Generative AI, Agentic AI and MCP

with AI/ML Foundations

Duration : 14 Days (8 hrs/day) – 112 hrs

Pre-requisite

- Working knowledge of the Python programming language

DAY 1 — Module 1 : Data Science

- Introduction to Data science
- Supervised and Unsupervised learning
- Supervised - Regression and Classification algorithms
- Unsupervised Learning algorithms
- Lab : Data analysis insights
- Lab : Regression
- Lab : Classification
- Lab : Unsupervised – K means clustering
- Lab : Anamoly detection
- Deploying a data science solution with backend (flask /fastapi) locally hosted

DAY 2 — Module 2 : Deep Neural Network

- Understanding Perceptron
- Lab: Perceptron
- Understanding Artificial Neural network / multi-layer perceptron
- Lab: ANN
- Understanding Convolutional Neural network – CNN
- Padding, strides, pooling, Transfer learning etc
- Flattening layer and prediction
- LAB: CNN prediction
- Understanding Recurrent neural network
- RNN, LSTM, GRU
- Lab: RNN

DAY 3 — Module 3 : Natural Language Processing

- Converting language to numbers capturing semantic relationship
- Techniques Bag of words, TF-IDF, Word2Vec
- Name entity recognition
- Lab: Sentiment Analysis lab on e-commerce Reviews dataset
- Lab: Name entity recognition
- Lab: Word cloud

DAY 4 — Module 4 : Introduction to Generative AI, LLMs and Hugging Face

- **Foundations of Generative AI**
  - Understanding the background and backdrop of Generative AI
  - Discriminative vs generative models
  - Understanding Transformer architecture – encoder, decoder and self-attention
  - Evolution of Large Language Models (LLMs) – base, instruct and chat models
  - Tokens, embeddings, context window, temperature and decoding parameters

    - Introducing AI-Driven SDLC Process Changes
- **Working with Hugging Face**
    - Introduction to the Hugging Face Hub – models, datasets and spaces
    - Pipelines, tokenizers and auto-classes
    - Selecting the right open-source model for a task

DAY 5 — Module 5 : Hugging Face Labs – Text and Multimodal Tasks

- **Labs – Text to Text tasks**
    - Lab: Text completion and text generation
    - Lab: Text classification and zero-shot classification
    - Lab: Language translation
    - Lab: Sentence similarity and semantic search
    - Lab: Summarization and question answering
- **Labs – Multimodal tasks**
    - Lab: Text to image generation
    - Lab: Image to text and image captioning
    - Lab: Object detection and image segmentation
    - Lab: Image based question answering (visual QnA)
    - Lab: Automatic Speech Recognition (ASR) and speech to text

DAY 6 — Module 6 : Ollama and Local Model Hosting | Module 7 : Prompt Engineering and Techniques

- Introduction to Ollama and local model hosting
- Pulling, running and managing open-source models – Llama, Mistral, Gemma, Phi
- Lab: Installing Ollama and running a local LLM
- Lab: Building a Python application on a locally hosted model
- Lab: Creating a web application on base models
- **Prompt engineering and techniques**
    - Anatomy of a prompt – system, user and assistant roles
    - Zero shot, one shot and few shot prompting
    - Chain of thought, role based and structured output prompting
    - Prompt templates, delimiters and output formatting
    - Guardrails, prompt injection and common failure patterns
    - Lab: Prompt engineering techniques

DAY 7 — Module 8 : Chatbots with LangChain

- **LangChain fundamentals**
    - LangChain architecture and ecosystem overview
    - Models, prompt templates and output parsers
    - Building chains for multi-step tasks
    - Connecting LangChain to hosted and local (Ollama) models
    - Lab: Simple LangChain application
- **Building conversational chatbots**
    - Conversation memory types – buffer, window and summary memory
    - Streaming responses and token usage tracking
    - Creating a customised conversational chatbot on an LLM
    - Lab: Chatbot creation with LangChain
    - Lab: Multi-turn chatbot with memory and a custom persona

DAY 8 — Module 9 : Vector Stores and Retrievers

- **Embeddings and vector stores**
  - Text embeddings and semantic similarity
  - Introduction to vector databases and their internals
  - Working with ChromaDB – collections, metadata filtering and persistence
  - Lab: Generating embeddings and storing them in ChromaDB
- **Document processing and retrievers**
  - Document loaders – PDF, CSV, DOCX and web pages
  - Chunking strategies, chunk size and overlap
  - Retrievers – similarity search, MMR and top-k tuning
  - Lab: Vector stores and retrievers

DAY 9 — Module 10 : RAG and SQL RAG

- **Retrieval Augmented Generation (RAG)**
  - Understanding the RAG architecture – indexing and retrieval pipelines
  - Building the augmented prompt with retrieved context and citations
  - Controlling hallucination and handling “no answer” cases
  - Evaluating RAG quality – relevance, groundedness and latency
  - Lab: PDF RAG
  - Lab: QnA RAG chatbot over custom documents
- **SQL RAG on structured data**
  - Bringing LLMs to relational data – text to SQL
  - Schema awareness, query generation, validation and safe execution
  - Combining SQL RAG with vector retrieval for hybrid answers
  - Lab: SQL RAG on a relational database with ChromaDB

DAY 10 — Module 11 : Hosting and Logging a Containerised RAG Application

- Wrapping a RAG pipeline in a FastAPI service
- Containerising the application with Docker – image, environment variables and persistence
- Hosting the container and exposing the RAG endpoint
- Introduction to LangSmith – tracing, runs and projects
- Logging and monitoring – latency, token cost, retrieval quality and failed runs
- Debugging and evaluating traces in LangSmith
- Lab: Containerising a RAG application and deploying it
- Lab: Logging a hosted containerised RAG app with LangSmith

DAY 11 — Module 12 : AI Agents – Single Agent to Multi-Agent Systems

- **Agent foundations**
  - Agentic AI vs other Generative AI applications
  - Reasoning and acting (ReAct) – tools, planning and memory
  - Designing a single agent – roles, goals and tool selection
  - Lab: Building a single agent with tool calling
- **Orchestrating agents with LangGraph**
  - Introducing LangGraph and how it compares with other frameworks
  - Understanding nodes, edges and conditional edges
  - State management, checkpoints and persistence
  - Lab: Building an agentic chatbot
  - Lab: Building a RAG agentic chatbot

- **Multi-agent patterns and control**
  - Multi-agent collaboration and the supervisor pattern
  - Routers and conditional branching between agents
  - Parallelisation and fan-out / fan-in workflows
  - Human in the loop – interrupts, approvals and state editing
  - Lab: Human in the loop approval workflow
  - Lab: Building multi-agent workflows with routing and parallelisation

DAY 12 — Module 13 : Model Context Protocol (MCP) Servers

- **MCP fundamentals**
  - What is the Model Context Protocol and the problem it solves
  - Agents vs MCP – comparison and when to use which
  - Architecture and primitives – hosts, clients, servers, tools and resources
  - Transports – stdio vs HTTP / SSE
- **Local and hosted MCP servers**
  - Building a local MCP server over stdio and exposing tools
  - Deploying a hosted MCP server over HTTP / SSE with basic authentication
  - Lab: Build and run a local MCP server
  - Lab: Deploy a hosted MCP server
- **MCP clients**
  - Connecting desktop, IDE and custom Python clients
  - Consuming MCP servers from LangGraph agents
  - Lab: An agent consuming local and hosted MCP servers

DAY 13 — Module 14: MLflow and Production Model Lifecycle Management

- **MLflow Fundamentals and Experiment Tracking**
  - What is MLflow and its role in MLOps/ GenAIOps
  - Experiment tracking, parameter logging, metrics logging, and artifact management
  - Managing reproducible ML workflows and model lineage
  - Best practices for organizing ML experiments
- **Model Registry and Version Management**
  - Model Registry concepts and governance workflows
  - Model versioning, stage transitions, and approvals
  - Managing model lifecycle from development to production
  - Model lineage, traceability, and compliance considerations
- **Automated Evaluation and Model Validation**
  - Performance metrics and evaluation strategies
  - Comparing multiple model versions using MLflow
  - Automated validation and model acceptance criteria
  - Monitoring drift and identifying retraining requirements
- **Model Retraining and Continuous Improvement**
  - End-to-end model retraining lifecycle
  - Triggering retraining using performance thresholds
  - Governance and approval workflows for retrained models
  - Production readiness and release management

DAY 14 — Module 15: Deployment and CI/CD for AI Applications

- **Deploying AI Applications with Containers**
  - Introduction to Docker containers and container ecosystems
  - Building container images for ML and RAG applications
  - Running AI applications in self-hosted environments
  - Infrastructure-agnostic deployment considerations
- **Self-Hosted and Azure Container Deployments**

- Deploying containerized applications to Azure Container Instances (ACI)
- Comparing self-hosted and cloud-hosted deployment approaches
- Logging, monitoring, and operational best practices

- **CI/CD Fundamentals for AI Applications**
  - DevOps principles for AI solutions
  - GitHub repositories, branching strategies, and pull request workflows
  - Understanding GitHub Actions workflows and automation pipelines
  - Feature branch validation and code quality checks
- **Automated CI/CD for ML and RAG Solutions**
  - Continuous Integration for model training and testing
  - Automated container image build and validation
  - Continuous Deployment to Azure Container Instances (ACI) (Demo Only)
  - Integrating MLflow, deployment approvals, monitoring, and retraining pipelines