CONCEPT

MLOps and LLMOps: Deploying, Monitoring and Optimising AI Models in Production

26 – 30 April 2027
Dubai - Five-star CBD venue



MLOps and LLMOps: Deploying, Monitoring and Optimising AI Models in Production

Ref: 935_21527 **Date:** 26 – 30 April 2027 **Location:** Dubai - Five-star CBD venue **Fees:** 21450 SAR

Introduction:

MLOps and LLMOps for deploying, monitoring and optimising AI models close the gap between a model that scores well in a notebook and a service that stays accurate, affordable and fast after release. Many teams still retrain by hand, cannot say which dataset produced the live model, notice drift only after users complain and see LLM token bills climb unchecked. This Core Concept course gives engineers a tool-neutral method to version, ship, serve, observe and roll back classical models and LLM applications. Participants design an MLOps and LLMOps Pipeline Blueprint with a monitoring dashboard for a case AI service.

Course Objectives:

- Assess a team's current ML and LLM release process against MLOps maturity levels and the ML Test Score rubric
- Version datasets, features, prompts and model artefacts so that any production prediction can be traced back to its training run
- Build CI/CD and continuous training pipelines that test data, retrain, register and promote models through staged environments
- Select batch, online, streaming or edge serving patterns and configure autoscaling, canary release and rollback for each
- Detect data drift, concept drift and LLM quality regressions with statistical tests, evaluation harnesses and alert thresholds
- Control token spend, latency and unsafe output of LLM services through caching, model routing, guardrails and tracing

Target Audience:

- Machine learning engineers who package, deploy and maintain predictive models behind production APIs
- Data scientists who hand trained models to engineering and must keep them accurate after release
- Platform and infrastructure engineers who run the compute, containers and pipelines that host AI workloads
- LLM application developers responsible for the cost, latency and answer quality of generative features
- Site reliability and observability engineers who add AI services to on-call, alerting and incident routines

Course Outline:

Day 1: Production ML and LLM Lifecycle and Maturity Baseline

- Notebook-to-Service Gap: Hidden Technical Debt in Production ML Systems
- Classical ML Lifecycle Versus LLM Application Lifecycle: Training, Prompting and Retrieval Stages
- MLOps Maturity Levels: Manual Process, Automated Training Pipeline and Full CI/CD Automation
- ML Test Score Rubric for Data, Model, Infrastructure and Monitoring Readiness
- Team Release Process Baseline Audit and Target Maturity Map

Day 2: Reproducibility, Versioning and Pipeline Architecture

- Data, Feature and Code Versioning with Content Hashes and Dataset Snapshots
- Experiment Tracking: Run Metadata, Hyperparameters, Metrics and Artefact Lineage
- Feature Store Design: Offline and Online Stores and Training-Serving Skew
- Model Registry Stages, Model Signatures and Promotion Approval Gates
- Workflow Orchestration with Directed Acyclic Graph Pipelines and Containerised Steps

Day 3: Continuous Delivery, Serving and Scaling of Models

- CI/CD for ML: Data Validation Tests, Model Quality Gates and Build Artefacts
- Continuous Training Triggers: Schedule, Data Volume and Performance Decay
- Batch Scoring, Online Inference APIs, Streaming and Edge Deployment Patterns
- Model Serving Runtimes, GPU Batching, Quantisation and Horizontal Autoscaling
- Shadow Deployment, Canary Release, A/B Testing and Automated Rollback

Day 4: Drift Monitoring, LLM Evaluation and Cost Control

- Data Drift and Concept Drift Detection with DDM, ADWIN and KSWIN Windowing Tests
- LLM Evaluation Harness: Golden Question Sets, Groundedness and Answer Relevance Scoring
- Prompt Version Control, Retrieval Tuning and RAG Chunk and Rerank Optimisation
- Guardrails and Output Filtering: Toxicity, PII Redaction, Schema Checks and Refusal Rules
- Token Budgets, Semantic Caching, Model Routing and Latency Percentile Targets

Day 5: Observability, Incident Response and Pipeline Blueprint Lab

- Distributed Tracing of Retrieval, Prompt and Model Calls with Span-Level Telemetry
- AI Service Incident Runbook: Severity Levels, Rollback to Last Good Model and Post-Incident Review
- Governance Hooks: Model Cards, Lineage Records and Change Approval Logs in the Pipeline
- Lab Build: Monitoring Dashboard for Drift, Quality, Cost and Latency on a Case AI Service
- MLOps and LLMOps Pipeline Blueprint Presentation and Peer Design Review

Skills You Will Gain:

- Model Lineage Tracing
- ML Pipeline Orchestration
- Continuous Training Design
- Inference Serving Optimisation
- Drift Detection Engineering
- LLM Quality Evaluation
- Token Cost Governance
- AI Service Observability

Why Attend This Course:

- Leave with an MLOps and LLMOps Pipeline Blueprint and a working dashboard design for a case AI service
- Stop discovering model decay from user complaints by wiring drift and quality alerts before release
- Cut surprise LLM bills and slow responses with caching, routing and budget limits set per endpoint
- Compare release, serving and on-call practice with engineers running AI in finance, retail, energy, telecoms and public services

Conclusion:

Models and LLM applications keep delivering value only when their data, code, prompts and artefacts move through one traceable, monitored pipeline. The course moves from lifecycle and maturity baselines, through versioning, experiment tracking, registries and orchestration, to continuous delivery, serving and scaling, then drift detection, LLM evaluation, guardrails and cost control. The final day covers tracing, incident runbooks and governance hooks, and builds an MLOps and LLMOps Pipeline Blueprint with a monitoring dashboard for a case AI service.