

**Intermediate Generative AI Engineering for Data Scientists and ML Engineers**

**Course Number:** AI-132WA  
**Duration:** 2 days

**Overview**

This Intermediate Generative AI (GenAI) course is for Machine Learning and Data Science professionals who want to learn advanced GenAI and LLM techniques like fine-tuning, domain adaptation, and evaluation. Participants learn how to leverage popular tools and frameworks, including Python, Hugging Face Transformers, and open-source LLMs.

**Prerequisites**

* Practical experience (6+ months) minimum in Python - functions, loops, control flow
  + Data Science basics - NumPy, pandas, scikit-learn
* Solid understanding of machine learning concepts and algorithms
  + Regression, Classification, Unsupervised learning (clustering, Neural Networks)
* Strong foundations in probability, statistics, and linear algebra
* Practical experience with at least one deep learning framework (e.g., TensorFlow or PyTorch) recommended
* Familiarity with natural language processing (NLP) concepts and techniques, such as text preprocessing, word embeddings, and language models

**Materials**

All Generative AI training students receive comprehensive courseware.

**Software Needed on Each Student PC**

All attendees must have a modern web browser and an Internet connection.

**Objectives**

* Master advanced fine-tuning techniques for LLMs to improve the quality and relevance of outputs
* Implement Retrieval Augmented Generation (RAG) to enhance LLM capabilities with real-time information retrieval
* Utilize vector embeddings for semantic search, recommendations, and other NLP applications
* Optimize LLM performance with techniques like quantization and pruning, ensuring efficient deployment and serving
* Address ethical considerations and adhere to best practices in AI development, mitigating bias, ensuring transparency, and safeguarding privacy

**Outline**

* Introduction
* Advanced Fine-Tuning and RAG Techniques
  + Advanced fine-tuning techniques for LLMs
  + Implementing Retrieval Augmented Generation (RAG)
    - Improving LLM output quality and relevance
  + Building a RAG-powered LLM application for a specific use case
* Vector Embeddings and Semantic Search
  + Introduction to vector embeddings and their applications in NLP
  + Using vector embeddings for semantic search and recommendation systems
    - Generating vector embeddings from text data
    - Implementing a similarity search using libraries like Faiss or Annoy
* LLM Optimization and Efficiency
  + Techniques for optimizing LLM performance
    - Quantization and pruning
  + Applying optimization techniques to reduce LLM model size and inference time
  + Strategies for efficient deployment and serving of LLMs in production
* Ethical Considerations and Best Practices
  + Addressing biases and fairness issues in LLMs
  + Ensuring transparency and accountability in LLM-powered applications
  + Best practices for responsible AI development and deployment
  + Navigating privacy and security concerns when working with LLMs and sensitive data
* Conclusion