

Data Science and Machine Learning with Python

Practical preparation, modelling, evaluation and deployment awareness

FA-0726 · 4 days; 5 days with Python refresher

Data & Analytics · Software Development

Introduction

Develop a practical Python data-science workflow through statistics, preparation, visualisation and selected machine-learning examples. Use compatible scikit-learn and XGBoost packages to inspect business-style data and evaluate model results.

The core programme is four days for learners with basic Python. An additional preparatory day supports learners who know another programming language but need Python orientation. Big-data and deployment topics are bounded introductions, not a promise of production-system expertise.

Learning outcomes

- Use Python and selected libraries to acquire, transform, clean and visualise sample data.
- Explain covariance, correlation, conditional probability and basic Bayesian/statistical concepts.
- Build selected regression, classification or clustering examples and interpret suitable evaluation metrics.
- Handle missing values, outliers and imbalance while avoiding training/evaluation leakage.
- Recognise Spark MLlib and text-feature workflows and their distributed-processing context.
- Explain deployment, monitoring and experiment-design considerations and identify further work needed.

Prerequisites

- Basics of computer programming

Basic Python programming for the four-day route; the five-day route begins with a Python refresher for learners with general programming knowledge.

- Basic understanding of how the filing system works

A supported compatible Python environment with required packages, locally or in an approved hosted notebook. Colab is optional; Anaconda and unrestricted OS administration are not universal requirements.

Training outline

1. Optional preparatory day — Python orientation

- Use a compatible editor or notebook; review syntax, collections, functions, files, encoding and library installation.
- Read and transform a small dataset and practise the Python constructs used in the core course.
- Consult current documentation instead of treating Python3.8/3.9 changes as recent.

2. Core day 1 — Statistics and exploratory analysis

- A Crash Course in matplotlib.
- Advanced Visualization with Seaborn.
- Covariance and Correlation.
- Conditional Probability.
- Bayes' Theorem.

Acquire approved sample data or a permitted prepared web-source example; inspect data structure and use pandas for bounded transformation.

Create selected Matplotlib/Seaborn views, interpret descriptive summaries and distinguish correlation from causation.

Review the analytical question and document source, cleaning and interpretation assumptions.

3. Core day 2 — Models and evaluation

- Supervised versus unsupervised methods and appropriate training/evaluation separation.
- Use a small polynomial-regression example to inspect overfitting; a split estimates generalisation but does not itself prevent overfitting.
- Introduce Bayesian methods and a supplied naïve-Bayes spam-classification example.
- Explore k-means, entropy, a decision tree and selected XGBoost behaviour.
- Bias/variance and cross-validation: estimate performance and compare settings, without assuming k-fold validation guarantees no overfitting.

4. Core day 3 — Real-world data and features

- Data Cleaning and Normalization
- Cleaning web log data
- Normalizing numerical data
- Detecting outliers
- Feature Engineering
- Imputation Techniques for Missing Data

Consider oversampling, undersampling and SMOTE where appropriate; apply resampling within training folds only, not before an evaluation split.

Use selected binning, transformations, encoding and scaling; respect order where time or grouped observations make random shuffling inappropriate.

Review a small pipeline, its evaluation evidence and limitations.

5. Core day 4 — Distributed examples, deployment and experiments

- Introduce PySpark, MLlib, decision trees and k-means through a prepared small example; distributed processing is not required for every dataset.
- TF-IDF and a bounded search example using supplied reference text, including a Wikipedia-style corpus where permitted.
- Review model packaging, real-time service interfaces, testing and monitoring through a supplied demonstration rather than an operational deployment promise.
- Introduce t-tests, p-values and A/B-test interpretation, including assumptions, effect size, sample size and experiment duration.
- Discuss common experimental pitfalls and complete a workflow review with clear next steps.