Industrialisation avancée des projets en Data Science

Programme *(Mis à jour le 21/08/2026)*

Structuration, automatisation et CI/CD

- Pourquoi industrialiser? (reproductibilité, fiabilité, montée en charge, maintenance)
- Cycle de vie d'un projet Data Science (POC ? production)
- La méthodologie CRISP-DM (Cross-Industry Standard Process for Data Mining)
- Séparation des responsabilités : ingestion des données, feature engineering, entraînement, évaluation, inférence
- Organisation type d'un projet Python sous la forme d'un package
- Gestion de la configuration
- Gestion des dépendances (pip, poetry, conda/mamba, pixi)
- Refactorisation d'un notebook ML en package Python
- Paramétrisation du pipeline d'entraînement

Automatiser l'intégration et le déploiement (CI/CD)

- Automatiser les tests et les builds
- Préparer un projet Data Science à être déployé en continu
- Introduction aux principes CI/CD appliqués à la Data Science
- Spécificités des workflows de machine learning par rapport au développement applicatif classique
- Pipelines CI : linting, tests, build d'artefacts
- Versionnement du code et des modèles
- Notions de reproductibilité et traçabilité
- Mise en place d'un pipeline CI simple : installation des dépendances, exécution des tests, build du package

Architecture d'un projet Data Science

- Concevoir une architecture robuste et évolutive.
- Comprendre les interactions entre composants
- Séparation des briques:
  – Architecture logique: batch vs temps réel, entraînement vs inférence.
  – Architecture données : sources, stockage et flux
  – Architecture applicative : services/APIs, jobs planifiés, monitoring

Orchestration des pipelines avec Apache Airflow ou Prefect/ray

- Orchestrer des pipelines de données et de ML.
- Gérer dépendances, erreurs et reprises.
- Concepts fondamentaux de l'orchestrateur (Airflow ou Prefect/ray)
  – Tâches, pipeline/flow et planification.
- Bonnes pratiques
- Gestion des paramètres et des environnements
- Orchestration des différentes phases : l'ingestion, entraînement, évaluation et déploiement du modèle
- Création d'un pipeline ML complet intégrant la gestion et reprise sur erreurs

Tests & Qualité des projets ML

- Garantir la fiabilité des pipelines et des modèles
- Comprendre les limites du testing classique en ML
- Tests unitaires adaptés à la datascience

Référence

THIP3698

Durée

3 jours / 21 heures

Prix HT / stagiaire

3750€

Objectifs pédagogiques

- Comprendre et organiser un projet de Machine Learning en différentes briques fonctionnelles
- Concevoir l'architecture applicative d'un projet Data Science.
- Déployer et exploiter une application de Machine Learning en production.
- Alimenter une application de Machine Learning avec des flux de données en temps réel.
- Évaluer et mesurer les performances de leur application de Machine Learning.

Niveau requis

- Connaissances de base en programmation et en scripting avec le langage Python.
- Pratique régulière des librairies de datascience numpy, pandas et matplotlib.
- Pratique régulière des librairies de machine learning scikit-learn et pour les réseaux de neurones TensorFlow ou PyTorch

Public concerné

- Développeurs d'applications de Data Science

Formateur

Les formateurs intervenants pour Themanis sont qualifiés par notre Responsable Technique Olivier Astre pour les formations informatiques et bureautiques et par Didier Payen pour les formations management.

Conditions d'accès à la formation

Délai : 3 mois à 1 semaine avant le démarrage de la formation dans la limite des effectifs indiqués

Moyens pédagogiques et techniques

Salles de formation (les personnes en situation de handicap peuvent avoir des besoins spécifiques pour suivre la formation. N'hésitez pas à nous contacter pour en discuter) équipée d'un ordinateur de dernière génération par stagiaire, réseau haut débit et vidéo-projection UHD
Documents supports de formation projetés
Apports théoriques, étude de cas concrets et exercices
Mise à disposition en ligne de documents supports à la suite de la formation

Dispositif de suivi de l'éxécution de l'évaluation des résultats de la formation

- Tests d’intégration de pipelines
- Tests fonctionnels sur l’inférence
- Validation des données : schémas, distributions, valeurs aberrantes
- Métriques de performance, biais, robustesse
- Outils : evidently, deepchecks…
- Écriture de tests unitaires et d’intégration et mise en place de validations simples sur les données

Déploiement & passage en production

- Exposer un modèle de Machine Learning
- Comprendre les contraintes de la production
- Différences entre batch et temps réel
- Exposition des modèles via API (REST)
- Conteneurisation avec Docker
- Notions de scalabilité et de résilience
- Introduction à Kubernetes
- Création d’une API d’inférence, conteneurisation de l’application ML. Exécution de l’application

Alimentation en données temps réel

- Intégrer des flux de données continus
- Différences batch / streaming.
- Introduction aux brokers de messages
- Cas d’usage du temps réel en ML
- Considérations de latence et de cohérence
- Réentraîner un modèle sur de nouvelles données

Monitoring, feedback et réentraînement

- Surveiller les performances en production.
- Mettre en place une boucle d’amélioration continue.
- Monitoring technique : latence, erreurs, disponibilité.
- Monitoring ML : dérive des données, dérive des performances, dérive conceptuelle.
- Collecte des feedbacks utilisateurs.
- Stratégies de réentraînement : périodique, déclenché par événement.
- Création d’un tableau de bord avec des indicateurs simples
- Détection basique de dérive
- Orchestration d’un réentraînement automatique
- Configurer un pipeline de réentraînement déclenché par une dérive de performance