Syllabus

Model Deployment Strategies

Introduction to Model Deployment

- Importance of model deployment in the machine learning lifecycle
- Challenges in deploying machine learning models in production environments

Batch vs. API vs. Real-time Deployment

- Understanding the differences and use cases for batch, Rest and real-time processing
- When to choose deployment type.

Deployment Strategies for Machine Learning Models

- Batch Deployment: Typical workflow, scheduling, and execution patterns
- API Deployment: Http protocol, REST API
- Real-time Deployment: Streamlining real-time model inference with continuous data input
- Hybrid Deployment: Combining batch and real-time models for complex applications

Syllabus

Understanding Batch Processing

Overview of Batch Processing

- What is batch processing and its relevance in machine learning (ML)?
- Use cases in ML and big data analytics

Implementing Batch Jobs in ML Pipelines

- Build batch processing for ML jobs
- Scheduling batch jobs for large datasets

Performance Optimization

- Optimizing batch processing jobs for large-scale data processing
- Monitoring and scaling batch jobs=

Syllabus

Rest API

Overview of REST Architecture

- REST principles: Statelessness, client-server architecture, and cacheability
- Differences between REST and other API architectures (SOAP, RPC)

Understanding HTTP Protocol and Request Methods

- HTTP request methods: GET, POST, PUT, DELETE, PATCH, and their use cases
- Understanding status codes (2xx, 4xx, 5xx) and response **messages**

Building REST APIs with Flask

- Introduction to Flask as a lightweight framework for REST API development
- Creating endpoints and routing in Flask
- Integrating Flask APIs with ML models for inference

Best Practices from the ML Industry

- Versioning APIs for backward compatibility
- API Error handling

Syllabus

Real-time Processing in ML

Introduction to Real-time Data Streaming

- Importance of real-time data in ML applications
- Key components of a real-time data processing system
- Concepts: Data ingestion, processing, and output in real time

Real-time Processing

- Apache Kafka: Using Kafka for real-time data ingestion and streaming

Designing and Scaling Real-time Pipelines

- Managing stateful and stateless streams
- Ensuring low latency and fault tolerance in real-time systems

Syllabus

Monitoring and Logging

Importance of Monitoring in ML

- Why monitoring is crucial for deployed ML models in production
- Monitoring model performance, accuracy, and drift

Monitoring System Metrics and Logs

- Gathering and visualizing system metrics (e.g., CPU, memory, disk usage)
- Setting up logging for model inference requests, predictions, and errors

Monitoring Tools

- Elasticsearch: Using Elasticsearch for log management and search
- Kibana: Visualizing logs and system metrics with Kibana dashboards

Alerting and Automation

- Setting up automated alerts for model degradation, anomalies, or failures
- Building feedback loops for model retraining based on monitoring results