Local Large Language Model (Local LLM) Development for Enterprise Environments Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

The Local Large Language Model (Local LLM) Development for Enterprise Environments training course is designed for technical professionals who want to build AI applications that meet organizational requirements for data privacy, security, and performance by deploying large language models within their own infrastructure.

The course covers local AI infrastructure planning, hardware requirements, and the selection of open-source language models. Participants learn to prototype, test, and compare models using LM Studio before moving on to building customized models with Ollama and integrating them into Python and JavaScript applications.

By the end of the program, participants will have the knowledge and practical skills needed to plan an end-to-end local AI solution that runs within enterprise infrastructure, provides API-based access, and uses internal company documents as a knowledge source.

Prerequisites

Participants are required to have basic Python or JavaScript programming knowledge and familiarity with command-line operations. Introductory knowledge of Docker is also recommended.

Who Should Attend

- Software developers interested in building AI-powered applications and solutions
- Data scientists and machine learning engineers
- DevOps professionals and infrastructure engineers
- Software architects and solution architects
- Technical team leaders responsible for enterprise AI development projects

Local Large Language Model (Local LLM) Development for Enterprise Environments Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

Upon completing this course, you will be able to:

- Compare local and cloud-based large language models in terms of cost, security, and performance.
- Plan local AI architectures while considering GDPR and KVKK compliance requirements.
- Determine appropriate hardware capacity for enterprise AI workloads.
- Evaluate open-source language models based on licensing, model size, and technical capabilities.
- Test language models with LM Studio and expose OpenAI-compatible local API services.
- Create customized model configurations using Ollama and Modelfile.
- Integrate local LLM solutions into Python and JavaScript applications.
- Configure model access and user management using Docker and Open WebUI.
- Design RAG-based AI assistants that work with internal organizational documents.

Local Large Language Model (Local LLM) Development for Enterprise Environments Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Outline

1. Planning Local AI Infrastructure

Enterprise Strategy and Technology Selection

- Comparing cloud-based and locally deployed language model architectures
- Evaluating data security, privacy, and compliance requirements under GDPR and the Turkish Personal Data Protection Law (KVKK)
- Analyzing token-based usage costs and the return on hardware investments

Hardware and Operating System Requirements

- Capacity planning for processors, GPU memory (VRAM), and system memory (RAM)
- Preparing Windows, macOS, and Linux server environments
- Configuring CUDA and Metal hardware acceleration frameworks

Open-Source Language Model Ecosystem

- Reviewing model licenses, sizes, and capabilities through Hugging Face model cards
- Exploring the use cases of GGUF, AWQ, and EXL2 model formats
- Comparing the effects of quantization techniques on model accuracy and inference speed

2. Model Prototyping and Evaluation with LM Studio

- Developing local language model prototypes using LM Studio
- Testing and comparing different language models
- Exposing OpenAI-compatible local API endpoints

3. Enterprise Local LLM Solutions with Ollama

- Creating organization-specific model configurations using Ollama and Modelfile
- Managing versions of customized language models
- Integrating local language models into Python and JavaScript projects
- Deploying Open WebUI with Docker and managing user access
- Designing RAG-based AI assistants that use internal corporate documents