Faysal Aziz

Python and Data Science

Intermediate programming and selected advanced analytical methods

FA-0657 · 5 days

Data & Analytics · Software Development

Extend Python programming into data acquisition, preparation, visualisation and selected machine-learning workflows, with supervised labs and a reviewed project.

Introduction

This intensive course is for learners who are comfortable with programming and want to use Python for data analysis and machine learning. It refreshes essential syntax and data structures, then develops data-handling and modelling workflows through a prepared lab.

Participants work with illustrative business-style datasets, files, encodings, numerical arrays, DataFrames and selected acquisition methods. The analytical progression covers preprocessing, supervised and unsupervised methods, recommendation, dimensionality reduction and introductory bandit, neural-network and NLP examples.

The five-day scope combines selected implementation exercises with demonstrations of the wider method catalogue. It does not promise hands-on mastery of every algorithm. A bounded project emphasises appropriate data preparation, evaluation and limitations rather than guaranteed predictions or business benefits.

Learning outcomes

The course teaches participants to:

- Use Python collections, functions, exceptions, regular expressions, recursion and selected object-oriented concepts.
- Manage packages and process files, encodings, CSV/Excel data and permitted external sources.
- Prepare, summarise and visualise data with suitable numerical and DataFrame libraries.
- Distinguish ML, deep learning and broader AI concepts, and explain a modelling lifecycle.
- Split data appropriately and fit preprocessing on training data to avoid leakage.

- Implement and compare selected regression, classification, clustering and recommendation examples.
- Explain dimensionality reduction, association rules, bandit methods, ANN/CNN and NLP examples at the depth supported by the lab.
- Evaluate a bounded project and communicate assumptions, evidence and model limitations.

Prerequisites

- Comfort with programming, functions, loops and basic data structures.
- Familiarity with files, folders and command-line or notebook workflows.
- Basic numerical/statistical concepts are helpful for interpreting models and evaluation.

Access to Google Colab or a prepared local Python environment with the selected compatible packages. Anaconda is one option, not a mandatory dependency. Any installation privileges, account access, internet connectivity and service limits must be checked before training.

Training outline

1. Day 1 – Python Programming and Data Handling

Python refresher

- Syntax structure
- Operations
- Python Variables
- Loops in Python
- Python collection Data Types
- Object-oriented programming concepts

Intermediate programming and numerical data

- Exception Handling
- Regular Expression
- NumPy arrays and numerical operations
- Matrix and its operation
- Functions in Python
- User Defined functions in Python
- Scope in Python

- Introduction to Methods

Selected advanced techniques

- Recursion
- Memoization
- Selected advanced libraries appropriate to the lab

2. Day 2 – Data Acquisition, Analysis and Preparation

Packages and data sources

- Packages in Python and PIP
- Pandas and Data frames
- Import and Export data from CSV and Excel
- Permitted web-data acquisition: respect access restrictions, privacy and service conditions
- Other sources of data
- Files and encodings, statistical summaries and selected data visualisations.

Machine-learning foundations

- What is Machine Learning?
- Process of Machine Learning
- Life Cycle of Machine Learning
- Applications, potential benefits and limitations of machine learning
- Types Of Machine Learning
- Reviewing the prepared ML development environment

Preprocessing and evaluation setup

- Load data and split training/validation/test subsets before learning transformations.
- Handle missing values, categorical encoding and scaling using training-data-fitted preprocessing.
- Use consistent transformations and suitable validation for the problem.

3. Day 3 — Supervised Learning and Ensembles

Supervised methods

- What is Supervised Learning?
- Types of Supervised Learning

Regression

- Simple Linear Regression
- Multiple Regression
- Polynomial Regression

Decision trees and random forests

- Selected decision-tree and random-forest regression implementations.

Classification

- Selected logistic regression, Naive Bayes and support-vector-machine examples.

Gradient boosting

- XGBoost as an ensemble/gradient-boosting method, not a neural network.
- Compare a prepared model using suitable held-out evaluation.

4. Day 4 — Unsupervised Methods, Recommendations and Reduction

Clustering

- K-means Clustering
- K-means Selecting the Number of Clusters
- Implementation of k-means clustering
- Implementation of Hierarchical Clustering

Association-rule mining

- Apriori example and interpretation of association rules; distinguish from clustering and causal evidence.

Collaborative filtering

- User based
- Item based
- Implementation

Dimensionality reduction

- Implementation of Principal Component Analysis
- Linear Discriminant Analysis: supervised projection and its distinction from unsupervised PCA
- Implementation of Kernel PCA

5. Day 5 — Selected Advanced Examples and Project

Bandit methods

- Implementation of Upper Confidence Bound (UCB)
- Implementation of Thompson Sampling

Upper Confidence Bound and Thompson Sampling are presented as introductory multi-armed-bandit examples, not a complete sequential reinforcement-learning curriculum.

Neural-network demonstrations

- Implementation of Artificial Neural Networks (ANN)
- Implementation of Convolutional Neural Network (CNN)

Use prepared ANN/CNN examples with compatible libraries; this is an introduction rather than an independent deep-learning specialisation.

Natural language processing

- Implementation of Natural Language Processing in Python

Bounded project and evaluation

- Prepare data, implement selected methods and evaluate against the project objective.
- Review leakage, errors, assumptions and communication of results.

© 2026 Faysal Aziz · Kuala Lumpur