**Advanced-Data Engineer - Machine Learning in Production**

Duration - 5 Days

Labs: Only tools provided in the training lab will be covered. The rest will be covered as theory.

Cloud labs will be covered only if the participants have access to the cloud.

**Pre-requisites: Intermediate level of Python programming knowledge**

**Day 1: Advanced Data Architecture and Engineering Principles**

Introduction to Advanced Data Engineering

Overview of Data Engineering vs. Data Science

Advanced Data Architecture

Principles Key concepts: Data lakes, data warehouses, and data marts Designing scalable data pipelines

ETL vs. ELT (Extract, Load, Transform)

Data wrangling, cleansing, and transformation techniques

Hands on : EDA Activities – use numpy, pandas , matplotlib and scipy modules

**Day 2: Advanced ETL Processes and Data Transformation**

Complex ETL Pipelines

Best practices for designing and managing ETL pipelines

ETL vs. ELT (Extract, Load, Transform)

Data wrangling, cleansing, and transformation techniques

Parallelism and optimization strategies for ETL processes

Modern Data Engineering Frameworks

Event-driven architectures vs. batch processing

Apache Spark and Eco systems

**Hands-on:**

Build a multi-step ETL pipeline using Apache Airflow

Use Apache Spark for transforming large datasets

**Day 3: Big Data Processing and Cloud Data Engineering**

Scaling Data Processing with Big Data Tools

Introduction to Apache Hadoop and Spark for large-scale data processing

Understanding RDDs, DataFrames, and Datasets in Spark

Optimizing Spark jobs (Caching, Partitioning, Broadcast joins)

Modern Data Engineering Frameworks

Data versioning and schema evolution (e.g., Delta Lake, Apache Iceberg)

**Hands-on:**

Set up a simple Data Lake using AWS S3 and Delta Lake

Design and implement a basic pipeline using Apache Kafka and Apache Flink

Setup Apache Iceberg

**Day 4: Machine Learning Pipelines and Automation**

Integrating Data Engineering with Machine Learning (ML) Pipelines

Overview of MLOps (Machine Learning Operations)

Creating end-to-end data pipelines for ML

Automating feature engineering, data pre-processing, and model deployment

**Hands-on:**

Build an end-to-end data pipeline integrating a simple ML model (using Apache Spark and TensorFlow or Scikit-learn)

Set up automated testing and monitoring for the pipeline

**MLOps**

What is MLOps? Overview of MLOps:

Bridging the gap between Data Science and DevOps

The importance of MLOps in scaling machine learning in production

Key concepts: CI/CD for ML, Automation, Monitoring, Model Governance

MLOps vs DevOps: Whatʼs the difference?

Machine Learning Lifecycle Stages of the ML lifecycle: Data collection, feature engineering, model development, deployment, and monitoring Challenges in productionizing ML models Overview of common ML pipelines (training, validation, deployment)

Tools and Technologies for MLOps

**Hands-on:**

Set up a basic Git repository and workflow for ML model code versioning

Introduction to DVC for versioning data and model artifacts

CI/CD Pipeline for Machine Learning

Integrating with version control and model artifact storage (GitHub, GitLab, S3, MLflow)

**Automating Model Training and Validation**

Automating model training using CI/CD pipelines

Model Deployment Automation Deployment strategies for ML models (A/B testing, Canary releases, Blue-Green deployments)

Streamlit and Flask Application

**Day 5: Model Monitoring, Performance, and Logging**

Monitoring ML Models in Production

Why monitoring is crucial for ML models

Key metrics for monitoring ML models: Latency, Throughput, Model Drift, Accuracy over time

Tools for monitoring: Prometheus, Grafana, ELK stack (Elasticsearch, Logstash, Kibana), Seldon Detecting and managing model drift

Model Logging and Auditing

Importance of logging in MLOps: Experiment tracking, Model versioning, Training data auditing

Tools for model tracking: MLflow,

**Hands-on:**

Set up Prometheus and Grafana to monitor an ML model in production

Use MLflow or Weights & Biases to log and track experiments and models Automation in MLOps Automating data pipelines, feature engineering, and model deployment

Setting up automated alerts for model performance and system health Integrating with monitoring tools (Prometheus, Grafana, CloudWatch)

Build a full MLOps pipeline from data ingestion, model training, deployment, and monitoring.