PYTHON & DATA

Introduction au machine learning avec Python

Découvrir les fondements du machine learning avec scikit-learn. Régression, classification, clustering, évaluation de modèles : une introduction concrète et appliquée.

DURÉE	TARIF HT	NIVEAU	LANGUE	GROUPE	FORMAT
5j (35h)	~~1990.00 €~~ **1194.00 €**	**Avancé**	**FR**	**3–12**	**Formation**

1 PUBLIC VISÉ

- Développeurs Python maîtrisant les bases du langage souhaitant s'initier au machine learning de manière concrète et appliquée.
- Personnes ayant suivi les formations Python : les fondamentaux du langage et Python : programmation orientée objet et traitement de données ou disposant d'une expérience Python équivalente.
- Ingénieurs ou scientifiques souhaitant intégrer des modèles prédictifs dans leurs projets Python.
- Professionnels souhaitant passer de l'analyse manuelle de données à la modélisation prédictive avec scikit-learn.

2 PRÉREQUIS

- Bonne maîtrise de Python : fonctions, listes, dictionnaires, boucles, POO et gestion des exceptions.
- Les formations Python : les fondamentaux du langage et Python : programmation orientée objet et traitement de données sont recommandées.
- Bases en statistiques descriptives : moyenne, écart-type, corrélation. Savoir utiliser le terminal et VS Code ou Jupyter Notebook.

3 OBJECTIFS PÉDAGOGIQUES

- Comprendre les fondements du machine learning : apprentissage supervisé, non supervisé et les grands paradigmes
- Préparer et transformer des données pour l'entraînement : feature engineering, encodage et normalisation
- Entraîner et évaluer des modèles de régression : régression linéaire et polynomiale
- Entraîner et évaluer des modèles de classification : régression logistique, arbres de décision, Random Forest, SVM, k-NN
- Maîtriser la validation croisée et les métriques d'évaluation adaptées à chaque problème
- Optimiser les hyperparamètres avec GridSearchCV et RandomizedSearchCV
- Appliquer le clustering non supervisé avec k-means et DBSCAN
- Construire des pipelines scikit-learn reproductibles et maintenables
- Déployer un modèle entraîné via une API FastAPI

4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) – Outils de données et fondamentaux du machine learning

Module 1 – numpy et pandas pour le machine learning (4h)

numpy : calcul vectoriel et matriciel (1h30)

- Les tableaux numpy : création, types, dimensions
- Opérations vectorielles : éviter les boucles avec le broadcasting
- Indexation, slicing et filtrage booléen
- Opérations matricielles : dot, reshape, transpose
- Fonctions statistiques : mean, std, min, max, percentile
- Cas pratique : préparer une matrice de features depuis des données brutes

pandas : manipulation de données tabulaires (1h30)

- Series et DataFrame : créer, lire et explorer
- Lire des fichiers CSV et Excel avec read_csv et read_excel
- Sélectionner des données : loc, iloc, filtres booléens
- Gérer les valeurs manquantes : isnull, fillna, dropna
- Transformer les données : apply, map, groupby, merge, pivot
- Cas pratique : nettoyer et explorer un dataset réel avec pandas

Visualisation avec matplotlib et seaborn (1h)

- matplotlib : courbes, histogrammes, nuages de points
- seaborn : heatmap de corrélation, boxplot, pairplot
- Visualiser les distributions et les relations entre variables
- Cas pratique : tableau de bord exploratoire d'un jeu de données ML

Module 2 – Introduction au machine learning (3h)

Les grands paradigmes du ML (1h)

- Qu'est-ce que le machine learning : apprendre à partir des données plutôt que programmer des règles
- Apprentissage supervisé : régression et classification
- Apprentissage non supervisé : clustering et réduction de dimensionnalité
- Le deep learning dans l'écosystème ML : où se situe scikit-learn
- Cas d'usage réels : détection de fraude, recommandation, prédiction de churn

L'API scikit-learn et préparation des données (2h)

- L'API unifiée scikit-learn : fit, predict, transform
- Train/test split : stratification et taille du jeu de test
- Encodage des variables catégorielles : OneHotEncoder, OrdinalEncoder
- Normalisation et standardisation : MinMaxScaler, StandardScaler, RobustScaler
- Feature engineering : créer de nouvelles variables pertinentes
- Sélection de features : corrélation, variance et importance
- Cas pratique : pipeline de préparation complet sur un dataset de prédiction de prix

Jour 2 (7h) – Régression et classification

Module 3 – Régression (3h)

Régression linéaire (1h30)

Le modèle linéaire : coefficients, interception et hypothèses

- Entraîner une régression linéaire avec scikit-learn
- Les métriques de régression : MAE, MSE, RMSE, R^2
- Analyser les résidus : détecter les violations des hypothèses
- Cas pratique : prédire le prix d'un bien immobilier

Régression avancée (1h30)

- Régression polynomiale : capturer les relations non linéaires
- Régularisation : Ridge (L2) et Lasso (L1) pour éviter l'overfitting
- ElasticNet : combiner L1 et L2
- Cas pratique : comparer régression linéaire, Ridge et Lasso sur un dataset bruité

Module 4 – Classification (4h)

Régression logistique (1h)

- De la régression à la classification : la fonction sigmoïde
- Classification binaire et multiclasse avec régression logistique
- Les métriques de classification : accuracy, précision, rappel, F1-score
- La matrice de confusion : interpréter les erreurs du modèle
- Cas pratique : classification de clients à risque de churn

Arbres de décision et Random Forest (1h30)

- L'arbre de décision : comment le modèle prend ses décisions
- Hyperparamètres clés : max_depth, min_samples_split, criterion
- Random Forest : l'ensemble d'arbres pour réduire la variance
- L'importance des features dans Random Forest
- Cas pratique : classifier des emails spam avec Random Forest

SVM et k-NN (1h30)

- Support Vector Machine : trouver le meilleur hyperplan de séparation
- Le kernel trick : SVM non linéaire avec noyaux RBF et polynomial
- k-Nearest Neighbors : classification par voisinage et choix de k
- Cas pratique : comparer SVM et k-NN sur un problème de classification

Jour 3 (7h) – Évaluation, optimisation et méthodes d'ensemble

Module 5 – Évaluation et sélection de modèles (3h)

Validation croisée (1h)

- Le problème du surapprentissage : train score vs test score
- La validation croisée k-fold : principe et implémentation
- Stratified k-fold : respecter les proportions de classes
- Cas pratique : comparer plusieurs modèles avec validation croisée

Métriques avancées (1h)

- La courbe ROC et l'AUC : évaluer un classifieur binaire
- Précision-rappel : quand l'AUC ne suffit pas (classes déséquilibrées)
- Gestion des classes déséquilibrées : class_weight et oversampling
- Cas pratique : évaluer un modèle de détection de fraude

Optimisation des hyperparamètres (1h)

- GridSearchCV : recherche exhaustive dans une grille de paramètres
- RandomizedSearchCV : recherche aléatoire plus efficace
- Cas pratique : optimiser un Random Forest avec GridSearchCV

Module 6 – Méthodes d'ensemble et Gradient Boosting (2h)

Gradient Boosting et XGBoost (1h)

- Le boosting : apprendre des erreurs des modèles précédents
- GradientBoostingClassifier : hyperparamètres clés
- XGBoost et LightGBM : les implémentations optimisées
- Cas pratique : comparer Random Forest et XGBoost sur un benchmark

Interprétabilité des modèles (1h)

- Feature importance : Random Forest et Gradient Boosting
- Permutation importance : une mesure plus robuste
- Partial Dependence Plots : visualiser l'effet d'une feature
- Cas pratique : expliquer les prédictions d'un modèle de crédit

Module 7 – Clustering et réduction de dimensionnalité (2h)

Clustering (1h)

- L'algorithme k-means : centroïdes, assignation et convergence
- Choisir k : la méthode du coude et le score de silhouette
- DBSCAN : clustering par densité sans nombre de clusters fixé
- Cas pratique : segmenter des clients par comportement d'achat

Réduction de dimensionnalité (1h)

- Le fléau de la dimensionnalité : pourquoi trop de features nuit
- PCA : principe, variance expliquée et choix du nombre de composantes
- Cas pratique : visualiser des données en 2D avec PCA

Jour 4 (7h) – Pipelines et atelier de projet

Module 8 – Pipelines scikit-learn (3h)

Construire un pipeline reproductible (1h30)

- Le Pipeline scikit-learn : chaîner des transformateurs et un estimateur
- ColumnTransformer : appliquer des transformations différentes par type de colonne
- Intégrer le feature engineering dans le pipeline
- Utiliser GridSearchCV sur un pipeline complet

Sauvegarder et versionner un modèle (1h30)

- Sérialiser un pipeline avec joblib : dump et load
- Versioning des modèles : bonnes pratiques pour la traçabilité
- MLflow : suivre les expériences, les paramètres et les métriques
- Cas pratique : pipeline complet sauvegardé, rechargé et loggé dans MLflow

Module 9 – Atelier de projet (4h)

- Sélectionner un jeu de données réel parmi une liste proposée
- Formuler le problème ML : type de tâche, métriques cibles

EDA avec pandas, numpy et seaborn

- Feature engineering et préparation des données
- Entraîner et comparer plusieurs modèles avec validation croisée
- Optimiser le meilleur modèle et construire le pipeline final avec joblib

Jour 5 (7h) – Déploiement et projet de synthèse

Module 10 – Déployer un modèle ML avec FastAPI (3h)

Exposer un modèle via une API REST (1h30)

- Charger le pipeline joblib au démarrage de l'API avec lifespan
- Créer un endpoint /predict : recevoir les features et retourner la prédiction
- Valider les entrées avec Pydantic : types, contraintes, valeurs manquantes
- Retourner la prédiction et le score de confiance dans la réponse JSON
- Documenter l'endpoint dans Swagger UI
- Cas pratique : API de prédiction de prix immobilier

Conteneuriser et déployer (1h30)

- Écrire un Dockerfile pour l'API FastAPI avec le modèle embarqué
- Docker Compose : API + dépendances
- Déployer sur un VPS ou Render : API accessible publiquement
- Monitorer les prédictions : logger les entrées et les sorties pour détecter le drift
- Cas pratique : déploiement complet du modèle de synthèse

Module 11 – Projet de synthèse (4h)

Réalisation d'un projet ML complet de bout en bout

- Cahier des charges : résoudre un problème de classification ou de régression sur un dataset réel
- Fonctionnalités : exploration et nettoyage des données avec pandas et numpy, EDA avec seaborn et matplotlib, pipeline scikit-learn avec feature engineering, comparaison d'au moins trois modèles avec validation croisée, optimisation des hyperparamètres avec GridSearchCV, interprétabilité avec feature importance, déploiement du meilleur modèle via FastAPI
- Étape 1 : EDA et préparation des données
- Étape 2 : entraînement, évaluation et sélection du modèle
- Étape 3 : pipeline final et sauvegarde joblib
- Étape 4 : déploiement FastAPI et présentation
- Revue collective : choix de modèles, métriques, interprétation des résultats
- Retour formateur individualisé sur le projet rendu

5 COMPÉTENCES VISÉES

- Préparer un jeu de données brut pour l'entraînement d'un modèle de machine learning
- Choisir et entraîner le modèle adapté selon le type de problème (régression, classification, clustering)
- Évaluer les performances d'un modèle avec les métriques appropriées et interpréter les résultats
- Optimiser un modèle par recherche d'hyperparamètres avec validation croisée
- Construire un pipeline scikit-learn de bout en bout reproductible
- Déployer un modèle entraîné en production via une API REST FastAPI

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée en présentiel ou distanciel (visioconférence). Le formateur alterne entre méthode démonstrative (live coding sur Jupyter Notebook avec des jeux de données réels), méthode interrogative (analyse des résultats des modèles et discussion des choix algorithmiques) et méthode active (exercices de modélisation et projet de synthèse fil rouge). L'accent est mis sur la compréhension intuitive des algorithmes et la capacité à choisir le bon modèle selon le contexte.

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Notebooks Jupyter de démonstration avec exercices et corrections par module
- Environnement de développement : VS Code ou Jupyter Lab + Python 3.12+ + scikit-learn + XGBoost + FastAPI
- Pour le distanciel : visioconférence (Zoom ou équivalent), partage d'écran, chat en direct
- Accès à la plateforme pédagogique LaPolaris (supports, ressources, émargement)

8 MODALITÉS D'ÉVALUATION

- En cours de formation : exercices de modélisation corrigés à chaque module
- En fin de formation : réalisation d'un projet ML complet de bout en bout avec déploiement FastAPI
- Questionnaire d'auto-évaluation des acquis en fin de parcours

9 CRITÈRES D'ÉVALUATION

- Exploration et nettoyage corrects des données avec pandas : valeurs manquantes, types, distributions
- Pipeline scikit-learn fonctionnel avec feature engineering intégré et reproductible
- Comparaison rigoureuse de plusieurs modèles avec validation croisée et métriques adaptées au problème
- Interprétation correcte des résultats : métriques, feature importance et limites du modèle
- API FastAPI fonctionnelle exposant le modèle avec validation Pydantic et documentation Swagger

10 MODALITÉS DE VALIDATION

Attestation de fin de formation délivrée à l'issue du parcours, conditionnée à une assiduité d'au moins 80 % et à la réalisation du projet de synthèse. L'attestation précise les objectifs atteints et les compétences acquises.

11 SUIVI ET ACCOMPAGNEMENT

- Feuilles d'émargement signées par demi-journée (présentiel) ou émargement numérique (distanciel)
- Traçabilité des activités pédagogiques réalisées
- Attestation d'assiduité délivrée en fin de formation
- Suivi individuel via les notebooks corrigés et le projet de synthèse

12 CONDITIONS D'ACCÈS

Formation accessible sur inscription directe, sans prérequis administratif particulier. Le financement peut être pris en charge par l'employeur dans le cadre d'un plan de développement des compétences, ou en autofinancement. LaPolaris est un organisme de formation en cours de certification Qualiopi.

13 DÉLAIS D'ACCÈS

Inscription possible jusqu'à **5 jours ouvrés** avant le début de la session. Pour toute demande urgente, nous contacter directement.

S PROCHAINES SESSIONS

20/04/2026 → 24/04/2026
À distance
12 places restantes

18/05/2026 → 22/05/2026
À distance
12 places restantes

01/06/2026 → 05/06/2026
À distance
12 places restantes

08/06/2026 → 12/06/2026
À distance
12 places restantes

15/06/2026 → 19/06/2026
À distance
12 places restantes

22/06/2026 → 26/06/2026
À distance
12 places restantes

ACCESSIBILITÉ · HANDICAP

Nos formations sont accessibles aux personnes en situation de handicap. Pour toute situation nécessitant un aménagement (matériel, temporel ou pédagogique), nous vous invitons à nous contacter avant l'inscription afin d'étudier les adaptations possibles. Référent handicap : contact@lapolaris.fr

LaPolaris
TÉL. +33762584798
EMAIL contact@lapolaris.fr
WEB lapolaris.fr