

Déploiement du modèle de Machine Learning

Bouzouita Hayette - Avril 2026



Sommaire

- Introduction
- Structure du dépôt Git
- Fonctionnalité de l'API
- Tests unitaires et fonctionnels
- Base de données
- Cas d'usage du modèle

Introduction

- **Projet** : Employee Attrition API
- **Contexte** : Déploiement d'un modèle de Machine Learning sur Hugging Face
- **Objectif du modèle** : Prédire le risque de départ d'un employé
- **Technologies principales** : FastAPI - PostgreSQL - Pytest - GitHub Actions

Dépôt Git : Structure

- Code source organisé par modules
- **app** : API, services, base de données, configuration
- **tests** : tests unitaires et fonctionnels
- **scripts** : création et alimentation de la base
- **.github/workflows** : pipeline CI/CD
- Autres : artifacts, data, docs
- Fichiers clés : requirements, .gitignore, README

Pipeline CI/CD

CI (ci.yml)

- déclenché sur push / pull request
- installe Python et uv
- synchronise les dépendances
- exécute les tests

CD (deploy.yml)

- déclenché sur push sur main
- prépare un snapshot propre
- utilise le secret HF_TOKEN
- déploie vers Hugging Face Spaces

API

Fonctionnalité de l'API

- API de prédiction du risque d'attrition des employés
- Documentation intégrée (OpenAPI/ Swagger)
- Validation automatique des entrées (FastAPI / Pydantic)
- Stockage de l'historique des prédictions

Endpoints

- **GET /** : point d'entrée de l'API
- **GET /health** : vérification du bon fonctionnement
- **POST /predict/** : prédiction à partir de données saisies
- **GET /predict/predictions** : consultation de l'historique
- **POST /predict/by-id/{employee_id}** : prédiction depuis un employé existant

Tests

Tests unitaires

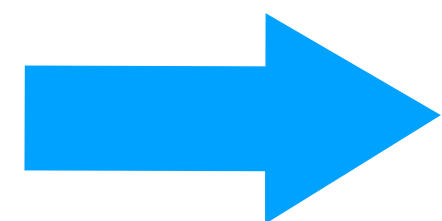


- test_prediction_service.py

Tests fonctionnels




- test_health
- test_predict



Couverture : 74%

Base de données

- Base locale principale : **PostgreSQL**
- 2 tables :
 - **employees** (stockage des données)
 - **predictions** (stockage de l'historique des prédictions -> traçabilité)
- Utilisation de **SQLAlchemy** comme ORM
- En déploiement Hugging Face : **SQLite**

predictions	
id 	int
input_data	json
prediction	int
probability	float
created_at	datetime

employees	
id 	int
age	float
revenu_mensuel	float
nombre_experiences_precedentes	int
annees_dans_entreprise	float
annees_dans_le_poste_actuel	float
satisfaction_employee_environnement	float
satisfaction_employee_nature_travail	float
satisfaction_employee_equipe	float
satisfaction_employee_equilibre_pro_perso	float
note_evaluation_actuelle	float
augmentation_salaire_precedente_pct	float
distance_domicile_travail	float
niveau_education	int
annees_depuis_la_derniere_promotion	float
annees_sous_responsable_actuel	float
a_suivi_formation	int
annees_experience_avant_entreprise	int
mobilite_interne	int
evolution_note	float
utilisation_pee	int

Cas d'usage

- Identifier les employés à risque de départ
- Amélioration de la rétention des talents
- Aide les RH à prendre des décisions (formation, augmentation, mobilité)
- Suivre l'évolution du risque dans le temps