

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

**Course Number:** AI-134WA
**Duration:** 5 days

**Overview**

This Comprehensive Generative AI (GenAI) course is for Machine Learning and Data Science professionals who want to dive deep into the world of GenAI and Large Language Models (LLMs). This course covers various topics, from the foundations of LLMs to advanced techniques like fine-tuning, domain adaptation, and evaluation. Participants gain hands-on experience with 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**

* Gain a deep understanding of Large Language Models (LLMs) and their foundational concepts, including generative AI and transformer architecture
* Master prompt engineering techniques to effectively communicate with LLMs and achieve desired outcomes across various NLP tasks
* Evaluate and compare different LLMs to select the most suitable model for specific natural language processing tasks
* Fine-tune and adapt open-source LLMs using domain-specific datasets to optimize performance for specialized applications
* Implement advanced fine-tuning techniques and Retrieval Augmented Generation (RAG) to enhance LLM capabilities
* Utilize vector embeddings for semantic search, recommendations, and other NLP applications
* Optimize LLM performance and efficiency with techniques like quantization and pruning for streamlined deployment and serving
* Navigate ethical considerations and implement best practices to address biases, ensure transparency, and protect privacy when working with LLMs and sensitive data

**Outline**

* LLM Foundations for ML and Data Science
	+ Overview of Generative AI and LLMs
	+ LLM Architecture and Training Techniques
		- Deep dive into the transformer architecture and its components
		- Exploring pre-training, fine-tuning, and transfer learning techniques
* Prompt Engineering for LLMs
	+ Introduction to Prompt Engineering
		- Techniques for creating effective prompts
		- Best practices for prompt design and optimization
	+ Developing prompts for various NLP tasks
		- Text classification, sentiment analysis, named entity recognition
* LLM Evaluation and Comparison
	+ Overview of metrics and benchmarks for evaluating LLM performance
	+ Techniques for comparing LLMs and selecting the best model for a given task
	+ Evaluating and comparing LLMs for a specific NLP task
* Fine-Tuning and Domain Adaptation
	+ Introduction to Open-Source LLMs
		- Advantages and limitations in ML and data science projects
	+ Preparing domain-specific datasets for fine-tuning LLMs
	+ Techniques for adapting LLMs to new domains and tasks using transfer learning
	+ Fine-tuning and adapting an open-source LLM for a specific domain
* 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