Curso ATDEV230 - Desenvolvimento de Aplicações com IA em Python e Azure AI

Duração: 35h

Introdução

Este curso apresenta os princípios base da Engenharia de IA — a disciplina que constrói sistemas de software com IA na prática. Ao contrário da ciência de dados tradicional, focada em criar e treinar modelos de raiz, a Engenharia de IA foca-se em aplicar e integrar modelos já existentes em aplicações reais. Vai aprender a integrar sem atritos serviços de IA pré-construídos e large language models, como os modelos GPT da OpenAI, nas suas aplicações Python. Vai familiarizar-se com o Microsoft Foundry, o portal unificado da Microsoft para gerir, testar, moderar e publicar modelos de IA. Vai aprender a ancorar large language models nos seus próprios dados, usando vector search, Azure AI Search e o padrão Retrieval Augmented Generation. Terá ainda experiência prática na construção de sistemas de IA agênticos com a Microsoft Agent Framework, alargando-os com skills e aplicando as práticas de engenharia que tornam os agentes fiáveis. Vai também aprender a manter as aplicações de IA seguras e mensuráveis, através de content safety e avaliações, e a publicar, alojar internamente e afinar modelos no Microsoft Foundry. No fim, terá as competências essenciais de Engenharia de IA para integrar capacidades avançadas de IA nas suas soluções, sem precisar de ser cientista de dados.

Público-alvo

Este curso destina-se a programadores profissionais de Python que queiram entrar no mundo da Engenharia de IA através da plataforma de IA da Microsoft, o Microsoft Foundry. Não é um curso para cientistas de dados que queiram construir modelos de IA de raiz ou analisar a fundo o funcionamento dos modelos existentes; é feito para programadores prontos a construir aplicações práticas assentes em IA.

No final do curso

Após a conclusão deste curso, os formandos deverão ser capazes de:
- Utilizar o Microsoft Foundry e os seus serviços de IA pré-construídos.
- Implementar vector search e semantic search com o Azure AI Search.
- Desenvolver aplicações de IA em Python com large language models.
- Aplicar prompt engineering, context engineering e Retrieval Augmented Generation.
- Construir agentes com a Microsoft Agent Framework e integrá-los com MCP.
- Avaliar, proteger e publicar modelos e agentes.

Pré-requisitos

Bom domínio de Python.

Exames de certificação

(não existem exames)

Conteúdo programático em detalhe

O que é a inteligência artificial?
- Definições de inteligência artificial

- Fundamentos de machine learning
- Domínios da inteligência artificial
- Engenharia de IA vs. engenharia de ML
- História, estado atual e futuro
- Panorâmica das frameworks e SDK de IA da Microsoft

Introdução ao Microsoft Foundry

- O que é o Microsoft Foundry?
- Desenvolvimento colaborativo de IA
- Publicar aplicações de IA com projetos
- Gerir recursos e projetos
- LAB: criar um projeto no Microsoft Foundry

Serviços de IA pré-construídos com as ferramentas Foundry

- Panorâmica dos Azure AI Services
- Azure Content Understanding
- Azure AI Speech
- LAB: criar um processador de sinistros assente em IA

Pesquisar por significado com vector search

- Captar significado semântico com embeddings
- Vector search
- Considerações de desenho em vector search
- LAB: usar vector embeddings e pesquisa semântica

Semantic search simplificada com o Azure AI Search

- O que é o Azure AI Search?
- Indexar os seus dados: Blob Storage, Azure SQL, OneLake e outros
- Construir e gerir índices de pesquisa
- Pesquisa por palavra-chave, vetorial e híbrida
- Melhorar a relevância com semantic reranking
- Extrair conclusões com AI enrichment e skillsets
- LAB: construir um índice de conhecimento pesquisável com o Azure AI Search

Fundamentos dos large language models

- Introdução à OpenAI e aos large language models
- O modelo Transformer
- Compreender os tokens
- A API de chat completions do Azure OpenAI
- Gestão de papéis: system, user e assistant
- Estender as capacidades do LLM com function calling
- LAB: construir um copiloto para os caminhos de ferro belgas

Construir sistemas de IA agênticos com a Microsoft Agent Framework

- Compreender o ciclo de raciocínio de um agente
- Princípios do desenho de agentes: modelos, ferramentas e instruções
- Construir agentes com a Microsoft Agent Framework
- Gerir as interações do agente com sessões
- Personalizar agentes de IA com middleware
- Tornar os seus agentes observáveis

• LAB: construir um agente de SQL

Processar diferentes modalidades com LLM

• Introdução aos large language models multimodais
• Visão com modelos GPT: conversar com as suas imagens
• Técnicas de prompting visual
• Processar áudio: speech-to-text e text-to-speech
• LAB: construir uma aplicação de análise visual com comando por voz

Prompt engineering e design patterns

• O que é o prompt engineering?
• Few-shot prompting
• Geração estruturada de queries
• Padrões de orquestração de LLM: prompt chaining, routing e parallelization
• Padrões de orquestração multiagente: group chat, handoff e magentic
• Poupar custos com semantic caching

Context engineering e Retrieval Augmented Generation

• O que é o context engineering?
• O padrão Retrieval Augmented Generation (RAG)
• Ancorar os agentes em dados recuperados
• System prompts contextuais: injetar informação do utilizador e memórias
• Gerir a janela de contexto
• Compactação e sumarização do contexto
• Proteger a KV-cache para evitar custos de tokens descontrolados
• Reduzir alucinações com grounding

Alargar agentes com skills

• O que são as agent skills?
• Skills vs. ferramentas vs. instruções
• Anatomia de uma skill
• Progressive disclosure: carregar contexto a pedido
• Registar skills com o AgentSkillsProvider
• As skills como context engineering escalável
• LAB: construir uma skill reutilizável para um agente

Desenhar agentes eficazes

• Quando usar um agente (e quando não usar)
• Construir ferramentas escaláveis para o seu agente
• Escrever as system instructions do seu agente
• Manter o agente no caminho certo com notas estruturadas
• Deixar o agente delegar trabalho em subagentes
• LAB: construir um agente fiável para concluir tarefas

Manter a IA segura com content safety e guardrails

• Princípios de IA responsável
• Evitar respostas inadequadas do LLM
• Tirar partido do serviço Azure AI Content Safety
• Detetar conteúdo nocivo, jailbreaks e prompt injection
• Implementar guardrails para agentes seguros e fiáveis

• Supervisão humana e controlos de segurança
• LAB: proteger uma aplicação de IA com o Azure AI Content Safety

Avaliar modelos e agentes com o Microsoft Foundry

• Garantir respostas coerentes e relevantes do LLM
• Usar informação correta nas respostas de IA
• Usar os evaluators nativos do Microsoft Foundry
• Implementar avaliações por código com o Azure AI Evaluation SDK
• Medir e melhorar o desempenho da IA
• LAB: avaliar o desempenho de uma aplicação RAG

Conhecimento unificado para agentes com o Foundry IQ

• O que é o Foundry IQ?
• Knowledge bases e knowledge sources
• Recuperação multifonte: Blob Storage, SharePoint, OneLake e a Web
• Agentic retrieval: planeamento de queries, decomposição e reranking
• Recuperação com respeito por permissões, com ACL e identidade do Microsoft Entra
• Ligar knowledge bases a agentes
• Foundry IQ, Fabric IQ e Work IQ
• LAB: construir uma solução de agentic retrieval com o Foundry IQ

Alojar large language models internamente

• Panorâmica dos modelos de linguagem open-source
• Publicar modelos de IA em dispositivos móveis e edge com o ONNX Runtime
• Alojar e publicar modelos de linguagem on-premises e na cloud com o Ollama

Publicar modelos e agentes com o Microsoft Foundry

• Panorâmica do catálogo de modelos
• Benchmarks de modelos
• Deployments standard vs. provisioned throughput vs. managed compute
• Opções de processamento de dados: global, data zone e regional
• Equilibrar custo, latência e residência dos dados
• Colocar agentes em produção com os Foundry Hosted Agents
• Monitorizar modelos e agentes publicados
• LAB: publicar um modelo e um hosted agent no Microsoft Foundry

Model Context Protocol (MCP): normalizar as integrações com LLM

• O que é o Model Context Protocol (MCP)?
• Compreender os componentes do MCP: hosts, clients e servers
• Definir prompts, tools e resources de MCP
• Construir servidores MCP com Python
• LAB: construir servidores e clientes MCP com Python

Afinar modelos de IA com o Microsoft Foundry

• Introdução ao fine-tuning de LLM
• Como decidir entre fine-tuning e RAG?
• Usar dados específicos da tarefa para melhorar o desempenho
• Supervised fine-tuning (SFT) para especialização de domínio
• Alinhar com preferências através de Direct Preference Optimization (DPO)
• Otimizar comportamentos complexos com reinforcement fine-tuning (RFT)

- Destilar modelos grandes em modelos mais pequenos e mais baratos
- Reduzir alucinações com fine-tuning