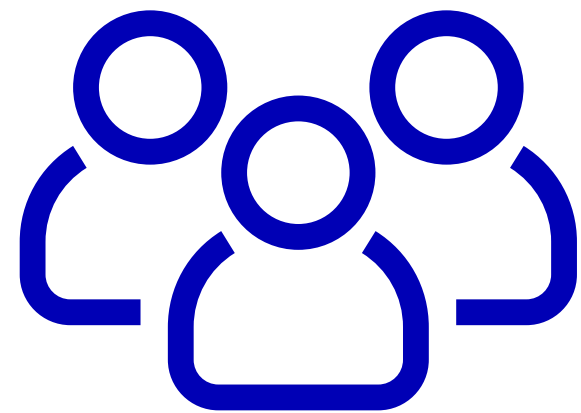


Monitoring d'une application ML avec Evidently AI, Prometheus & Grafana

Salsabil ZAGHDOUDI - DEV IA 2025

Auteur : Jérémy Vangansberg

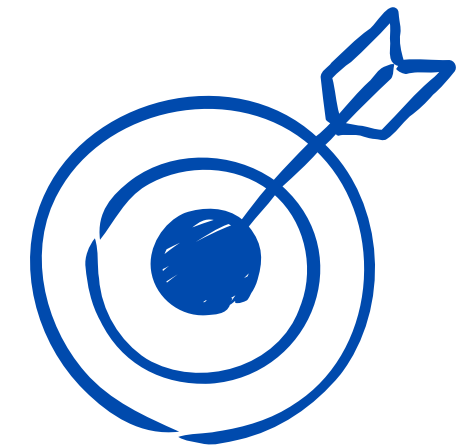
Contexte



Vous faites partie
de l'**équipe**
technique de
DataInsight Corp.



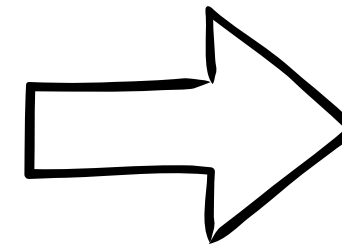
Mission : créer un
système complet
pour monitorer une
API de prédiction
ML.



Objectifs : analyser
la robustesse, les
performances et
les dérives du
modèle.

Pourquoi un monitoring ?

- En production, le suivi API + modèle est essentiel pour garantir la fiabilité et la stabilité



Assure un monitoring global de l'API pour garantir sa stabilité et sa disponibilité.

Surveiller :

- les **performances de l'API** (temps de réponse, requêtes, erreurs),
- l'**état de l'infrastructure** (CPU, RAM, conteneurs).
- la **création d'alertes en temps réel** pour détecter rapidement les anomalies.
- **Grafana fournit des tableaux de bord** visuels basés sur **les données collectées par Prometheus**,

Prometheus & Grafana

Prometheus et Grafana sont deux outils populaires utilisés dans le domaine de **la surveillance et de l'observabilité** des systèmes informatiques.

Prometheus

- Open source
- Collecte et stocke en temps réel des données sur l'état de performance d'un système
- Permet de suivre des métriques clés : usage du CPU, mémoire utilisée, nombre d'erreurs
- Aide à vérifier rapidement si le système fonctionne correctement

Grafana

- Outil de visualisation de données open source.
- Utilise les données collectées par Prometheus.
- Permet de créer des graphiques et des tableaux de bord pour visualiser et analyser les performances.

Grafana

Les requêtes

Catégorie	Signification	Exemples de codes	Interprétation
<u>2xx</u> — Succès	La requête s'est bien passée.	200 OK 201 <u>Created</u> 204 No Content	✅ L'API fonctionne normalement.
<u>4xx</u> — Erreur côté client	Le client a envoyé une mauvaise requête.	400 Bad Request (JSON incorrect) 401 <u>Unauthorized</u> (pas d'authentification) 404 Not Found (<u>endpoint</u> inconnu) 422 <u>Unprocessable Entity</u> (données invalides)	⚠️ L'API fonctionne, mais la requête envoyée est incorrecte.
<u>5xx</u> — Erreur serveur	L'API a planté ou a une erreur interne.	500 Internal Server Error 502 Bad <u>Gateway</u> 503 Service <u>Unavailable</u>	❌ Problème sérieux : bug serveur, crash ou panne.

Evidently AI

Evidently AI offre un suivi complet des modèles de machine learning en production. Il détecte différents types de dérives et problèmes liés aux données et aux prédictions :

Data drift :

changement des données d'entrée par rapport au jeu d'entraînement.

Concept drift :

évolution du lien entre les features et la cible (ex. le comportement utilisateur change).

Target Drift :

modification de la distribution de la cible en production.

Evidently AI

Performance

Monitoring : suivi de métriques comme précision ou F1-score pour vérifier si le modèle reste performant.

Data Quality

Monitoring : détection de valeurs manquantes, aberrantes ou incohérentes.

Explainability

Reports : rapports expliquant les contributions des features aux prédictions.

Comment utiliser Evidently AI ?

- installer Evidently :

```
bash
```

```
pip install evidently
```

- générer un rapport simple avec quelques lignes de code :

```
python
```

```
from evidently.report import Report
```

```
from evidently.metrics import DataDriftTable
```

```
report = Report(metrics=[DataDriftTable()])
```

```
report.run(reference_data=df_ref, current_data=df_prod)
```

```
report.save_html("data_drift_report.html")
```


Comment installer Evidently AI ?

- GitHub
- Community

Get Started

What is Evidently?

Quickstart >

Setup

Installation

Evidently Cloud

Self-hosting

Evaluation library

Introduction

Run an evaluation >

Optimize prompts >

Generate data >

Setup

Installation

How to install the open-source Python library.

Evidently

Evidently is available as a Python package. Install it using the **pip package manager**:

```
pip install evidently
```

To install **evidently** using **conda installer**, run:

```
conda install -c conda-forge evidently
```

Evidently LLM

To run evaluations specific to LLMs that include additional dependencies, run:

```
pip install evidently[llm]
```

Comment installer Evidently AI ?

Tracely

To use tracing based on OpenTelemetry, install the sister package **tracely**:

```
pip install tracely
```



Qu'est-ce qu'il faut connaître pour ce projet ?

Lecture d'un rapport Evidently

- comment lire un graphique de drift,
- interpréter un score "Drift detected : True/False",
- repérer quelles features posent problème

Comment l'intégrer au projet ?

- Evidently ne remplace pas Grafana/Prometheus : il sert au monitoring du modèle,
- Vous allez générer un rapport Evidently et le stocker dans un dossier,
- Vous allez exposer les métriques Evidently pour Grafana.

Le rôle : modèle = Evidently ; API + serveur = Prometheus/Grafana.

Ne pas confondre

Evidently AI

- S'occupe du modèle de ML et des données.
- Répond à : "Est-ce que mon modèle voit toujours les mêmes types de données ? Est-ce qu'il reste performant ?"
- Produit :
 - des rapports de dérive (data / target / concept),
 - des rapports de performance (F1, précision, etc.).

Evidently = santé du modèle et de ses données

Prometheus

- S'occupe de collecter des métriques chiffrées en continu :
- temps de réponse API,
- nombre de requêtes,
- erreurs 4xx / 5xx,
- CPU, RAM, etc.

Grafana

- Se connecte à Prometheus (et éventuellement à d'autres sources).
- Sert à afficher ces métriques :
- graphiques, jauges, tableaux, dashboards temps réel.

Dataset

Pour analyser les dérives et surveiller les performances, nous utiliserons deux jeux de données :

- **Jeu de référence** : Issu du dataset d'entraînement, il servira de base pour détecter les dérives dans les données d'entrée.
- **Jeu de test** : Utilisé pour évaluer les performances initiales du modèle et établir des benchmarks.

Ces jeux de données seront essentiels pour configurer les analyses dans Evidently AI.

Vous pouvez utiliser un dataset standard comme **les iris ou titanic**.

Modalités pédagogiques

Travail en groupe : Vous réalisez le projet en groupe pour acquérir une maîtrise professionnelle complète sur l'intégration et le monitoring des applications ML.

Préparation professionnelle : Ce projet reflète une situation concrète en entreprise et peut servir comme référence pour des certifications ou des entretiens techniques.

Groupe de 3 personnes:
1 Chef de projet
1 Développeur API / Back-end
1 Ingénieur Data / Monitoring

Modalités d'évaluation

Inspection du code

Votre code sera évalué sur la clarté, la modularité et le respect des bonnes pratiques.

Livrable sur GitHub

Le dépôt doit inclure :

- Le code source de l'API et du modèle.
- Les configurations pour Prometheus, Grafana, et Evidently AI.
- Un fichier Docker Compose orchestrant tous les composants.
- Une documentation complète (README) expliquant comment exécuter et utiliser le projet.

Livrables

Un dépôt GitHub contenant :

- Une API FastAPI opérationnelle pour les prédictions.
- Un tableau de bord Grafana fonctionnel pour visualiser les métriques collectées par Prometheus.
- Des rapports Evidently couvrant les dérives (data, concept, target) et les performances.
- Un fichier docker-compose.yml orchestrant l'ensemble des services.
- Une documentation claire expliquant le fonctionnement de chaque composant et les étapes pour reproduire le projet.

Étapes détaillées du projet

1. Entraînement du modèle :

- Préparation des données (division en jeux d'entraînement, test et référence).
- Entraînement d'un modèle (par exemple, RandomForestClassifier).
- Exportation du modèle pour utilisation dans l'API.

2. Développement de l'API FastAPI :

- Création d'un endpoint /predict.
- Enregistrement des données reçues pour surveiller les distributions et les prédictions.

Étapes détaillées du projet

3. Mise en place de Prometheus et Grafana :

- Ajout de `prometheus_fastapi_instrumentator` à l'API pour collecter des métriques.
- Configuration de Prometheus pour collecter les métriques :
 - Latence, volume de requêtes, taux d'erreurs.
- Création d'un tableau de bord Grafana pour visualiser les métriques.

4. Monitoring des dérives et

performances avec Evidently AI :

- Analyse des dérives (data, concept, target) à l'aide du jeu de référence.
- Suivi des performances (précision, F1-score).
- Génération de rapports visuels.

Étapes détaillées du projet

5. Orchestration avec Docker Compose :

- Conteneurisation de tous les composants (API, Prometheus, Grafana, Evidently).
- Création d'un fichier docker-compose.yml pour l'orchestration.

Ressources

- [FastAPI Documentation](#)
- [Prometheus Docs](#)
- [Grafana Tutorials](#)
- [Evidently AI](#)
- <https://blog.zwindler.fr/2020/04/13/decouvrir-prometheus-et-grafana-par-lexemple/>