

## Déploiement et cycle de vie d'un modèle de ML

Cursus concerné : MLOps

**Description détaillée :** L'objectif est de reprendre votre modèle de machine learning entraîné et d'implémenter tout le cycle de vie du modèle en production. Implémenter l'architecture et appliquer tous les principes MLOps.

SPRINT	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation du projet
2	Cahier des charges	Définition des besoins métiers et du modèle utilisé:  - Choix du sujet et du modèle - Définition des métriques et exigences de performances - Schéma d'implémentation - Récupération de nouvelles données	Documentation externe Parcours DS	Format PDF  Artéfact du modèle sauvegardé
	ML et Version Control	Instancier un dépôt Git : l'organiser et le documenter ML sur Github :  - Récupérer la modélisation finale du côté DS - Implémenter des tests unitaires - Appliquer la métrique d'évaluation - Procédure d'utilisation et installation	Git/Github Tests unitaires	Dépôt Github :  - Organisation - Commit - Tests

## <u>DataScientest.com</u>



	API	Créer une API du modèle. Cette API devra gérer :  - Prédictions du modèle - l'authentification des utilisateurs une ou des techniques de votre choix Techniques possibles : Basic HTTP, JWT Token, clé API, OAuth	FastAPI Flask Sécurité des API	API sur le dépôt Github
	Base de données	Créer une base de données en backend pour l'API :  - Gestion et authentification des utilisateurs Stockage des prédictions - Cette base de données SQL devra être isolée Ingestion de nouvelles données et "live data" - Choix de la BDD justifiée : MySQL - FlaskSQL - MongoDB - HBase	SQL noSQL	Base de données accessible par l'API
3	Isolation	Conteneuriser cette API avec Docker. Isoler la partie l'API et BDD	Docker	docker-compose avec l'installation sur le GIT
	Testing et Monitoring	Monitorer de façon périodique l'évolution du projet :  - Assurer le maintien des performances grâce à la surveillance des métriques d'alerte  - Tracker les paramètres et alerter des déviations des performances  - Fine-tuning des performances en cas d'alerte  - Gestion "data drift"  - réentrainement périodique du modèle	MLFlow OU Airflow	MLflow user interface OU Airflow user interface

## <u>DataScientest.com</u>



4 CI/CD + 1 optionnel au choix	CI / CD	Créer une pipeline CI/CD pour gérer l'intégration et le déploiement de votre application.  - Gestion des versions du modèle - Intégration automatisée vers la production - Exécution des tests	Git / Github	GitHub Action
	IHM (optionnel)	Créer une interface WEB Front pour l'API. Cette interface interactive permettra aux utilisateurs d'utiliser l'outil de manière intuitive	Bootstrap HTML CSS	Interface Web
	Cloud (optionnel)	Déployer cette solution clé en main sur le Cloud. Avec un lien publique pour se connecter	AWS	Lien de connexion sur GitHub
	HTTPS (optionnel)	Utiliser Kubernetes pour l'orchestration des conteneurs et Traefik pour l'implémentation SSL	Kubernetes Traefik	Lien HTTPS

Support de soutenances: Slides de présentation + Démo de la solution

## <u>DataScientest.com</u>