Course Title:

Traditional AI and ML topics

🕒 **Duration:**

- **Total Duration:** 16 hours
- **Schedule:** 4 hours/day over 4 days

📚 **Course Objectives:**

By the end of this course, participants will be able to:

- Machine Learning Algorithms
- Deep Learning Algorithms
- Building Models using Machine Learning
- Building Models using Deep Learning

👥 **Target Audience:**

- Data Scientist
- Data Engineers
- AI Enthusiasts

Level: Intermediate

Pre-requisites:

- Proficiency in Python

📖 **Course Delivery Methodology:**

- **Hands-on Training:** 70% of the course will be practical, focusing on hands-on labs, demos, simulations, and project work.
- **Theory/Concepts:** 30% of the course will cover theoretical foundations to provide the necessary conceptual understanding.

📖 **Proposed Course Outline:**

Day 1
⏰ **Duration: 4 Hours**

Module 1: Machine Learning for Data Engineers – Foundations & Use Cases

📘 Theory

- What is Machine Learning?
- Categories of ML:
    - Supervised Learning
    - Unsupervised Learning
    - Semi-supervised Learning
    - Reinforcement Learning
- Key Concepts:
    - Bias and Variance
    - Overfitting and Underfitting
    - Generalisation in ML

Descriptive Statistics

- Variance
- Percentile
- Box Plot
- Z-score

Probability Concepts

- Probability Basics

- Probability Distributions

Relationships in Data

- Correlation
- Causation

Statistical Foundations

- Central Limit Theorem
- Various Standard Data Distributions (e.g., Normal, Poisson, Binomial, etc.)
    -
- ML Project Lifecycle:
    - Problem definition
    - Data acquisition and cleaning
    - Exploratory Data Analysis (EDA)
    - Feature engineering
    - Model training and evaluation
    - Deployment considerations
- Dataset Splitting:
    - Train, Test, Validation sets
- Cross-validation Techniques
- Machine Learning for Data Engineers:
    - Overview of relevant use cases and real-world applications

📊 **Case Study**

Use Case: Build an end-to-end ML system using a toy dataset
Goal:

- Learners should be able to identify the right category of ML techniques (Supervised [Classification or Regression], Unsupervised, Semi-supervised) to solve a business problem

- Learners should be able to translate a business problem into a Machine Learning problem

✅ **Hands-on Lab:** *(Optional for this module, or can be planned on Day 2 based on dataset readiness)*

Day 2
⏰ **Duration: 4 Hours**

Module 2: Classical Supervised Learning Algorithms – Classification & Regression

📘 Theory

- **Linear Regression & Logistic Regression**
    - Mathematical intuition and real-world applications
- **Feature Importance & Interpretability Basics**
    - Understanding model influence using coefficients and impurity-based scores
- **Decision Boundaries & Cost Functions**
    - Visualising classification thresholds and loss minimisation
- **Hyperparameter Tuning**
    - Grid Search, Random Search, and Best Practices
- **Decision Trees & Random Forests**
    - Splitting criteria, pruning, ensemble advantages
- **Ensemble Learning**
    - Bagging vs Boosting overview

- **Model Evaluation Metrics**
  - Classification: Accuracy, Precision, Recall, F1-score, ROC-AUC, Confusion Matrix
  - Regression: MSE, RMSE, MAE, R^2 Score *(if needed as extension)*

📊 Case Study / Hands-On Lab

Use Cases:

1. **Regression Task:** Build a regression model to estimate **cloud cost expenses**
2. **Classification Task:** Build a classification model to predict the **probability of data workflow failure** based on relevant pipeline/log attributes

Goal:

- Gain hands-on familiarity with classical ML algorithms
- Confidently apply appropriate models for **classification and regression** tasks
- Interpret results and performance metrics to inform business decisions

---

Day 3
⏰ **Duration: 4 Hours**

Module 3: Unsupervised Learning – Clustering & Dimensionality Reduction

📘 Theory

- **Clustering Techniques:**
  - **K-Means Clustering**
  - **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
- **Using Clustering:**
  - Discovering hidden patterns in large-scale data
  - Use in anomaly detection, grouping behavior, and market segmentation

- **Dimensionality Reduction Techniques:**
  - **PCA (Principal Component Analysis)** – Concept, math, and usage
  - **t-SNE (t-Distributed Stochastic Neighbor Embedding)** – Use for 2D/3D visualization
- **Applications in the Data Engineering Domain:**
  - Data Deduplication (grouping similar records)
  - Data Tagging (auto-labeling based on unsupervised groups)
  - Schema Classification (grouping datasets based on feature structure/metadata)

📊 Case Study / Hands-On Lab

Use Case:
Cluster stores of a supermarket chain (e.g., **Walmart**) based on **store-level and demographic attributes**
Example: income level, store size, footfall, region, etc., using dummy data

Goal:

- Understand how clustering can help in business segmentation
- Identify **store segments** to optimize operations and design **targeted marketing strategies**

✅ **Hands-on:**

- Implement K-Means and DBSCAN
- Visualize clusters
- Compare effects of dimensionality reduction using PCA/t-SNE

Day 4
⏰ **Duration: 4 Hours**

Module 4: Anomaly Detection & Data Quality Monitoring

📘 Theory

- **Statistical Anomaly Detection**
    - Concepts of **Mean** and **Standard Deviation**
    - **Z-Score**, **Box Plot**, and understanding **data distributions**
    - Key statistical indicators to identify outliers and anomalous behavior
- **Density-Based Anomaly Detection**
    - Using **DBSCAN** to detect data points that lie in low-density regions
    - Applicability in identifying rare patterns or faulty data
- **One-Class SVM**
    - Introduction to **One-Class Support Vector Machines**
    - Use in modeling the "normal" class and identifying deviations
- **Isolation Forest**
    - Tree-based approach for detecting anomalies in high-dimensional datasets
    - Comparison with other anomaly detection techniques in terms of scalability and interpretability

📊 Case Study / Hands-On Lab

Use Case:
Build an **alerting ML model** to proactively **identify and raise data quality issues** in a data pipeline
Example: Detect schema drifts, missing values, out-of-range entries, or operational anomalies

Goal:

- Gain proficiency in applying **unsupervised anomaly detection** methods
- Build intelligent monitors to **alert on data quality deviations**
- Use models like **One-Class SVM** and **Isolation Forest** for practical applications in data engineering workflows

✅ **Hands-on:**

- Implement anomaly detection using statistical thresholds and ML models
- Evaluate on synthetic or real-world datasets
- Visualize flagged anomalies and interpret causes

💻 Software and Hardware Requirements

Software Requirements:

- **Programming Languages & Frameworks:**
  - Python 3.x
  - Jupyter Notebook / VS Code
  - TensorFlow

Hardware Requirements:

- Minimum **8 GB RAM** (16 GB recommended)
- **Processor:** Intel i5/i7 or equivalent AMD Ryzen
- **GPU:** (Optional) NVIDIA GPU with CUDA support for deep learning tasks
- **Storage:** Minimum 50GB free space