Dans ce brief vous utiliserez pytest.

Intéressez-vous au mot clé "assert", quel est sa fonction ?

Qu'est-ce que le Test Driven Development ?

Testez les doctests

A quoi sert selon vous les test automatisés ? Donnez un exemple de scénario catastrophe qui arriverai si vous ne faites pas un des tests dans le cas de l'application fournie dans le drive (voir plus loin).

A partir de l'application flask basique fournie dans le drive de la promo (render\_app\_package.zip) :

Ecrivez les tests suivants dans un fichier test\_app.py séparé :

* Assurez vous que la forme de sortie du vectorizer est cohérente avec la forme d'entrée du classifieur (Logistic regression)
* Assurez-vous que la probabilité de la prédiction de la classe prédite est comprise entre 0 et 1 (en sortie de predict\_proba() du classifieur)
* Assurez-vous que la requête GET sur la page d'accueil (route "/") renvoie bien le code HTTP 200.
* Assurez-vous que le classifieur est bien une instance de le classe LogisticRegression (classe importée depuis sklearn.linear\_model)

Placez test\_app.py dans le dossier tests à la racine du projet.

-> Aide : Utilisez ce code suivant pour générer un client HTTP factice :

import sys

import os

# Rajout du dossier parent de ./tests dans le PYTHONPATH pour atteindre le fichier app.py

sys.path.append(os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..")))

import pytest

from app import app as flask\_app

@pytest.fixture

def client():

flask\_app.config["TESTING"] = True

with flask\_app.test\_client() as client:

yield client

(Ici "from app import app" correspond à l'objet "app" créé dans le fichier app.py)

Définissez :

* Tests unitaires
* Tests automatisés
* Tests fonctionnels
* Tests d'intégration
* Tests de non-régression

Par la suite intégrer les tests via un workflow de github actions en n'y insérant cette step (dans le ficher .yml) :

- name: Run tests

run: pytest -vv

env:

HOST\_MONGODB: dummy

MONGO\_DB\_APPNAME: dummy

PASSWORD\_MONGODB: dummy

USER\_MONGODB: dummy

Il est important de vérifier que les tous les tests sont bien passés, pour cela, utiliser la mention "needs" : <https://docs.github.com/fr/actions/reference/workflow-syntax-for-github-actions#jobsjob_idneeds>

Enfin, rédigez une doc (un README.md) dans le dossier /tests pour expliquer les tests que vous soumettez lors du versionnement, soyez le plus explicite possible.

Les GitHub actions qui vous seront utiles : <https://github.com/actions/checkout> <https://github.com/actions/setup-python> <https://github.com/johnbeynon/render-deploy-action>

La doc indispensable :

<https://docs.pytest.org/en/stable/>

<https://docs.github.com/fr/actions/how-tos/writing-workflows/building-and-testing/building-and-testing-python>

<https://docs.github.com/fr/actions/get-started/quickstart>

<https://openclassrooms.com/fr/courses/7155841-testez-votre-projet-python/7414151-codez-votre-premier-test-1>

<https://resources.github.com/learn/pathways/automation/essentials/application-testing-with-github-actions/>

<https://docs.github.com/fr/actions/reference/workflow-syntax-for-github-actions>

<https://circleci.com/blog/what-is-yaml-a-beginner-s-guide/>

L'excellent site github resources :

<https://resources.github.com/actions/what-is-github-actions/> <https://resources.github.com/learn/pathways/automation>

<https://resources.github.com/learn/pathways/automation/essentials/building-a-workflow-with-github-actions/>

La doc en plus : <https://www.honeybadger.io/blog/flask-github-actions-continuous-delivery/>

<https://medium.com/@ma11hewthomas/github-actions-publish-and-view-test-results-report-23bf4f3872ff>

<https://docs.pytest.org/en/6.2.x/fixture.html>

<https://openclassrooms.com/fr/courses/7155841-testez-votre-projet-python/7414196-utilisez-les-fixtures>

<https://www.cloudbees.com/blog/yaml-tutorial-everything-you-need-get-started>

Livrable : le fichier test\_app.py qui contient les fonctions de test + le fichier workflow main.yml modifié (présent dans le dossier de l'app niveau .github/worflow/main.yml) + le fichier README.md explicatif