DATA SCIENCE | ADVANCED | 5 DAYS

MLOps, LLMOps & AI Deployment Engineering (Advanced)

Take your AI from notebooks to production-grade ML & GenAI systems

Why MLOps?

Reliable deployment

Deploy ML and LLM models reliably on Kubernetes.

End-to-end pipelines

Build ML and GenAI pipelines with CI/CD and approvals.

Governance and reuse

Implement feature stores, model registries, and lifecycle controls.

Production monitoring

Monitor classical models and LLMs for performance, drift, and hallucinations.

RAG at scale

Design and deploy RAG systems with vector databases and scalable inference.

What you'll be able to do

- ✓ Design MLOps & LLMOps architectures for cloud, on-prem, or hybrid environments
- ✓ Use MLflow for experiment tracking, model registry & lifecycle management
- ✓ Implement Feast as a feature store for consistent training & inference data
- ✓ Build CI/CD pipelines for ML using GitHub Actions or Azure DevOps
- ✓ Serve models using BentoML and Seldon Core on Kubernetes
- ✓ Orchestrate ML workflows with Kubeflow Pipelines
- ✓ Scale inference with Kubernetes autoscaling and modern serving patterns
- ✓ Monitor models with Prometheus & Grafana, plus LLM observability using tools like Arize Phoenix / LangSmith
- ✓ Deploy a production-grade RAG system with vector DB + embedding model + chat model

Who this is for

- DevOps & Platform Engineers
- Data Engineers & Analytics Engineers
- ML Engineers / Data Scientists moving into production
- Cloud / SRE teams supporting AI workloads
- Fraud / risk / forecasting models (Classical ML)
- AI copilots, RAG-based assistants, and LLM-powered products (GenAI)

Curriculum

01 Foundations of MLOps & AI Deployment

- MLOps vs DevOps vs DataOps vs LLMOps
- Architecture of modern ML & LLM systems
- From experimentation to production: challenges & patterns

02 Experiment Tracking & Model Management (MLflow)

- Tracking metrics, parameters & artifacts
- Model registry: staging, production, rollback
- Integrating MLflow with CI/CD and storage

03 Feature Stores with Feast

- Why feature stores matter in production
- Offline vs online features
- Building and using a feature repository with Feast

04 CI/CD for ML (GitHub Actions / Azure DevOps)

- CI/CD patterns for ML & GenAI
- Automated training, validation & deployment
- Approval workflows and gated promotions

05 Production Model Serving with BentoML & Seldon

- Production-ready model servers vs custom APIs
- Packaging models with BentoML
- Kubernetes-native deployment with Seldon Core
- A/B testing, canary and shadow deployments

06 Pipelines & Orchestration with Kubeflow

- Designing reusable pipeline components
- End-to-end ML workflows
- Scheduling, caching & lineage

07 Scaling Inference on Kubernetes

- Horizontal & event-based autoscaling
- CPU vs GPU inference strategies
- Load testing and performance tuning

08 Monitoring, Observability & LLM Evaluation

- Classical monitoring: latency, errors, data drift
- LLM observability: hallucination detection, RAG quality, trace analysis
- Dashboards and alerts with Prometheus, Grafana & LLM tooling

09 Governance, Security & Lifecycle Management

- Model approvals, access control & audit trails
- Compliance, rollback strategies & long-term lifecycle management

10 Classical MLOps Mini-Capstone

- Build and deploy a traditional ML model with MLflow, Feast & Kubernetes
- Implement CI/CD and monitoring end to end

11 LLMOps & Serving Large Models

- LLM serving patterns: vLLM vs Ollama vs TGI
- Token-per-second optimization & KV cache usage
- Quantization strategies for large models (e.g. 70B) on limited hardware
- Architectures for secure enterprise LLM serving

12 Capstone: Deploy a Production-Grade RAG System

- Teams design and deploy a complete RAG-based AI application:
- Stand up a vector database
- Deploy an embedding model for document indexing
- Serve a chat model (e.g. Llama / Mistral) via vLLM
- Implement RAG retrieval & generation pipeline
- Add monitoring for latency, relevance & hallucinations
- Deploy everything on Kubernetes with autoscaling & dashboards
- This capstone ties together MLOps, LLMOps and platform engineering into one real-world project.