RER – Monitoring & Testing

# Contexte :

Identifier et comparer des solutions de monitoring pour la garantie de la qualité et d'effectuer des tests unitaires. Intégrer une solution de solution de monitoring et testing d’un modèle de machine learning. Le monitoring est aussi utilisé pour corriger et détecter des éventuels dysfonctionnements.

# Problématique(s) :

Comment intégrer un outil de monitoring pour un modèle ?

Comment et pourquoi effectuer des tests unitaires ?

# Mots clés :

* Test unitaire : Un test unitaire est une procédure automatisée permettant de vérifier qu'une unité spécifique de code (généralement une fonction ou une méthode) fonctionne correctement. L'objectif est de s'assurer que chaque unité individuelle du logiciel fonctionne comme prévu.
* Fixture : En programmation et en tests, une fixture est un ensemble de données de configuration utilisées comme base pour effectuer des tests. Dans le contexte de tests unitaires, une fixture peut inclure la configuration initiale de l'environnement nécessaire à l'exécution des tests.
* Mocks : Un mock est un objet simulé qui remplace un objet réel pendant les tests. Les mocks sont utilisés pour simuler le comportement d'objets réels, généralement pour isoler l'unité testée du reste du système.
* Assertion : Une assertion est une déclaration ou une expression qui est supposée être vraie à un point spécifique d'un programme. Dans le contexte des tests, les assertions sont souvent utilisées pour vérifier si les résultats obtenus correspondent aux attentes.
* Unittest : unittest est un framework de test unitaire intégré à la bibliothèque standard de Python. Il fournit des fonctionnalités pour écrire et exécuter des tests de manière structurée.
* Pytest : pytest est un framework de test populaire pour Python. Il offre une syntaxe concise et expressive, la découverte automatique des tests, des fonctionnalités avancées telles que les fixtures, et une extensibilité grâce à des plugins.
* Monitoring : Le monitoring est le processus de surveillance constante des systèmes informatiques, des applications ou des services pour collecter des données sur leurs performances, leur disponibilité et leur comportement. Il vise à détecter les problèmes potentiels et à assurer un fonctionnement optimal.
* MLflow : MLflow est une plateforme open-source destinée à la gestion du cycle de vie complet des projets de machine learning. Elle prend en charge la gestion des expériences, le suivi des modèles, l'emballage et le déploiement des modèles.
* Arize : Arize est une plateforme de monitoring des modèles de machine learning. Elle se concentre sur la détection des dérives, la surveillance en temps réel et la génération d'alertes pour assurer la performance des modèles.
* Superwise.ai : Superwise.ai est une plateforme qui offre des fonctionnalités d'AutoML, de monitoring des modèles, d'interprétabilité et d'explication des modèles pour faciliter le développement et la gestion des modèles de machine learning.
* Collecteurs : Dans le contexte du monitoring, les collecteurs sont des composants qui rassemblent des données provenant de différentes sources, telles que des applications, des serveurs, des bases de données, etc., pour les rendre disponibles pour l'analyse.
* Déclencheurs : Les déclencheurs, ou "triggers" en anglais, sont des mécanismes qui activent des actions en réponse à des événements spécifiques. Dans le contexte du monitoring, un déclencheur pourrait, par exemple, déclencher une alerte lorsqu'une métrique dépasse une certaine limite.
* Intégration continue : L'intégration continue est une pratique de développement logiciel dans laquelle les modifications de code sont automatiquement intégrées dans le projet principal à chaque modification, suivi de l'exécution automatisée des tests pour s'assurer de la stabilité du projet.
* Déploiement continu : Le déploiement continu étend le concept de l'intégration continue en automatisant également le déploiement des versions validées du logiciel vers des environnements de production.
* Grafana : Grafana est une plateforme open-source de visualisation et de monitoring des métriques. Elle prend en charge la création de tableaux de bord interactifs pour afficher et analyser les données de performance.
* Métriques système : Les métriques système sont des mesures quantitatives des performances d'un système informatique. Elles incluent des informations telles que l'utilisation du CPU, la consommation de mémoire, le débit du réseau, etc.
* Métriques de performance : Dans le contexte du monitoring des applications, les métriques de performance mesurent divers aspects du comportement d'une application, tels que le temps de réponse, la charge de travail, les erreurs, etc. Ces métriques aident à évaluer les performances de l'application et à détecter les problèmes potentiels.

# Hypothèses :

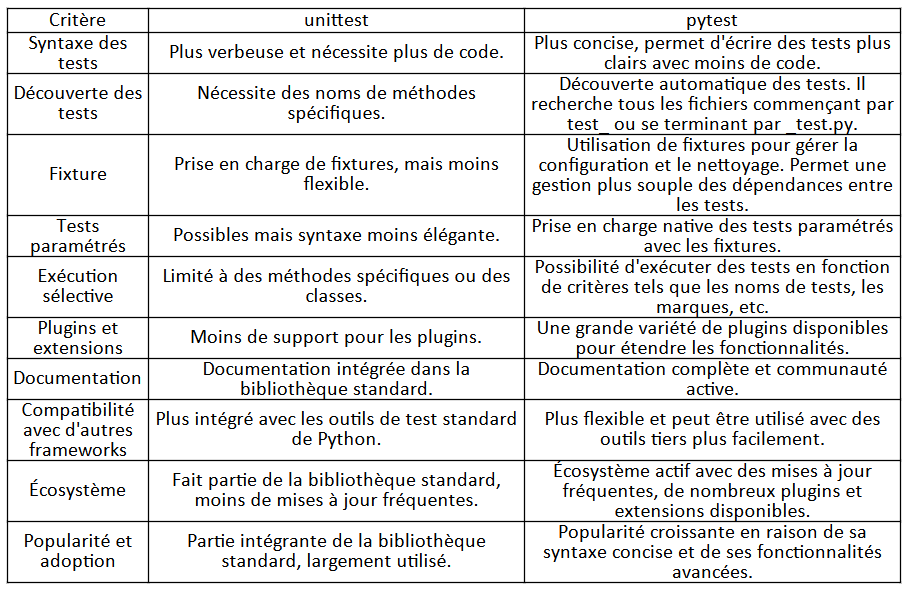
* Briand : Il n’y a pas d’ordre de l’exécution du testing et du monitoring.
* Briand II : Effectuer du testing reviens à observer la courbe d’évolution de nos hyperparamètres choisis sur différents cas de figure et sur la durée.
* Adrien : Il faut faire des tests sur toutes les fonctions de notre programme.
* Axel : Pytest est plus simple que Unittest pour faire des tests unitaires en python.

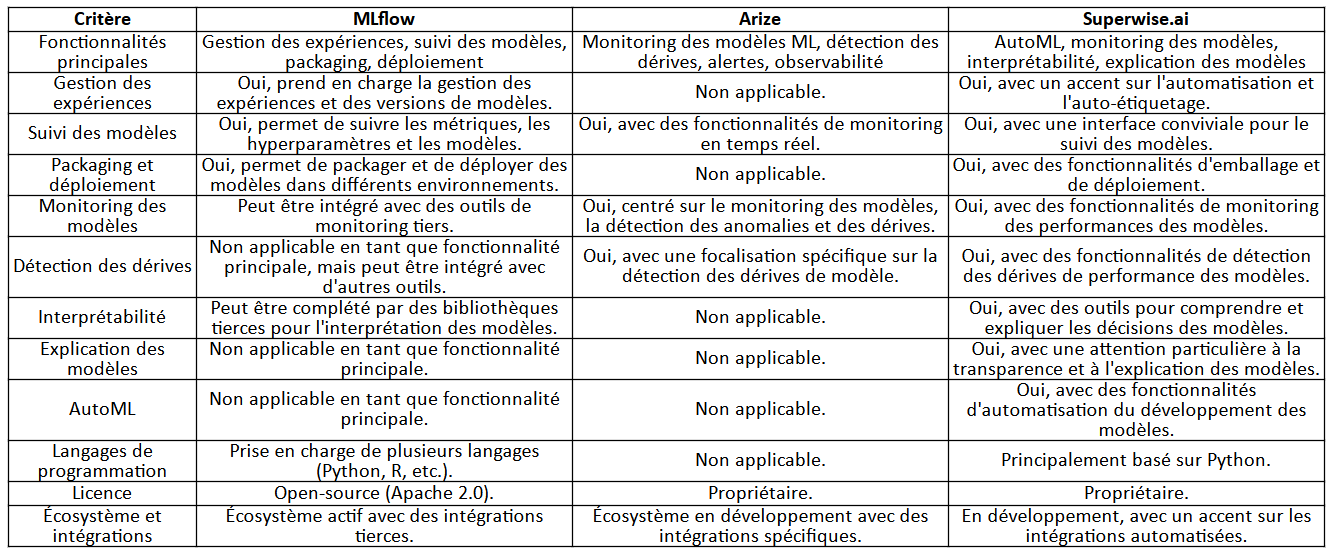
Vrai, la syntaxe de Pytest est plus concise et permet d’écrire des tests plus clair avec moins de code. De plus, Pytest est plus souvent mis à jour que Unittest.

* Aude : Les tests unitaires peuvent être intégrés au code grâce à Docstring.
* Tetyana : Il est obligatoire de faire du testing et du monitoring pour de meilleures performance du modèle.
* Jean-Paul : Contrairement au testing, le monitoring se fait en continue.
* Seydou : Est-ce que les dashboards vus précédemment sont les mêmes que ceux du monitoring.
* Étienne : On peut passer par plusieurs technologies pour réaliser du testing et du monitoring (réseau, console, fichiers de log).

# Plan d'action :

* Explorer les ressources
* Traiter les mots clés
* Répondre aux problématiques
* Réponde aux hypothèses
* Rendre les livrables : RER et workshops
* Comparaison pytest / unittest :



* Comparaison MLflow / Arize / superwise.ai :