Faysal Aziz

Advanced Python for AI

A three-day applied workshop in modelling, language-model integration and bounded agents

FA-0553 · 3 days

Data & Analytics · Agentic & Generative AI · Software Development

Apply Python modelling workflows, evaluate selected ML/deep-learning methods and build a supervised LLM or agent prototype.

Introduction

This three-day advanced workshop is for Python users with prior data-science and basic machine-learning experience. It reviews data preparation and model evaluation, demonstrates selected neural-network approaches and then introduces language-model APIs, local inference and bounded agent workflows.

The practical objective is a guided prototype, not mastery of every algorithm or deployment of an autonomous production system. CNNs, sequence models, transfer learning, GANs and reinforcement learning are compared at the depth the timetable allows; selected techniques are implemented.

Local-model exercises depend on a compatible runtime, available memory/accelerator resources and the chosen model's access and licence conditions. API-based exercises use approved accounts and non-confidential sample data, with explicit limits and human review of tool actions.

Learning outcomes

The workshop teaches participants to:

- Clean, transform and visualise data with Python data-science libraries.
- Train and evaluate selected regression, classification or clustering models with appropriate validation.
- Explain selected neural-network architectures and inspect a guided deep-learning example.
- Integrate a language model through an API and evaluate a small local-inference example where resources permit.
- Build a bounded agent workflow with limited tools and reviewed actions.

- Discuss model selection, overfitting, deployment constraints and the limitations of a combined AI prototype.
- Present a capstone and identify the additional data, evaluation and engineering needed for a real deployment.

Prerequisites

- Proficiency in Python: Strong command of Python programming, including familiarity with syntax, control structures, data types, and functions. Experience with data science libraries such as NumPy and Pandas is essential.
- Mathematical Background: Solid understanding of algebra, calculus, statistics, and probability to interpret data and model outcomes effectively.
- Basic ML Knowledge: Familiarity with fundamental machine learning concepts and terminology, including supervised and unsupervised learning, overfitting, and model evaluation metrics.
- Software and Tools: Comfort with Python development environments (e.g., Jupyter Notebooks, PyCharm) and package management tools (conda, pip). Basic experience with version control systems like Git is advantageous.
- Analytical Skills: Ability to think critically about problems and data, with experience in data manipulation and visualization tools.
- Curiosity and Experimentation: An eagerness to learn new concepts, experiment with different algorithms, and apply creative solutions to problems in the rapidly evolving fields of data science and AI.

A configured Python development environment with compatible packages and approved API access where required. Local LLM inference may use a separate PyTorch/Transformers environment from TensorFlow/Keras examples; no accelerator or unlimited model-hosting capacity is assumed.

Training outline

1. Day 1 – Python Data Preparation and Analytics

- Introduction to Data Science with Python: Overview of data science principles and the role of Python in data analysis.
- Data Manipulation with Pandas: Techniques for data cleaning, transformation, and analysis using Pandas.
- Data Visualization with Matplotlib and Seaborn: Creating insightful visualizations to interpret data effectively.
- Applied exercises with public, synthetic or approved non-confidential datasets.

2. Day 2 — Machine Learning and Neural Networks

- Supervised Learning Algorithms: Implementing regression and classification models using Scikit-learn.
- Unsupervised Learning Techniques: Exploring clustering and dimensionality reduction methods.
- Model Evaluation and Validation: Assessing model performance and applying cross-validation techniques.
- Neural Network Architectures: Understanding the structure and function of neural networks.
- Build and train a selected TensorFlow/Keras neural-network example; inspect results rather than promise coverage of every deep-learning architecture.
- CNN image-processing example: architecture and selected demonstration.
- Sequence models: RNN/LSTM concepts, applications and limitations.

3. Day 3 — Language Models, Bounded Agents and Capstone

- Agentic AI concepts: task planning, tools and bounded action; distinguish a prototype from reliable autonomous operation.
- Designing Agentic Systems: Principles and frameworks for developing agentic AI systems.
- Build a small supervised agent prototype with constrained tool access and testable stopping conditions.
- Understanding LLMs: Overview of Large Language Models and their applications in natural language processing.
- API Integration of LLMs: Techniques for integrating LLMs into applications using APIs.
- Local LLM inference with a suitably sized Hugging Face model: compatible runtime, resource needs, licence/access conditions and a demonstrable test configuration.
- Conceptual comparison of transfer learning, GANs and reinforcement learning; optional further study rather than mastery in the core timetable.

Capstone and Review

Develop and present a guided capstone connecting selected Python/ML and language-model components for an illustrative problem; discuss failure cases and limitations.

© 2026 Faysal Aziz · Kuala Lumpur