Mastering Generative AI in Python

Duration: 24 Training Hours | **Level:** Beginners/Intermediate

Course Overview

"Mastering Generative AI in Python" is a comprehensive course designed to help you gain both the theoretical understanding and hands-on experience necessary to excel in the field of

Generative AI using Python. Anchored around an HR use case, the course takes learners through a full-stack GenAI project using modern tools and frameworks like HuggingFace, LangChain, Gradio, and introduces foundational concepts of AI agents and Model Context Protocol (MCP). The full stack application eventually gets deployed using Docker and Kubernetes. The course is structured to help learners build real-world applications through modular and progressively complex content. It emphasizes practical, production-ready skills that are immediately applicable in industry environments.

About this course

- Training delivery: Online Live Training
- Skill Level: Beginners/Intermediate
- Duration: 24 Training Hours

Learning Outcomes

By the end of this course, learners will be able to:

- Understand the fundamentals of Artificial Intelligence and Generative AI
- Apply foundational and advanced concepts of LLMs in real-world projects
- Build user-friendly interfaces for AI using Gradio
- Integrate multimodal data inputs and outputs into AI workflows
- Use HuggingFace and LangChain to build robust AI applications
- Design effective prompts using advanced prompting techniques
- Implement structured outputs using function calling and validation
- Employ vector databases and retrieval-augmented generation for context-aware AI
- Understand basic AI agent concepts and architectures
- Get introduced to Model Context Protocol (MCP) fundamentals
- Utilize guardrails to ensure responsible AI behavior
- Deploy a full-stack GenAI application using Docker and Kubernetes
- Ensure model reliability and security in production environments

Course Objectives

The course objectives are to:

1. **Introduce Key Concepts in AI and Machine Learning:** Provide learners with a solid understanding of Artificial Intelligence, including the foundational theories behind machine learning and deep learning.
2. **Explore Generative AI Technologies:** Familiarize learners with the components and mechanisms of Generative AI models, focusing on how they are designed, built, and applied.
3. **Teach the Fundamentals of Large Language Models (LLMs):** Help learners understand the architecture, training, and deployment of LLMs, covering essential concepts like tokenization, embeddings, and fine-tuning.
4. **Demonstrate Practical Use of AI Tools:** Guide learners in using popular AI frameworks like HuggingFace, LangChain for building AI-driven applications in Python.
5. **Develop Proficiency in Prompt Engineering:** Equip learners with the knowledge and skills to craft effective prompts that optimize the performance of AI models in various use cases.
6. **Apply Embeddings and Vector Databases:** Teach the importance of embeddings and how to work with vector databases for AI applications such as search and recommendation engines.
7. **Build Interactive Frontends for AI Applications:** Enable learners to develop user interfaces using Gradio, integrating generative AI features for enhanced functionality.
8. **Introduce AI Agent Concepts:** Provide foundational understanding of AI agents and their basic implementation patterns.
9. **Introduce Model Context Protocol:** Familiarize learners with MCP basics and its role in modern AI applications.
10. **Secure and Operationalize GenAI Applications:** Empower learners to deploy and manage full-stack GenAI apps using Docker, Kubernetes, and CI/CD workflows.

Where can I apply these learning outcomes?

Learners who complete this course will be well-positioned for a variety of AI and ML-focused roles, such as:

• **Generative AI Developer** - Building AI-powered applications and solutions
• **AI/ML Developer** - Developing machine learning models and systems
• **LLM Application Developer** - Creating applications that leverage large language models
• **Junior AI Engineer** - Working on AI implementation projects
• **AI Product Developer** - Building AI-driven products and features • **Data Scientist** - Applying AI in business processes and data analysis
• **DevOps Engineer** - Specializing in AI/ML deployment and operations

These skills are especially relevant in sectors undergoing AI-driven transformation such as **HRTech, EdTech, FinTech, HealthTech, Retail, Customer Experience, and Enterprise Software**.

Course Prerequisites

• Basic understanding of computers and software development concepts
• Basic Python programming knowledge (variables, functions, loops, conditionals)
• Familiarity with command line/terminal operations
• Basic understanding of APIs and web services (helpful but not mandatory)

Module 1: Introduction to AI

- What is AI?
- AI: Key concepts
- Standard AI models
- Machine Learning
- Deep Learning

Module 2: Foundational Models and Generative AI

- How do we Learn?
- Human Brain and AI
- Self-Supervised Learning
- Foundational Models
- Transfer Learning
- Transformers
- Introduction to GenAI
- GenAI use cases
- GenAI Application Lifecycle

Module 3: Introduction to LLMs

- What is an LLM?
- Characteristics of an LLM
- Building Blocks of an LLM
- Types of LLMs
- Fine-tuning LLMs
- Open Source and Closed Source Models
- Deploying LLMs Locally
- Deploying LLMs on Cloud

- Evaluating an LLM

Module 4: Building UIs with Gradio

- Introduction to Gradio
- Creating basic interfaces
- Advanced Gradio features
- Deploying Gradio applications
- Integrating LLMs with Gradio
- Building chatbots with Gradio

Module 5: Multimodal AI

- Text-to-image models
- Text-to-video models
- Image understanding capabilities
- Audio processing in AI
- Multimodal embeddings
- Building multimodal applications

Module 6: Introduction to HuggingFace

- What is HuggingFace?
- What Can We Do with HuggingFace?
- Features of HuggingFace (Models, Datasets, Spaces)
- Transformers library
- Using HuggingFace inference endpoints

Module 7: Prompt Engineering

- What is Prompt Engineering?
- How to write an effective prompt
- Prompt Engineering Techniques:
    - Zero-shot prompting
    - Few-shot prompting
    - Chain-of-Thought
    - Tree of Thoughts
    - ReAct prompting
    - Meta-prompting

Module 8: Structured Outputs

- Reliable JSON output generation
- Using Pydantic with LLMs
- Function calling
- JSON schema validation
- Parsing and error handling
- Real-world applications

Module 9: Embeddings and Vector Databases

- What are embeddings in AI/ML?
- Text and multimodal embeddings
- What is a Vector Database?
- Introduction to vector database options (Pinecone, ChromaDB, Weaviate, Milvus/Zilliz, FAISS)
- Applications of Embeddings

Module 10: Retrieval Augmented Generation (RAG)

- What is RAG?
- Why is RAG important?
- How does RAG work?
- Basic components (retriever, generator)
- Simple RAG implementation example
- Advanced RAG techniques (introductory level)
- Common RAG challenges and solutions

Module 11: LangChain Framework

- What is LangChain?
- Core components (LLM chains, Retrieval chains, Memory, Tools)
- Tool Calling and Routing Chains
- Using various models with LangChain
- Building RAG applications with LangChain

Module 12: Introduction to AI Agents

- What is an AI Agent?

- Basic agent concepts and architecture
- Simple agent implementation
- Agent vs. traditional AI models
- Use cases for AI agents
- Introduction to LangGraph (basic concepts)

Module 13: Model Context Protocol (MCP) - Introduction

- Introduction to Model Context Protocol
- Why MCP matters for GenAI applications
- Basic MCP concepts and architecture
- Simple MCP server setup
- Integration with LLM applications

Module 14: Guardrailing GenAI Applications

- Tools & Techniques for Implementing Guardrails
- Overview of guardrailing solutions
- Data Curation and Bias Detection
- Human-in-the-Loop Processes
- Regular Review and Updates

Module 15: AI for Business Applications

- How GenAI can drive business transformations
- Industry-specific use cases
- Integrating GenAI with existing business processes
- Developing an AI strategy
- ROI measurement

Module 16: Overview of LLMOps

- What is LLMOps and its importance?
- Model versioning, retraining, and updates
- Scaling infrastructure for LLMs
- CI/CD pipelines for LLM deployment
- Monitoring and observability basics

Module 17: Securing LLM Applications

- Best Practices to secure LLM Apps
- Prompt injection prevention
- Information leaks and PII protection
- Preventing jailbreaks
- LLM reliability and safety measures

Module 18: Final Project: Full-Stack GenAI Application

- Designing a comprehensive application
- End-to-end GenAI application development
- Containerization with Docker
- Deployment strategies
- Performance optimization
- Project presentation and review