MLOps and LLM Observability Course

Karthikeyan Vaiyapuri

What you will Learn ?

What you will Learn?

MLOps foundations: end-to-end lifecycle, roles & workflows
CI/CD for ML: automate testing, packaging and deployment pipelines
MLflow essentials: experiment tracking, model & dataset versioning, model registry
Lightweight deployment: expose models through Flask / Streamlit APIs
Large Language Models (LLMs): what they are, common use cases, basic prompt design
LLM observability basics: prompt / response logging, latency & cost metrics, simple health dashboards
Best-practice pipelines: reproducible environments, rollback strategies, minimal governance
Hands-on mini-project: build a small, fully tracked, monitored MLOps workflow you can reuse at work

*

What not to expect ?

What not to expect?

Deep-dive **ML theory** (statistics, advanced algorithms, math proofs)
Hyper-parameter tuning frameworks, AutoML or neural-network architecture search
Enterprise-grade **Kubernetes**, multi-cloud or GPU cluster orchestration
Production-scale **LLM fine-tuning**, RLHF, or model compression techniques
In-depth **security, compliance** or cost-optimization audits
Full-stack **data-engineering tools** (Airflow, Spark, Kafka, feature stores)
Guaranteed coverage of every edge case—focus is on **beginner-friendly, core concepts only**

Day 01

Topics	Session Type	Description	Outcome
Foundation & Introduction	Conceptual Learning	MLOps fundamentals, CI/CD principles, LLM introduction, observability concepts	Understanding of MLOps ecosystem and workflow requirements
	Demo & Hands-on Lab	Basic ML workflow setup, simple model training, Flask API deployment, basic logging implementation	Working ML model API with prediction logging and health checks

Day 02

Topics	Session Type	Description	Outcome
Environment & Model Tracking	Conceptual Learning	Environment management, MLflow architecture, experiment tracking principles, model versioning strategies	Knowledge of MLflow components and version control best practices
	Demo & Hands-on Lab	Anaconda/Docker setup, MLflow installation, experiment tracking, model registry usage, dataset versioning	Complete MLflow environment with tracked experiments and registered models

Day 03

Topics	Session Type	Description	Outcome
Validation & Deployment	Conceptual Learning	Model validation techniques, performance metrics, monitoring strategies, deployment patterns	Understanding of model validation and production deployment considerations
	Demo & Hands-on Lab	Model validation implementation, Flask/Streamlit deployment, API development, metrics collection	Production-ready model serving with comprehensive validation and monitoring

Day 04

Topics	Session Type	Description	Outcome
LLM Observability	Conceptual Learning	LLM observability fundamentals, monitoring tools overview, prompt engineering basics, response quality assessment	Knowledge of LLM-specific monitoring requirements and quality metrics
	Demo & Hands-on Lab	LLM monitoring tools setup, prompt logging implementation, response tracking, inference metrics dashboard	Functional LLM monitoring system with prompt/response tracking

Day 05

Topics	Session Type	Description	Outcome
Best Practices & Integration	Conceptual Learning	MLOps best practices, CI/CD pipeline design, model lifecycle management, production considerations	Comprehensive understanding of production MLOps workflows
	Demo & Hands-on Lab	CI/CD pipeline creation, mini-project integration, end-to-end workflow testing, deployment automation	Complete MLOps pipeline ready for workplace implementation

01

ML Basics & Keywords

Keywords - Core ML Terminology

Model Types

Supervised Learning - Learning with labeled training data

Unsupervised Learning - Finding patterns in unlabeled data

Classification - Predicting categories/classes (e.g., spam/not spam)

Regression - Predicting continuous numerical values (e.g., house prices)

Keywords - Core ML Terminology

Data Concepts

Features - Input variables/attributes used for prediction

Target/Label - The output variable you're trying to predict

Training Data - Data used to teach the model

Test Data - Data used to evaluate model performance

Dataset - Complete collection of data for ML project

Keywords - Core ML Terminology

Model Development Process

Training - Process of teaching the model using training data

Prediction/Inference - Using trained model to make predictions on new data

Model Parameters - Internal settings learned during training

Hyperparameters - Settings you configure before training (e.g., learning rate)

02

What is MLOps?

What is MLOps?

Definition

MLOps = Machine Learning + Operations
Practices for deploying and maintaining ML models in production reliably and efficiently
Bridge between ML development and IT operations

Why MLOps Matters ?

Scale: Deploy models at enterprise level
Reliability: Ensure consistent model performance
Automation: Reduce manual intervention
Collaboration: Align data scientists and engineers

03

MLOps vs Traditional Software Development

MLOps vs Traditional Software Development

Traditional Software	MLOps
Code-centric	Data + Code + Model centric
Deterministic outputs	Probabilistic outputs
Binary success/failure	Performance degradation
Static Functionality	Dynamic model behavior

04

Key MLOps Concepts

Key MLOps Concepts

01
Data Management

- Data versioning and lineage
- Data quality monitoring

02
Model Development

- Experiment tracking
- Model versioning

03
Model Deployment

- Automated deployment pipelines
- A/B testing and rollbacks

04
Monitoring & Observability

- Performance tracking
- Data drift detection

05

MLOps Workflow Overview

MLOps Workflow Overview

Workflow Diagram

Data Collection → Data Preparation → Model Training → Model Validation → Model Deployment → Monitoring → Feedback Loop

Key Stages

Data Pipeline: Collect, clean, validate data
Training Pipeline: Train, validate, test models
Deployment Pipeline: Deploy models to production
Monitoring Pipeline: Track performance and retrain

06

What is CI/CD?

What is CI/CD?

Continuous Integration (CI)

Automated testing of code changes
Frequent integration of code into shared repository
Early detection of bugs and conflicts

Continuous Deployment (CD)

Automated deployment to production
Consistent release process
Rapid delivery of features and fixes

07

CI/CD in Traditional Software vs ML

CI/CD in Traditional Software vs ML

01

Traditional Software CI/CD

- Code testing and deployment
- Binary pass/fail tests
- Immediate rollback capability

02

ML CI/CD Challenges

Data dependencies: Models depend on training data
Model performance: Requires statistical validation
A/B testing: Gradual rollout and comparison
Model decay: Performance degrades over time

08

CI/CD Pipeline for ML Models

CI/CD Pipeline for ML Models

Pipeline Stages

1) **Code Commit**
   - Data scientist pushes model code
2) **Automated Testing**
   - Unit tests for code
   - Data validation tests
   - Model performance tests
3) **Model Training**
   - Automated training on fresh data
   - Model validation and comparison
4) **Deployment**
   - Staging environment testing
   - Production deployment
   - Performance monitoring

09

Introduction to Large Language Models (LLMs)

Introduction to Large Language Models (LLMs)

What are LLMs?

- **Neural networks** trained on massive text datasets
- **Transformer architecture** with billions of parameters
- Capable of **understanding and generating** human-like text
- **Few- shot learning** capabilities

Key Characteristics

Scale: Billions to trillions of parameters
Versatility: Multiple tasks without retraining
Context awareness: Understanding of conversation flow

10

Popular LLMs and Applications

Popular LLMs and Applications

Major LLMs

Model	Developer	Parameters	Key Features
GPT-4	OpenAI	~1T	Test generation, reasoning
Claude	Anthropic	~175B	Helpful, harmless, honest
LLaMA	Meta	7B-65B	Open-source, efficient
Gemini	Google	Variable	Multimodal capabilities

11

LLM Challenges

LLM Challenges

01

Technical Challenges

Computational Requirements: High memory and processing needs
Latency: Response time considerations
Hallucinations: Generating incorrect information
Bias: Reflecting training data biases

02

Operational Challenges

Cost Management: Expensive inference
Scalability: Handling multiple users
Monitoring: Different metrics than traditional ML

12

What is Observability in ML Systems?

What is Observability in ML Systems?

Definition

Ability to understand system behavior from external outputs
Monitoring, logging, and tracing of ML systems
Proactive detection of issues before they impact users

Three Pillars of Observability

Metrics: Quantitative measures of system performance
Logs: Detailed records of system events
Traces: End- to- end request flow tracking

13

Traditional ML vs LLM Observability

Traditional ML vs LLM Observability

Traditional ML Monitoring

Accuracy, precision, recall

Data drift detection

Model performance metrics

LLM-Specific Observability

Prompt quality and relevance

Response coherence and accuracy

Token usage and costs

Latency and throughput

Safety and bias detection

14

Key Metrics to Monitor in ML Systems

Key Metrics to Monitor in ML Systems

01

Performance Metrics

Accuracy: Overall correctness
Latency: Response time
Throughput: Requests per second
Error rates: Failed predictions

01

Data Quality Metrics

Data drift: Changes in input distribution
Feature drift: Changes in feature relationships
Completeness: Missing data detection

03

Business Metrics

User satisfaction: Feedback scores
Cost per prediction: Resource utilization
Model ROI: Business impact measurement

15

LLM-Specific Monitoring Metrics

LLM-Specific Monitoring Metrics

Quality Metrics

Relevance: Response appropriateness
Coherence: Logical consistency
Factuality: Accuracy of information
Safety: Harmful content detection

Operational Metrics

Token consumption: Cost tracking
Cache hit rates: Efficiency optimization
Model switching: Load balancing

User Experience Metrics

Response satisfaction: User ratings
Task completion: Success rates
Engagement: Usage patterns

16

Lab Session Overview

Lab Session Overview

Today's Hands-on Activities

- Environment Setup
  - Install required tools
  - Configure development environment
- Basic MLOps Workflow
  - Create simple ML model
  - Set up basic tracking
  - Implement simple deployment
- LLM Introduction
  - Explore LLM APIs
  - Basic prompt engineering
  - Simple observability setup

17

Lab Prerequisites Check

Lab Prerequisites Check

Required Installations

✓ Python 3.10+
✓ Jupyter Notebook
✓ Git
✓ Basic ML libraries (pandas, scikit- learn)

Knowledge Check

✓ Basic Python programming
✓ Understanding of ML concepts
✓ Familiarity with command line
✓ Jupyter Notebook usage

19

Q&A and Discussion

Thanks

Karthikeyan Vaiyapuri