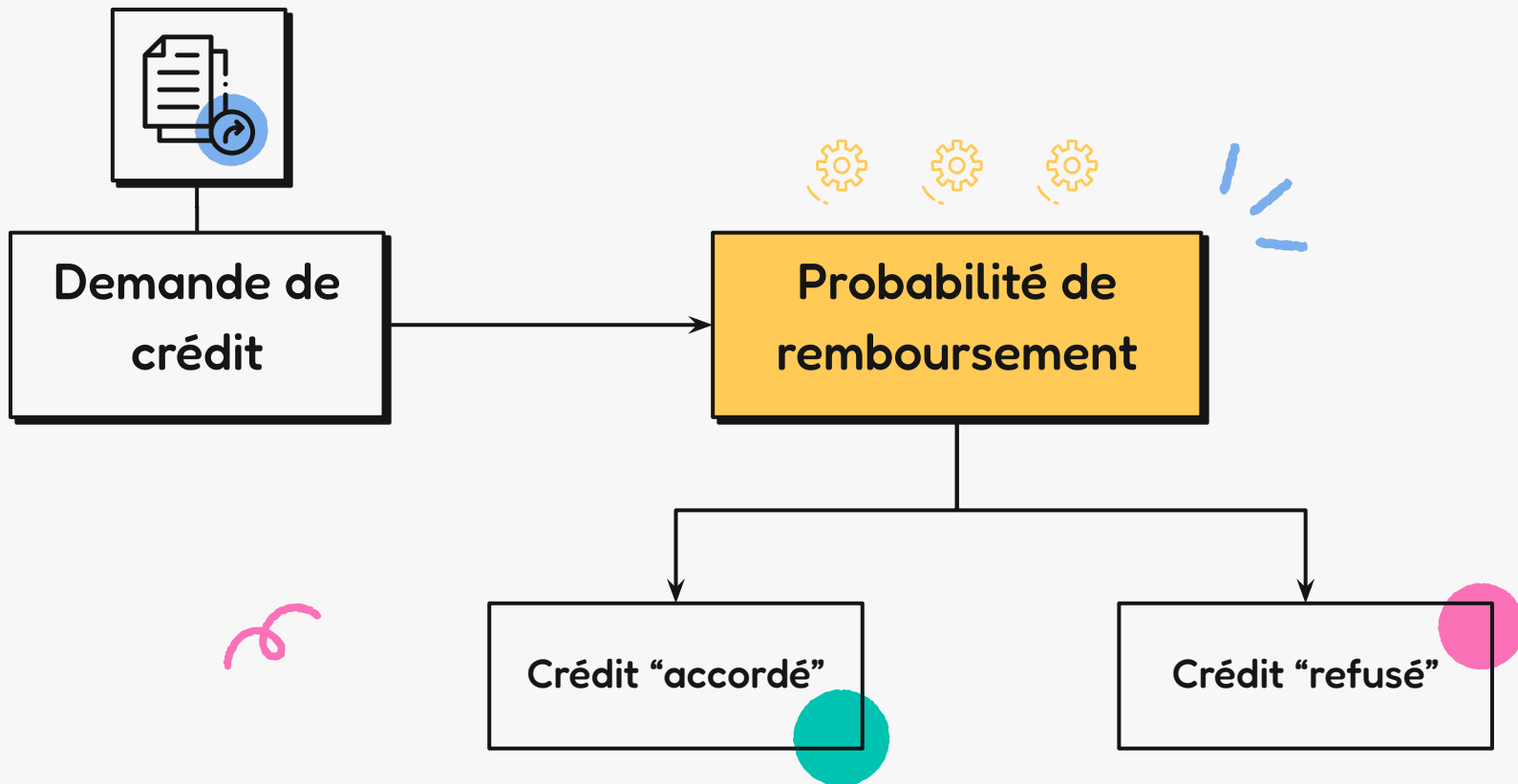


# Implémentez un modèle de scoring

Soutenance Projet 7  
OpenClassrooms  
Formation Data Scientist  
24 Juillet 2023



# Outil de scoring



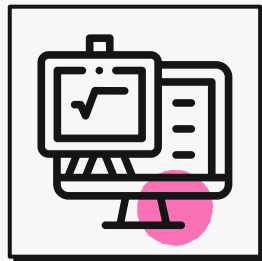
# Missions



## Construire et mettre en production



**Modèle de scoring**

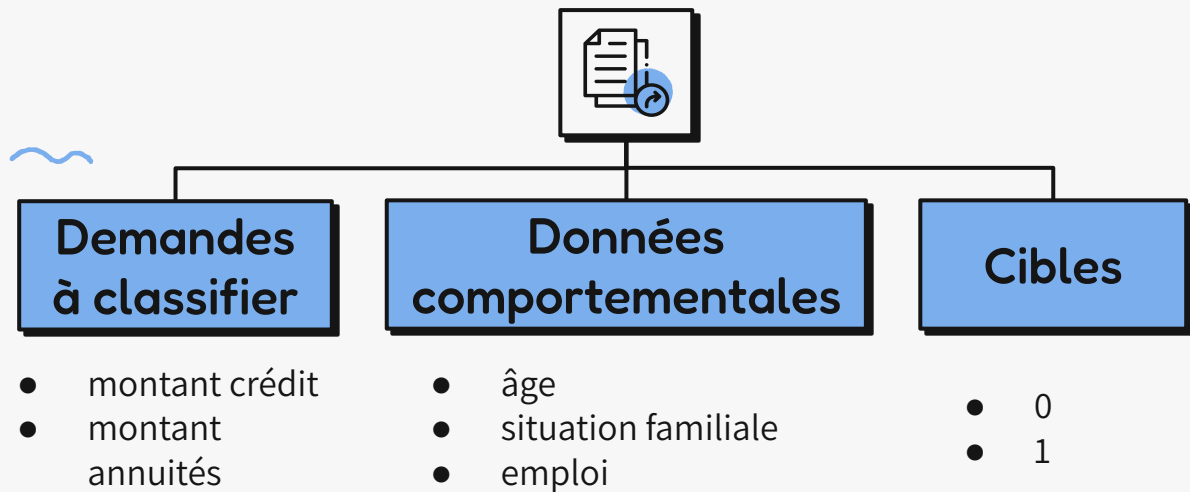


**Dashboard chargés  
de clientèle**

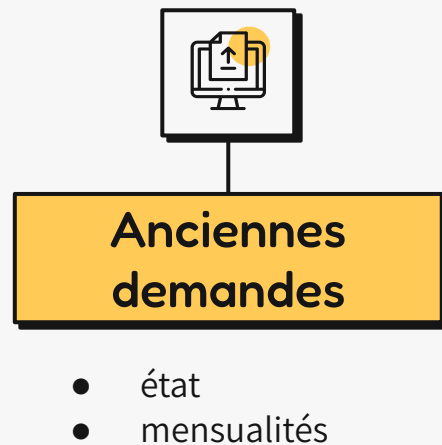
# Données sources



application\_train.csv

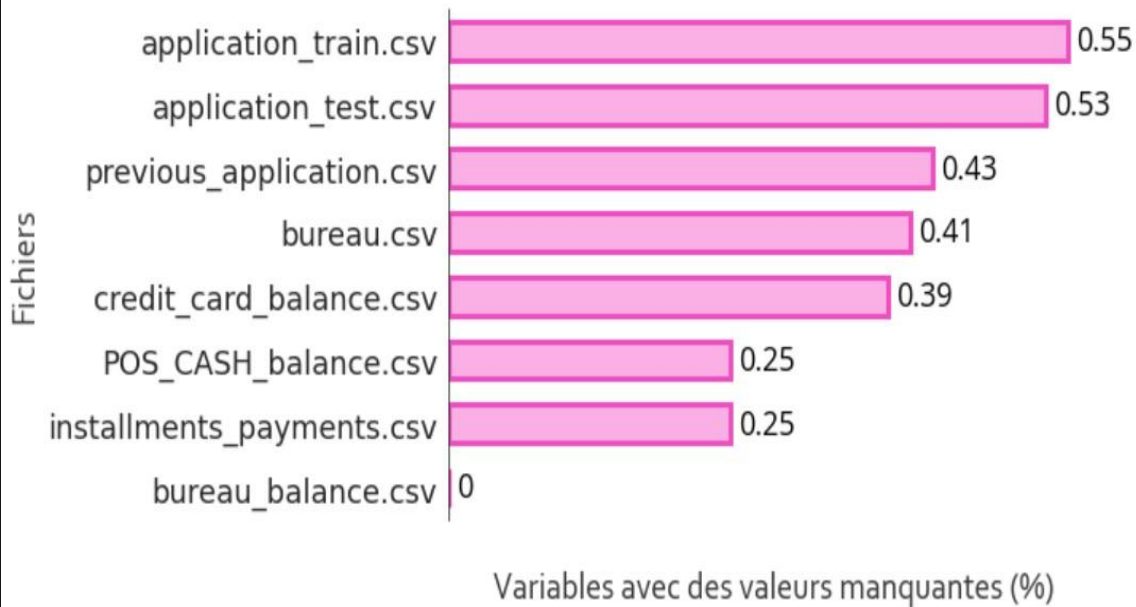


6 autres fichiers .csv



# Analyse exploratoire

Pourcentage de variables avec des valeurs manquantes  
- par fichier -

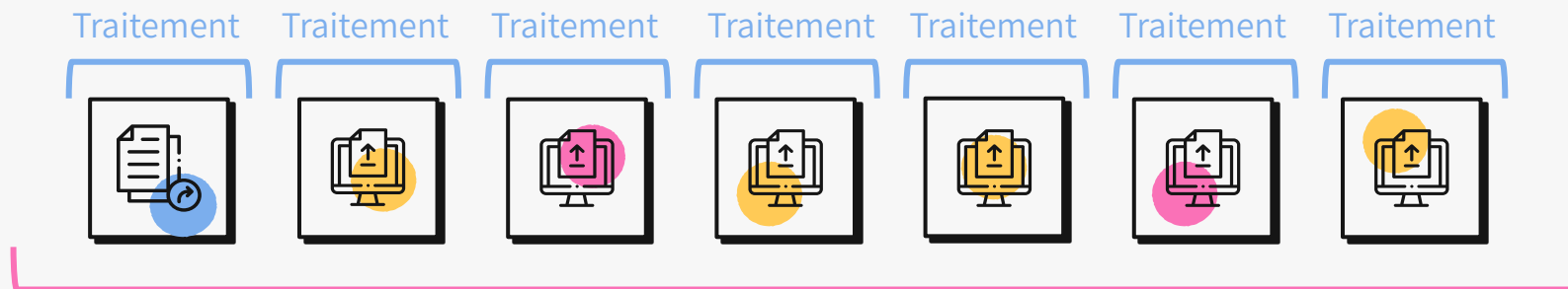


**\*0 Doublons**

# Traitement des données



Nettoyage + Agrégation + Feature Engineering



Jointure à gauche



Nouveau jeu de données : 307k lignes, 458 colonnes  
Une ligne = Une demande de crédit



# Plan de la soutenance



**01**

**Démarche de modélisation**

**02**

**Pipeline de déploiement continu**

**03**

**Analyse du data drift**

**04**

**Démonstration Dashboard**



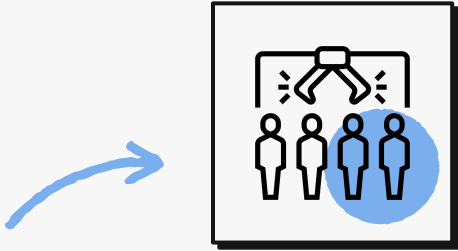
01

# Démarche de **modélisation**

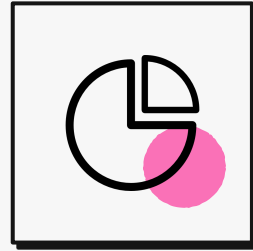




# Préparation des données



**Echantillon  
représentatif**



**Séparation  
Test/Train**



**Transformations**





# Transformations



## Features catégorielles

- Binaires: *OrdinalEncoder*
- Autres: *One hot encoding*

## Features numériques

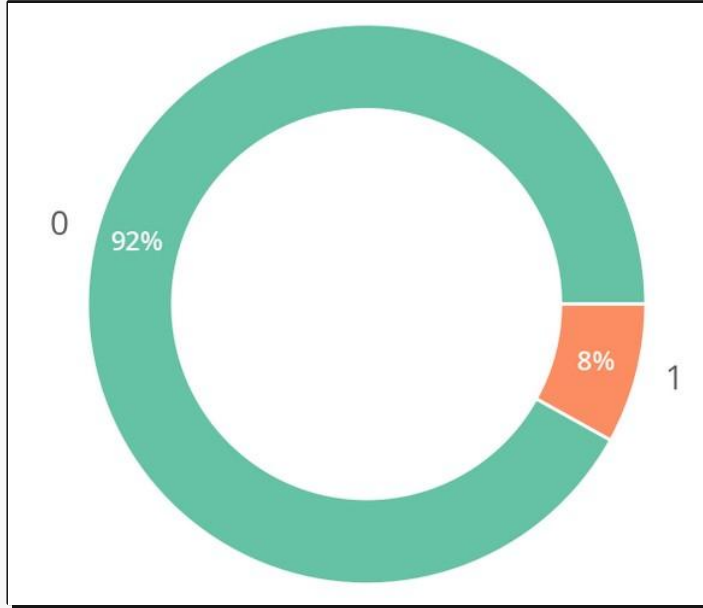
- *RobustScaler*
- *SimpleImputer* (médiane)

+ Ré-échantillonnage



# Déséquilibre des classes

Répartition de la cible dans le jeu d'entraînement



## Stratégies possibles :

**Oversampling**



**Undersampling**



# RandomUnderSampler

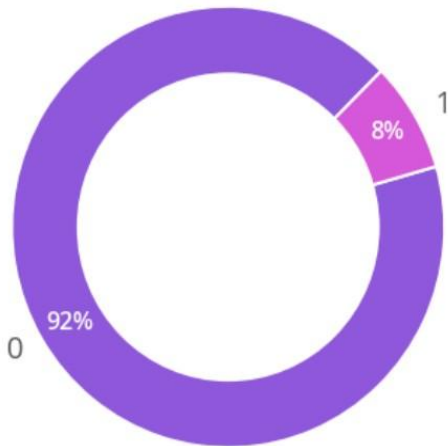


- Perte d'informations

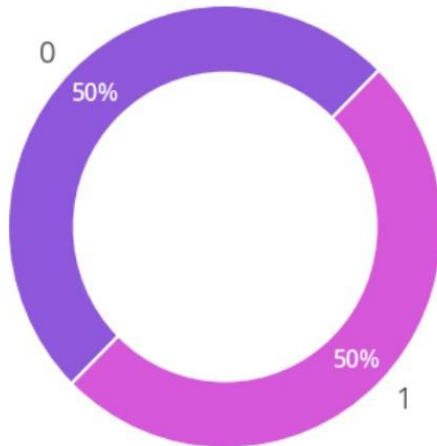


- rapide
- taille du jeu de données n'augmente pas
- valeurs manquantes ok

avant (107\_627 individus)

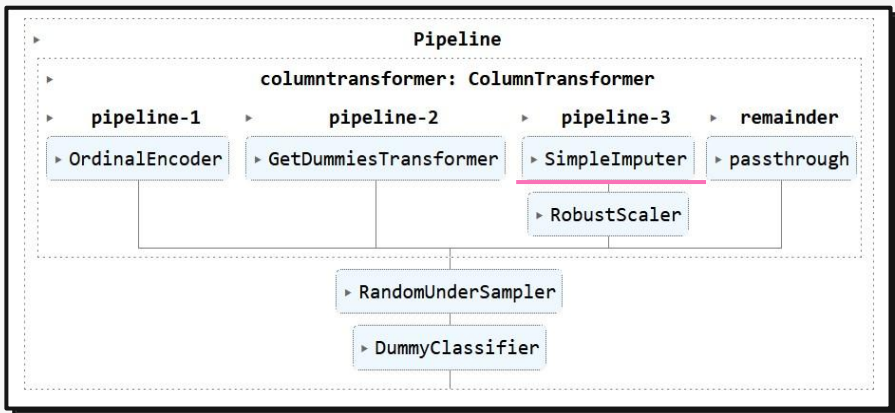


après (17\_376 individus)



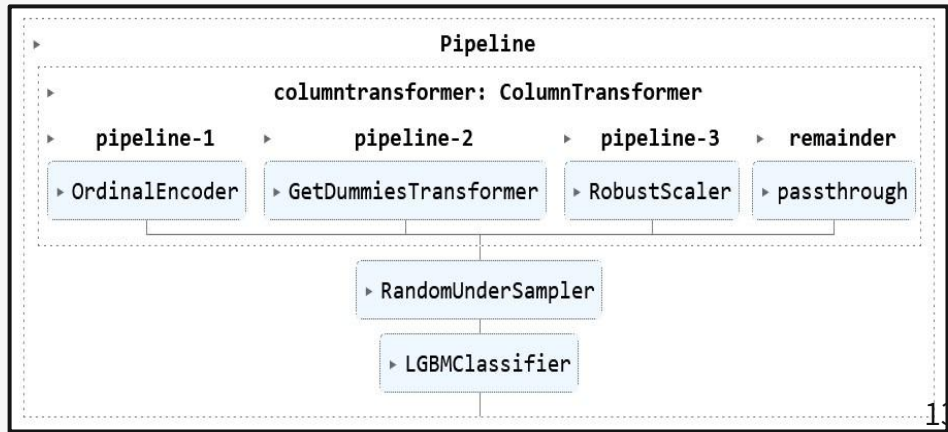
Répartition de la cible dans le jeu d'entraînement

# Pipelines

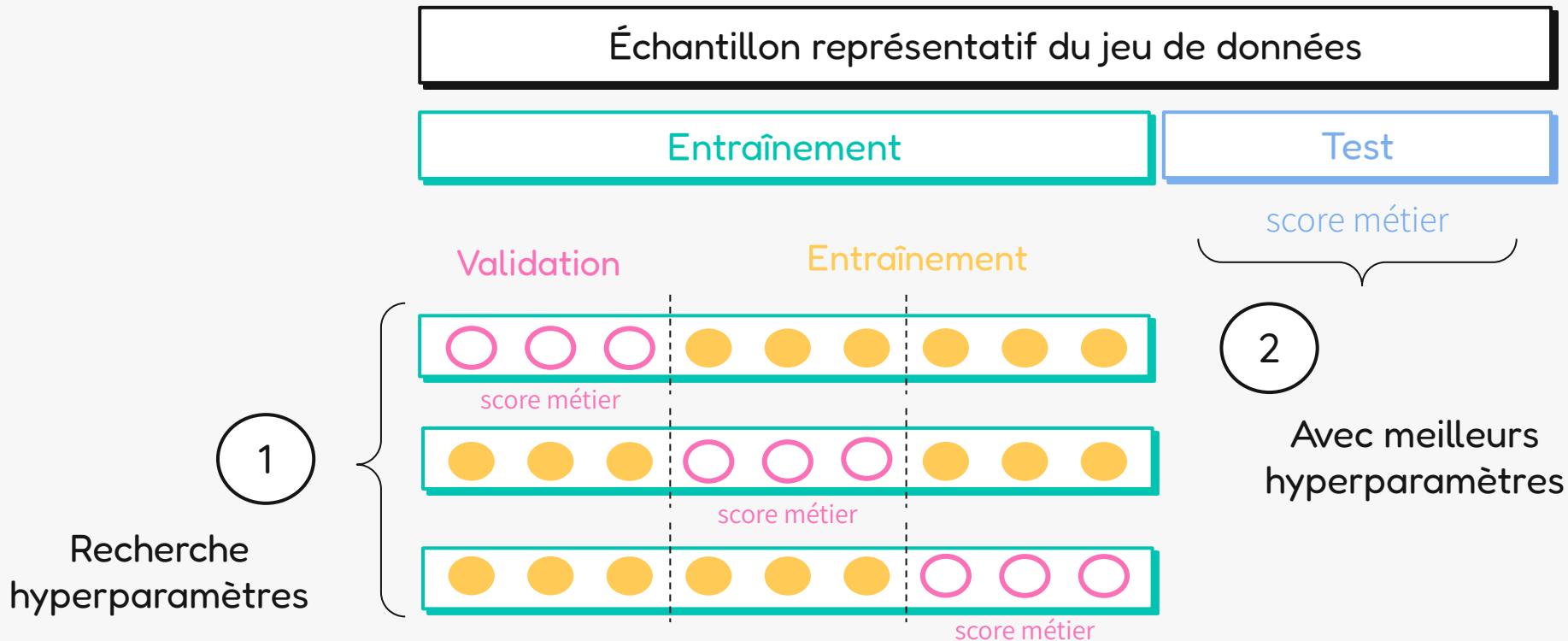


Avec Imputation

Sans Imputation



# Entraînement



# MLflow Tracking - Code

Notebook Jupyter

```
# Log parameter, metrics, and model to MLflow
with mlflow.start_run():
    grid.fit(X, y)

    best_model_index = grid.best_index_
    mlflow.log_metric("mean_time_fit",
                     grid.cv_results_['mean_fit_time'][best_model_index])
    mlflow.log_metric("std_time_fit",
                     grid.cv_results_['std_fit_time'][best_model_index])

    for param, best_value in grid.best_params_.items():
        mlflow.log_param(param, best_value)

    mlflow.sklearn.log_model(grid, "model")
```

# Score métier



Matrice de confusion\*



		Classes Réelles	
		<b>—</b> Client pas en défaut/Classe 0	<b>+</b> Client en défaut/Classe 1
Classes Prédites	<b>—</b> Pas défaut/0	Vrais Négatifs	Faux Négatifs (erreur de type 2)
	<b>+</b> Défaut/1	Faux Positifs (erreur de type 1)	Vrais Positifs

\*normalisée



# Score métier



		Matrice de coûts	
		Classes Réelles	
		— Client pas en défaut/Classe 0	+ Client en défaut/Classe 1
Classes Prédites	— Pas défaut/0	0 Vrais Négatifs	10 Faux Négatifs (erreur de type 2)
	+ Défaut/1	1 Faux Positifs (erreur de type 1)	0 Vrais Positifs

Score métier =  $\frac{10 \cdot \text{FN} + \text{FP}}{10}$  ← À minimiser

# Autres métriques

À maximiser

Balanced Accuracy

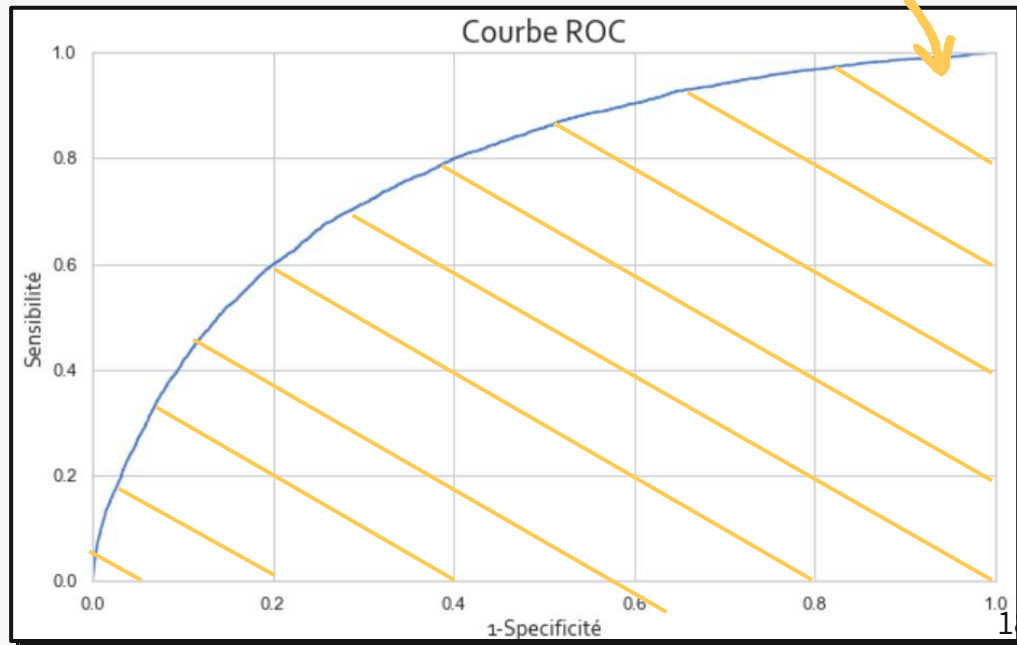
AUC

Sensitivity + Specificity

2


- Sensitivity =  $\frac{TP}{TP + FN}$

- Specificity =  $\frac{TN}{TN + FP}$



# Résultats - jeu de test

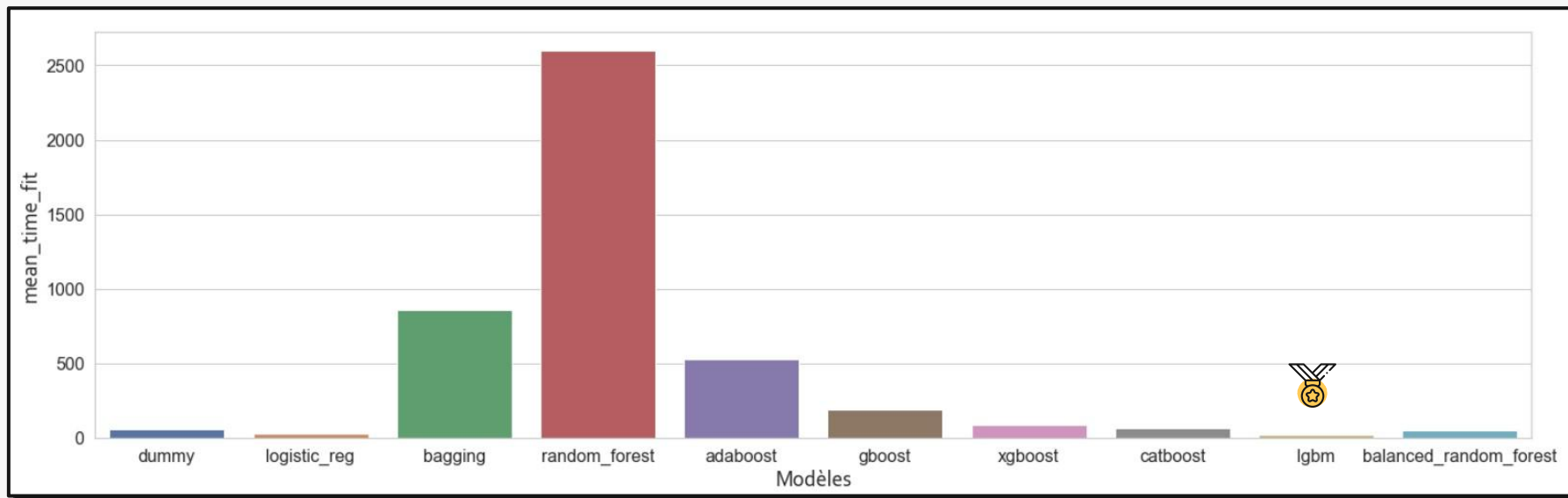


Modèle		Seuil	Score Métier	Balanced accuracy	AUC
LightGBM		0.547	0.051	0.704	0.704
GBoost		0.533	0.051	0.703	0.703
CatBoost		0.523	0.052	0.695	0.695
XGBoost		0.619	0.054	0.680	0.680
AdaBoost		0.496	0.054	0.685	0.685
Bagging		0.690	0.062	0.620	0.620
Forêt Aléatoire		0.694	0.063	0.615	0.615
Régression logistique		0.506	0.077	0.525	0.525
Dummy		0.001	0.081	0.500	0.500



# LightGBM

Temps d'entraînement moyen





# LightGBM

**Meilleurs  
résultats**

**Plus  
rapide**

**Gère valeurs  
manquantes**

**Optimisation bayésienne**



**HyperOpt**

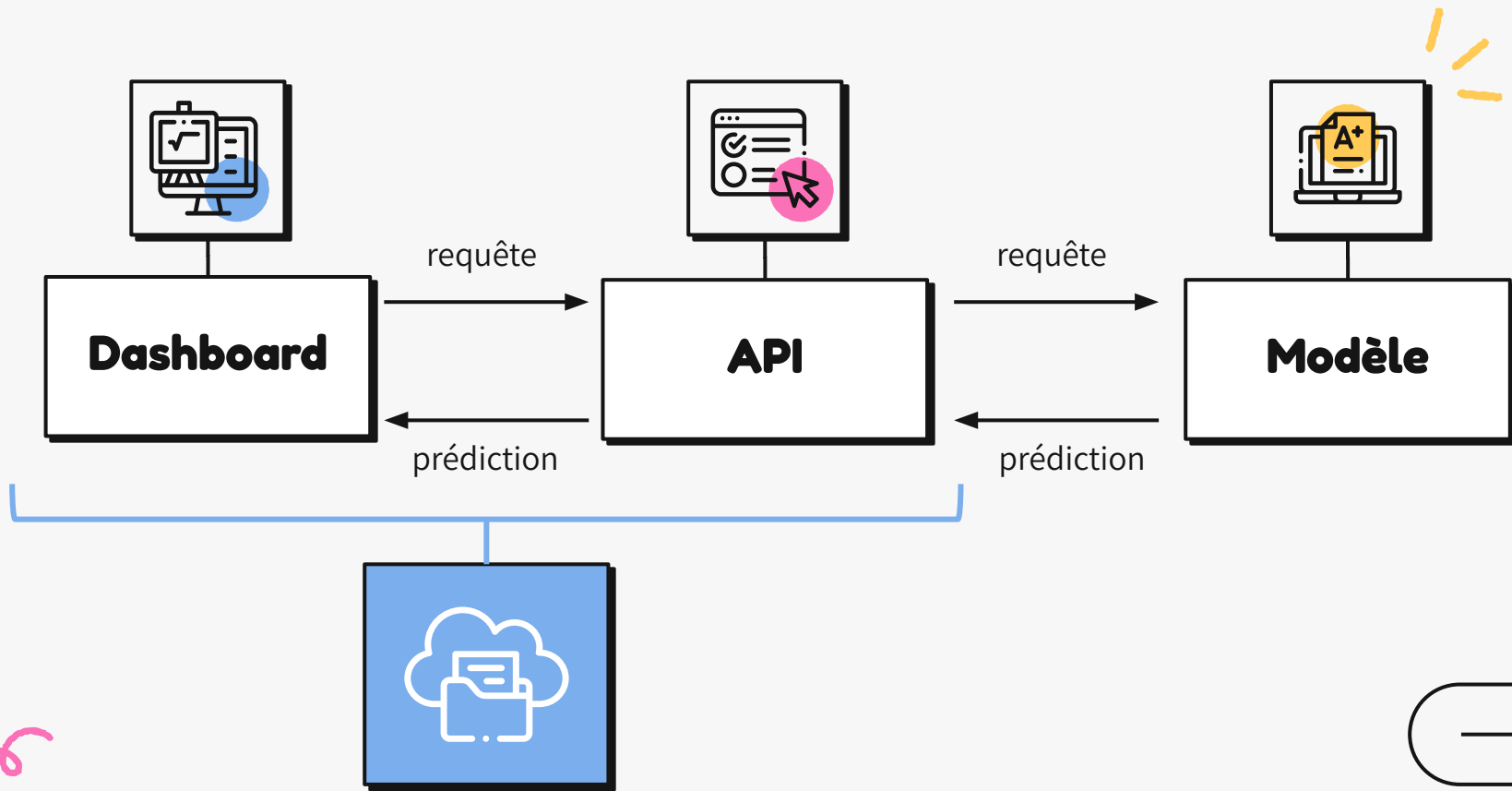


02

# Pipeline de déploiement continu



# Mise en production



# Contrôle de versions



- Code (API, Dashboard)
- Historique (versions, modifications)

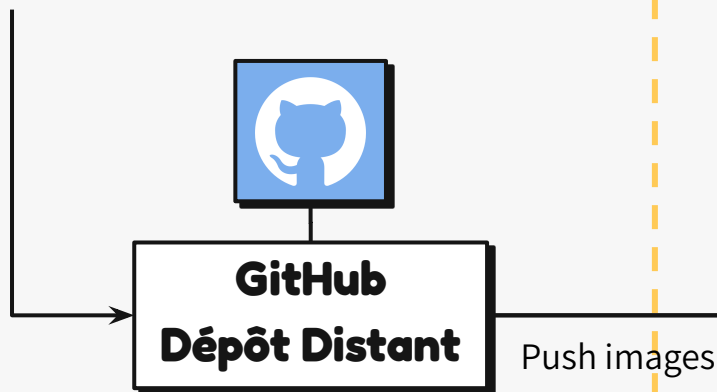
- Code (API, Dashboard)
- Historique (versions, modifications)
- Fichier Workflow



# Déploiement continu

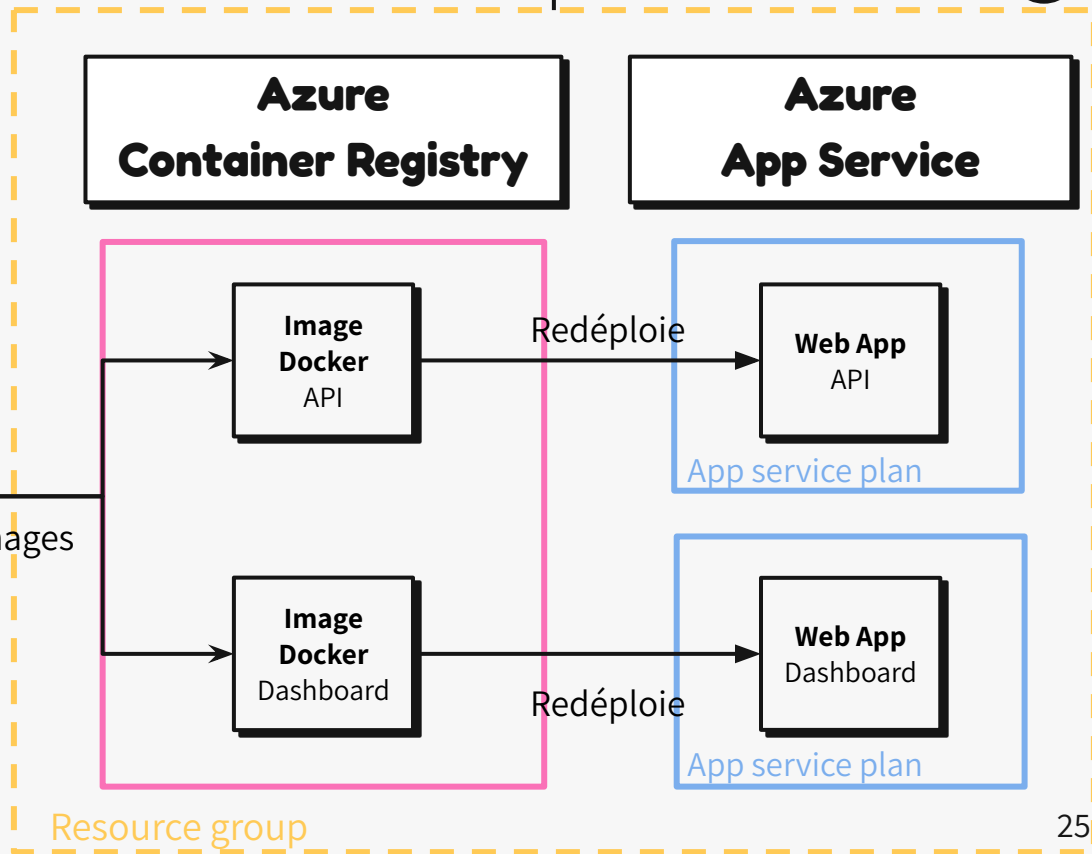


Modifications  
branche "main"



Workflow GitHub Actions :

- Tests unitaires
- Build images docker



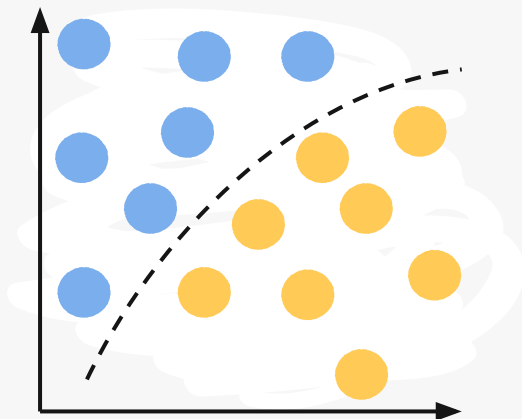
03

# Analyse du **Data Drift**

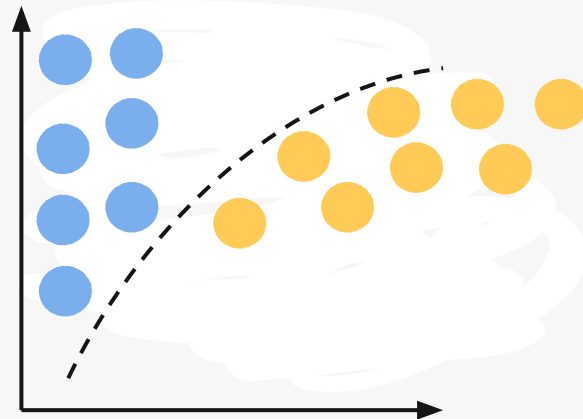


# Data Drift : définition

Entraînement

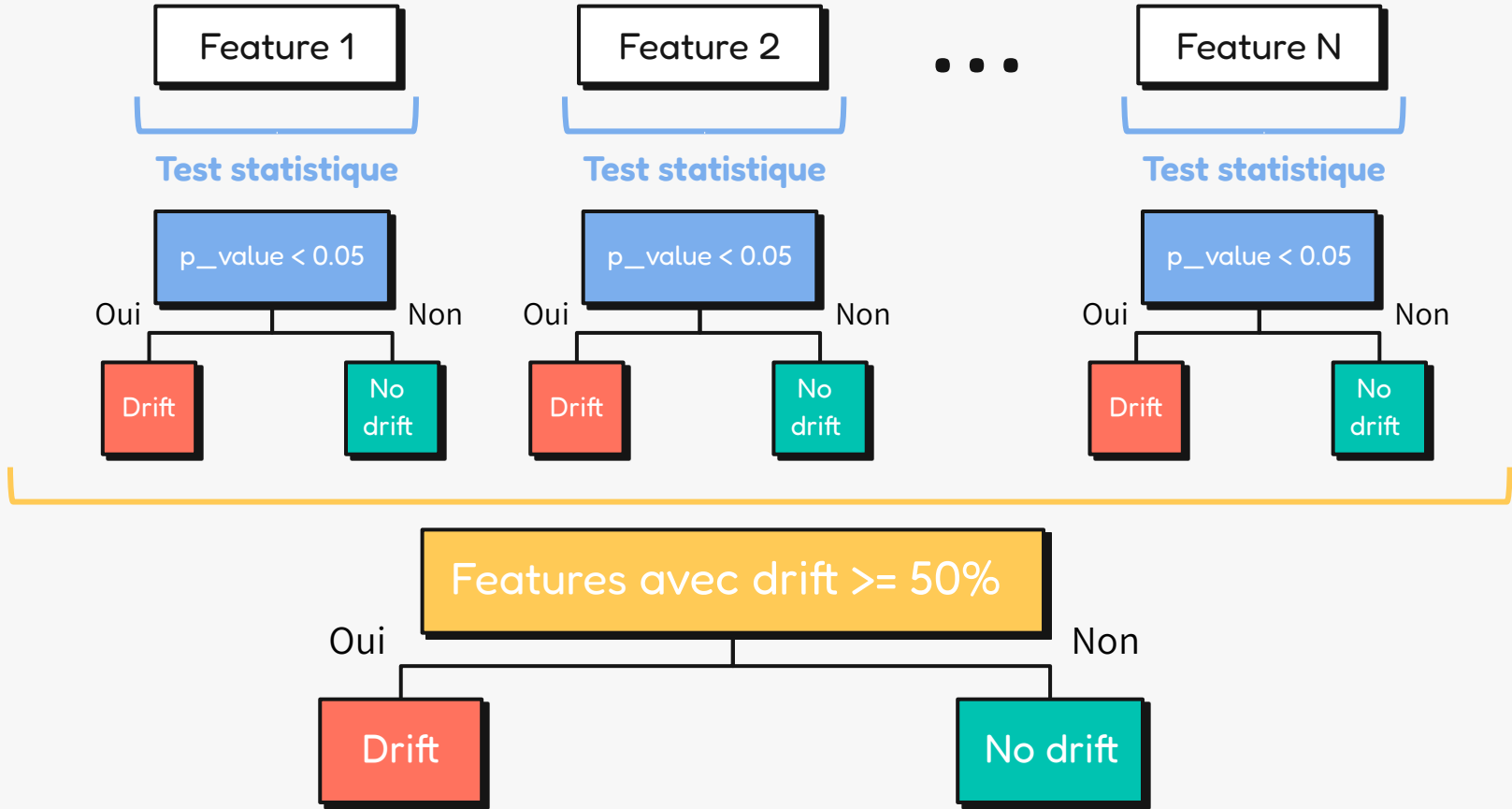


Production



 RISQUES : Prédictiones erronées => pertes

# Algorithme Evidently





En-tête rapport evidently

## Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

458

Columns

62

Drifted Columns

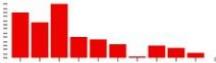
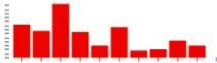
0.135

Share of Drifted Columns



## Data Drift Summary

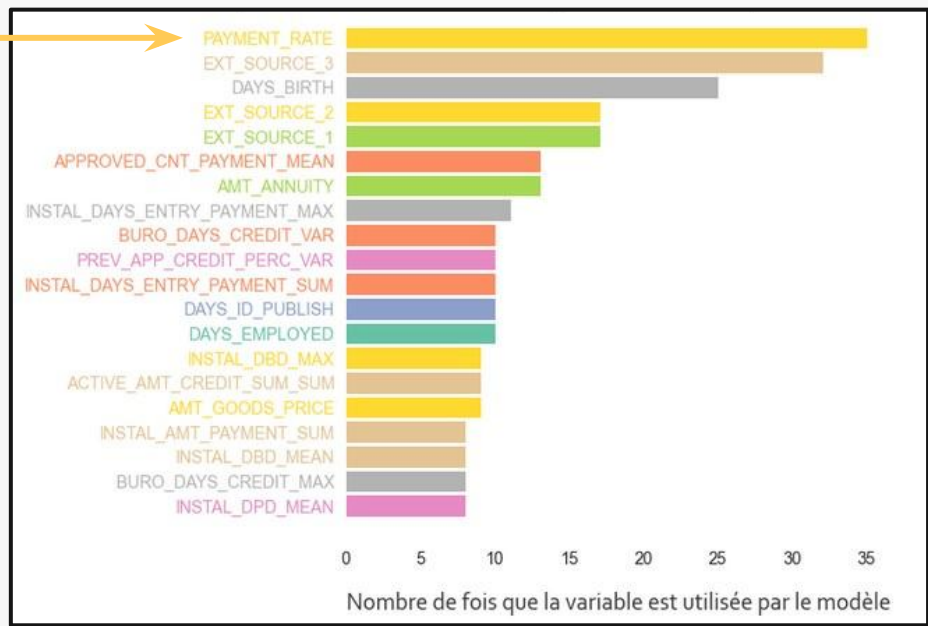
Drift is detected for 13.537% of columns (62 out of 458).

Q PAYMENT\_RATE X

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
> PAYMENT_RATE	num			Detected	K-S p_value	0

# Résultats

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
> PAYMENT_RATE	num			Detected	K-S p_value	0



Top des 20 variables les plus utilisées par le modèle lgbm

**04**

# Démonstration Dashboard



# Conclusion : Améliorations possibles



Travailler sur le jeu de données entier

Réduire le nombre de variables

Renommer les variables

Personnaliser le pré-traitement des données

Intégrer la détection de data drift au pipeline





# Annexes

# Plan des annexes



**01**

## **Critères d'accessibilité WCAG**

Critères pris en compte dans le dashboard

**02**

## **Tests unitaires**

Captures d'écran

**03**

## **Exemples commandes Git**

Captures d'écran

**04**

## **Dépôt sur GitHub**

Captures d'écran

**05**

## **MLflow Tracking**

Captures d'écran



# Dashboard - Critères d'accessibilité WCAG :

Critère	Description	Mise en application
1.4.3	Rapport de contraste d'au moins 4,5:1	Choix d'un thème qui vérifie ce critère (Streamlit)
1.4.1	La couleur n'est pas utilisée comme la seule façon de véhiculer de l'information	Utilisation de formes différentes dans le graphique pour distinguer le client sélectionné et les autres clients
1.1.1	Tout contenu non textuel présenté à l'utilisateur a un équivalent textuel	Graphiques légendés Widgets (composants d'interface ou de saisie) ont tous des titres qui décrivent leur fonction
1.4.4	Le texte peut être redimensionné jusqu' à 200% sans l'aide d'une technologie d'assistance	Ne semble pas possible avec Streamlit
2.4.2	Les pages Web présentent un titre qui décrit leur sujet ou leur but	Page principale et onglets titrés

# Critère 1.4.3 :

Thème du dashboard

## WCAG contrast ratio

Check if the color contrasts of the selected colors are enough to the WCAG guidelines recommendation.  
For the details about it, see some resources such as the [WCAG document](#) or the [MDN page](#).

Primary color



Background color



5.23 : 1



WCAG AA

Lorem ipsum

Secondary background color



4.67 : 1



WCAG AA

Lorem ipsum

Text color



12.53 : 1



WCAG AAA

Lorem ipsum

11.18 : 1



WCAG AAA

Lorem ipsum

# Tests Unitaires - Configuration

02



Objectif : tester **rapidement** une unité de code **indépendamment** des autres fonctionnalités

```
import pytest
import responses
from fastapi.testclient import TestClient
```

```
@pytest.fixture(scope='session')
def monkey_session():
    with pytest.MonkeyPatch.context() as mp:
        yield mp
```

```
@pytest.fixture(scope='session')
def client(monkey_session):
    monkey_session.setenv('DATA_PATH', 'api/data')
    monkey_session.setenv('MODEL_PATH', 'api/models')
```

```
from api.app.main import App
app_test = App('AppTest', 'App to test functions', '0.1').create_app()

with TestClient(app_test) as client:
    yield client
```

```
@pytest.fixture
def mocked_responses():
    with responses.RequestsMock(assert_all_requests_are_fired=True) as rsps:
        yield rsps
```

On simule le comportement (mock) de :

- 1 Constantes
- 2 Requêtes HTTP (grâce à un client de test)
- 3 Réponse d'une requête HTTP

3

# Tests Unitaires - API

## Librairie Pytest



On teste :

- ① Le code de réponse
- ② Les données de la réponse

```
import os
CLIENT_ID_TEST = 386902
```

```
def test_check_client_in_database_status_code_ok(client):
    response = client.get('/in_database', params={'client_id': CLIENT_ID_TEST})
    assert response.status_code == 200
    assert 'check' in response.json() ①
```

```
def test_check_client_in_database_should_return_true(client):
    response = client.get('/in_database', params={'client_id': CLIENT_ID_TEST})
    assert response.json() == {"check": True} ②
```

```
def test_check_client_in_database_should_return_false(client):
    response = client.get('/in_database', params={'client_id': 0})
    assert response.json() == {"check": False} ②
```

# Tests Unitaires - Dashboard

## Librairie Pytest



```
class TestDashboard:
    @classmethod
    def setup_class(cls):
        mp.setenv('URL_API', URL_API_TEST)
        from dashboard import functions
        mp.setattr(functions, 'store_request', cls.mock_action)

    @classmethod
    def teardown_class(cls):
        mp.delenv('URL_API')
        if os.path.isfile('database.pkl'):
            os.remove('database.pkl')

    @staticmethod
    def mock_action(response, response_time, params, endpoint, result=np.nan):
        request_log = pd.DataFrame([{'time': response_time, 'params': params, 'endpoint': endpoint,
                                     'status': response.status_code, 'result': result}])
        pd.concat([pd.DataFrame(), request_log])

    def test_get_all_client_ids(self, mocked_responses):
        mocked_responses.get(URL_API_TEST + "client_ids",
                             json={"ids": [386902, 215797]},
                             status=200,
                             content_type="application/json")

        from dashboard import functions
        resp = functions.get_all_client_ids()
        assert resp == [386902, 215797]
```

1

2

On imite :

- 1 Le comportement d'une fonction externe

On teste :

- 2 Que notre fonction renvoie bien la réponse reçue

# Couverture de test



Objectif : connaître le **pourcentage de lignes de notre code qui ont été testées.**

Coverage report: 59%

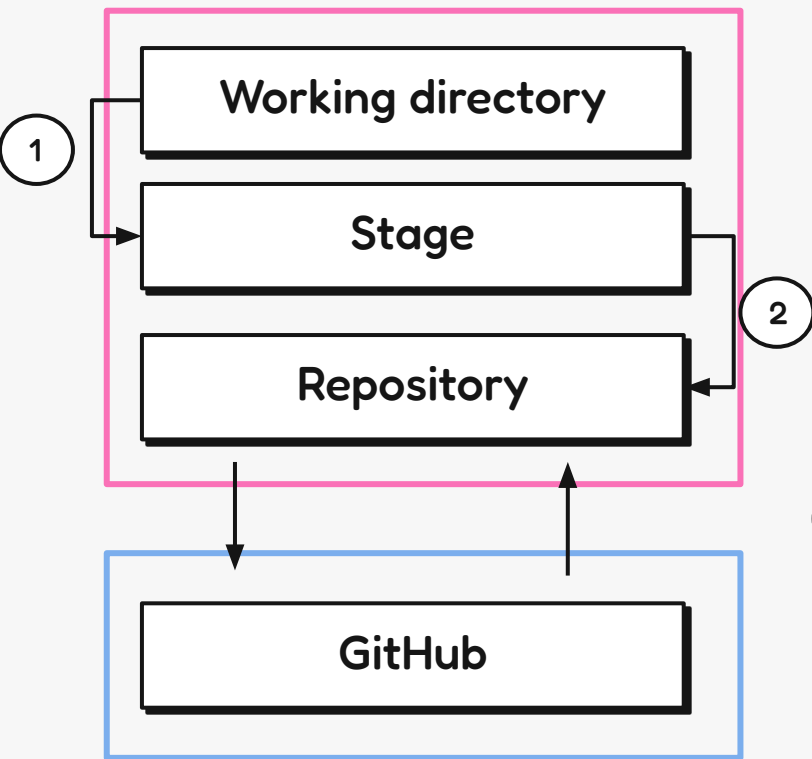
*coverage.py v7.2.7, created at 2023-07-06 01:24 +0200*

Module	statements	missing	excluded	coverage
api\app\main.py	168	38	0	77%
dashboard\functions.py	242	132	0	45%
<b>Total</b>	<b>410</b>	<b>170</b>	<b>0</b>	<b>59%</b>



# Exemples commandes Git :

Dépôt local



1 **Git Add** : indexe le fichier

```
ataScientist/Projet 7/projet7api_dash/oc-projet7 (test)
$ git add note_technique/NoteTechnique.pdf
```

**Git Status** : vérifie que le fichier est bien indexé

```
ataScientist/Projet 7/projet7api_dash/oc-projet7 (test)
$ git status
On branch test
Your branch is up to date with 'origin/test'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   note_technique/NoteTechnique.pdf
```

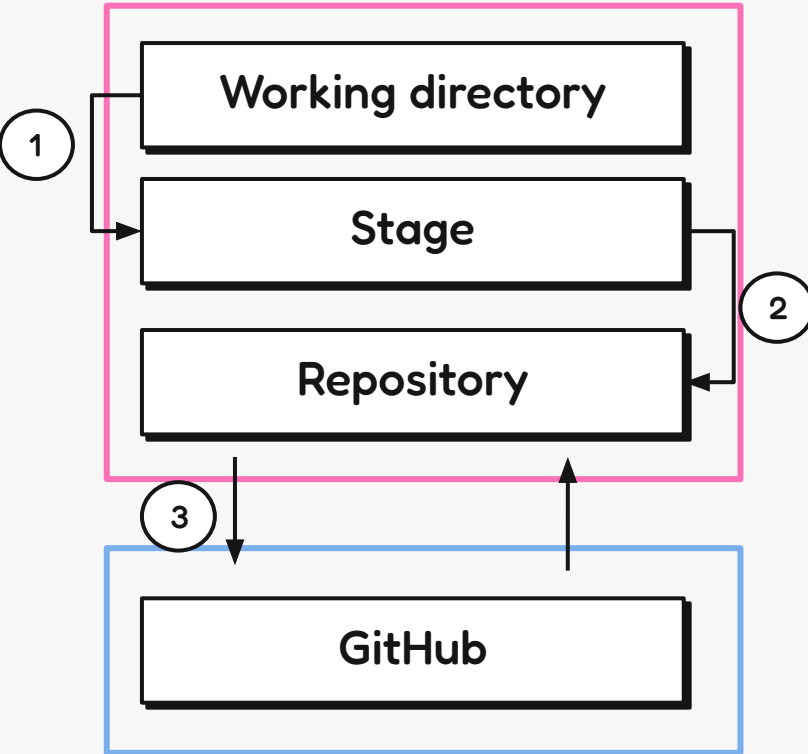
2 **Git Commit** : crée nouvelle version du projet

```
ataScientist/Projet 7/projet7api_dash/oc-projet7 (test)
$ git commit -m 'ajout de la note technique'
[test de3ff35] ajout de la note technique
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 note_technique/NoteTechnique.pdf
```

Dépôt Distant

# Exemples commandes Git :

Dépôt local



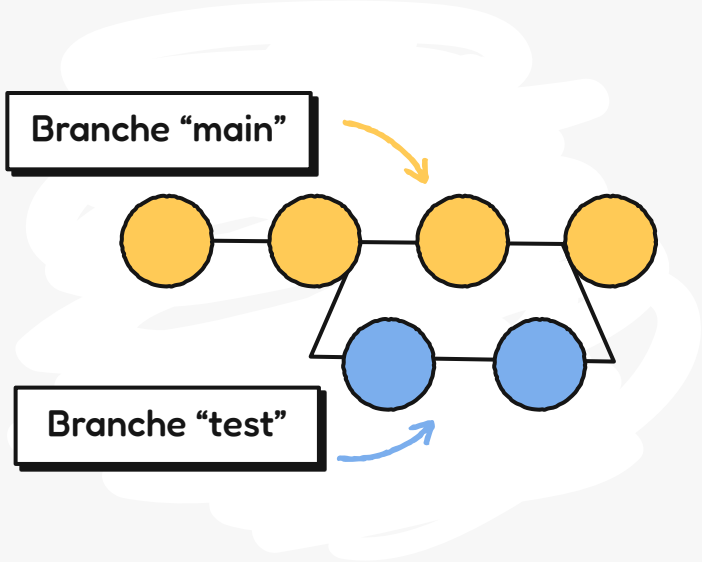
3

**Git Push** : envoie commits vers dépôt distant

```
ataScientist/Projet 7/projet7api_dash/oc-projet7 (test) /OpenClassRooms/Format
$ git push -u origin test
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 889.72 KiB | 12.71 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/   /oc-projet7.git
   d044ac1..de3ff35  test -> test
branch 'test' set up to track 'origin/test'.
```

Dépôt Distant

# Exemples commandes Git :



**Git Checkout:** se place sur la branche principale

```

ataScientist/Projet 7/projet7api_dash/oc-projet7 (test)
$ git checkout main
Updating files: 100% (33/33), done.
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
  
```

**Git Branch:** vérifie qu'on est bien sur la branche principale

```

ataScientist/Projet 7/projet7api_dash/oc-projet7 (main)
$ git branch
* main
  docker
  notebook
  reload
  test
  
```

**Git Merge:** fusionne branche test avec branche principale

```

ataScientist/Projet 7/projet7api_dash/oc-projet7 (main)
$ git merge test
Updating bcd5e1e..de3ff35
Updating files: 100% (33/33), done.
Fast-forward
  
```

# GitHub Actions - Détails Build

**1**

Require approval from specific reviewers before merging  
Branch protection rules ensure specific people approve pull requests before they're merged. [Add rule](#) [×](#)

**2**

This branch has no conflicts with the base branch  
Merging can be performed automatically.

[Merge pull request](#) You can also open this in [GitHub Desktop](#) or view [command line instructions](#).

Projects Security Insights Settings

All workflows  
Showing runs from all workflows

67 workflow runs

Event Status Branch Actor

**3**

**Merge pull request #5 from J28u/test**  
Build and push docker image to Azure Container Registry - ocpojet7  
#25: Commit 622f4a6 pushed by [main](#) [now](#)  
[In progress](#) [...](#)

**build**  
succeeded 20 hours ago in 7m 58s

Search logs

- Set up job 3s
- Run actions/checkout@v2 22s
- Checkout LFS objects 0s
- Set up Python 9s
- Install dependencies 59s
- Test with pytest** 42s
- Set up Docker Buildx 2s
- Log in to registry 2s
- Build and push container api image to registry 2m 49s
- Build and push container dashboard image to registry 2m 40s
- Post Build and push container dashboard image to registry 0s
- Post Build and push container api image to registry 0s



**Test with pytest**

42s

40

41 ===== 27 passed, 1016 warnings in 41.03s =====

# MLflow Tracking - Table View

mlflow 2.3.2

Experiments

Models

GitHub

Docs

## Experiments



Search Experiments



Default



Default



Provide Feedback



Share

Experiment ID:

0

Artifact Location:

file:/

Notebook/mlruns/0

> Description

Edit

Table view

Chart view

Q metrics.rmse < 1 and params.model = "tree"



Sort: custom\_score\_test



Refresh

Columns

Time created: All time

State: Active

<input type="checkbox"/>		Run Name	Created	Duration	Source	Models
<input type="checkbox"/>		redolent-mare-265	1 month ago	5.2min	C:\Users\...	sklearn
<input type="checkbox"/>		painted-bear-335	1 month ago	17.4min	C:\Users\...	sklearn
<input type="checkbox"/>		rebellious-hawk-51	1 month ago	9.5min	C:\Users\...	sklearn

# MLflow Tracking - Chart View

Default [Provide Feedback](#)

[Share](#)

Experiment ID: 0 Artifact Location: file:/Notebook/mlruns/0

> Description [Edit](#)

Table view

Chart view

metrics.rmse < 1 and params.model = "tree"



Sort: custom\_score\_test



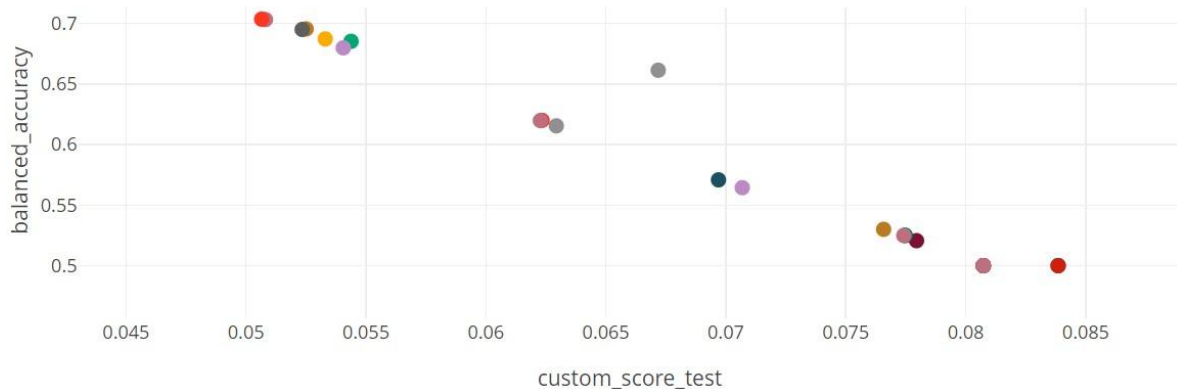
Refresh

Time created: All time

State: Active

Run Name
redolent-mare-265
painted-bear-335
rebellious-hawk-51
incongruous-crow-819
auspicious-crane-864
painted-midge-568

69 matching runs



# Thanks!



CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution

