Fiche du cours

70 h

Titre :

ML300 - Machine Learning (Python & scikit-learn)

Description :

Apprendre le Machine Learning appliqué avec Python : cadrer un problème (régression/classification), préparer les données (Pandas/NumPy), construire des pipelines scikit-learn, entraîner/évaluer des modèles (linéaires, arbres/forêts, gradient boosting, SVM, k-NN, Naive Bayes), explorer l'apprentissage non supervisé (k-means, PCA, DBSCAN), régler les hyperparamètres et interpréter (feature importance, SHAP – aperçu). On insiste sur validation, éviter les fuites de données, gestion du déséquilibre, et un mini-déploiement (API simple).

Objectifs :

- Cadrer un problème ML, choisir métriques adaptées (RMSE, MAE, AUC/F1, etc.).*
- Nettoyer, transformer et standardiser les données ; gérer valeurs manquantes et variables catégorielles.*
- Construire des pipelines scikit-learn (pré-process + modèle) et éviter la data leakage.*
- Entraîner/évaluer des modèles supervisés, tuner (Grid/Randomized Search) avec cross-validation.*
- Appliquer l'unsupervised (clustering, réduction de dimension).*
- Expliquer un modèle (feature importance, SHAP – aperçu) et livrer un prototype (API).

Chapitres :

1. Fondamentaux ML & setup : types de problèmes, biais-variance, train/valid/test, métriques • env Python (venv), Pandas/NumPy*
2. Préparation & EDA : nettoyage, valeurs manquantes, encodage (one-hot/ordinal), scaling, EDA graphique*
3. Régression : linéaire, Ridge/Lasso/ElasticNet, métriques (MAE/RMSE), régularisation*
4. Classification : logistique, Naive Bayes, k-NN ; courbes ROC/PR, seuils, coûts d'erreur*
5. Arbres & Ensembles : Decision Tree, Random Forest, Gradient Boosting (XGBoost/LightGBM – aperçu)*
6. SVM & marges : kernels, réglages, limites et usages pratiques*
7. Non supervisé : k-means, DBSCAN, PCA (réduction), évaluation (silhouette)*
8. Pipelines & Tuning : Pipeline, ColumnTransformer, Grid/RandomizedSearchCV, déséquilibre (class weights, resampling)*

9. Interprétabilité & robustesse : importances, partial dependence, SHAP (aperçu) ; drift, validation temporelle (aperçu séries)*

10. Capstone & déploiement : projet de bout en bout + API FastAPI (aperçu), sérialisation prudente, README & bonnes pratiques

À la fin :

Vous saurez préparer des données, construire des pipelines scikit-learn robustes, évaluer/tuner des modèles, appliquer clustering/PCA, expliquer les résultats et livrer un prototype exploitable — portfolio-ready. Vous saurez aussi choisir les métriques adaptées, éviter la data leakage via les pipelines et exposer le modèle par une petite API (FastAPI) avec une doc claire.