Artificial Intelligence & Machine Learning Course Syllabus

A Comprehensive, Implementation-Focused Curriculum built on Deep Algorithmic Foundations

Core Focus: Classical ML, PyTorch Foundations, Introductory NLP & Computer Vision

Curriculum Design Overview: This program bridges theoretical machine learning intuition with practical, framework-driven implementation. Moving from scratch implementations using NumPy, students naturally transition into **PyTorch** to design, debug, and evaluate modern Deep Learning models for Natural Language Processing and Computer Vision.

Module 1: Foundations & Classical Machine Learning

Sprint 1: Introduction to ML & Core Vector Mathematics

- **Lesson 1:** What is Machine Learning? Supervised vs. Unsupervised Learning framework, and the structural lifecycle of an ML project.
- **Lesson 2:** Essential Data Operations: Working with NumPy arrays, matrix dimensions, vectorization mechanics, and handling feature/activation vectors.

Sprint 2: Linear Regression (Single Variable)

- **Lesson 1:** Linear Regression mathematical intuition, defining the Cost Function, and the optimization mechanics of Gradient Descent.
- **Lesson 2:** Coding Lab: End-to-end implementation of single-variable Linear Regression from scratch using NumPy.

Sprint 3: Multivariate Regression & Data Preprocessing

- **Lesson 1:** Multivariate Linear Regression, Matrix Vectorization formulas, Feature Scaling (Standardization), and Normalization techniques.
- **Lesson 2:** Coding Lab: Implementing multivariate regression from scratch and analyzing the performance impact of data scaling.

Sprint 4: Advanced Regression & Binary Classification

- **Lesson 1:** Feature Engineering strategies, Polynomial Regression, and foundational Hyperparameter Tuning (Learning Rate diagnostics).
- **Lesson 2:** Transition to Classification: Logistic Regression intuition, Decision Boundaries, and the Logistic Cost Function.

Sprint 5: Logistic Regression & Regularization

- **Lesson 1:** Coding Lab: Pure NumPy implementation of Logistic Regression for robust binary classification.
- **Lesson 2:** The Generalization Problem: Overfitting vs. Underfitting diagnostics, and applying Regularization penalties (L1/L2 weights).

Module 2: Deep Learning Foundations & PyTorch Integration

Sprint 6: Neural Network Intuition & Forward Propagation

- **Lesson 1:** Neural Network Foundations: Biological analogy, layers, hidden units, demand prediction mappings, and forward propagation math.
- **Lesson 2:** Coding Lab: Building a forward propagation inference loop from scratch (e.g., Coffee Roasting model parameters).

Sprint 7: Introduction to PyTorch Framework

- **Lesson 1:** PyTorch Ecosystem Fundamentals: Tensors, autograd engine, computational graphs, CPU/GPU device allocation, and basic tensor manipulations.
- **Lesson 2:** Practical Lab: Converting foundational NumPy neural network code into clean, scalable PyTorch operations.

Sprint 8: Optimization, Multi-Class Models & Training Mechanics

- **Lesson 1:** Multi-class Classification architectures, Softmax regression, and managing numerical roundoff errors with Cross-Entropy.
- **Lesson 2:** Training Loops in PyTorch: Implementing forward passes, computing loss, backpropagation (backward passes), and weight updates via the Adam optimizer.

Sprint 9: Diagnostics, Debugging & Model Selection

- **Lesson 1:** Systematic evaluation workflows: Training, Validation, and Test dataset splitting protocols for architectural model selection.
- **Lesson 2:** Performance Debugging: Diagnosing High Bias vs. High Variance anomalies, plotting learning curves, and tuning regularization parameters.

Sprint 10: Skewed Datasets & ML Ethics

- **Lesson 1:** Advanced evaluation metrics for imbalanced/skewed datasets: Precision, Recall, F1-Score, and precision-recall trade-offs.
- **Lesson 2:** Responsible Machine Learning: Identifying fairness bias, detecting data leakage, and applying baseline ethical mitigation principles.

Module 3: Tree-Based Models & Unsupervised Learning

Sprint 11: Decision Tree Architectures

- **Lesson 1:** Decision Tree fundamentals: Mathematical Entropy, Information Gain criteria, and recursive splitting logic.
- **Lesson 2:** Categorical and Continuous Variables: Handling data via One-Hot Encoding and setting numeric splitting thresholds.

Sprint 12: Tree Ensembles & Boosting

- **Lesson 1:** Ensemble Methods: Bootstrap Aggregation (Bagging), Regression Trees, and Random Forest construction.
- **Lesson 2:** Gradient Boosting Paradigms: XGBoost mechanics, algorithmic advantages, and hyperparameter controls.
- **Coding Lab:** Building, optimizing, and benchmark-comparing Random Forests vs. XGBoost architectures.

Sprint 13: Clustering & Dimensionality Reduction

- **Lesson 1:** K-Means Clustering: Optimization objectives, iterative cluster assignments, and selecting cluster numbers via the Elbow method.
- **Lesson 2:** Principal Component Analysis (PCA): Dimensionality reduction intuition, variance retention, and high-dimensional data visualization.

Sprint 14: Anomaly Detection Systems

- **Lesson 1:** Density Estimation framework: Applying Gaussian (Normal) Distributions and joint probability math for anomaly scoring.
- **Lesson 2:** Coding Lab: Developing a functional anomaly detection pipeline for identifying data outliers or system failures.

Module 4: Applied Domains (Introductory NLP & Computer Vision)

Sprint 15: Foundations of Natural Language Processing (NLP)

- **Lesson 1:** Text Preprocessing Pipelines: Tokenization concepts, stop-word elimination, stemming/lemmatization, and Bag-of-Words vs. TF-IDF numerical vectorization.
- **Lesson 2:** Distributed Representations: Word Embeddings intuition (Word2Vec mapping semantics) and processing text sequences.

Sprint 16: Deep Learning for NLP in PyTorch

- **Coding Lab 1:** Sentiment Analysis: Designing and training a PyTorch model to classify text datasets (e.g., movie or product reviews).
- **Coding Lab 2:** Transition to Sequences: Introduction to sequential layers and a high-level conceptual understanding of the Attention mechanism and Transformers.

Sprint 17: Foundations of Computer Vision (CV)

- **Lesson 1:** Digital Image Basics: Processing image matrices, understanding color channels (RGB), resizing, and structural spatial transformations.
- **Lesson 2:** Convolutional Neural Networks (CNNs): Feature extraction mechanics using filters/kernels, padding, stride, and pooling operations.

Sprint 18: Deep Learning for CV in PyTorch

- **Coding Lab 1:** Image Classification: Developing a custom CNN structure in PyTorch to classify object datasets (e.g., CIFAR-10 or simple geometric shapes).
- **Coding Lab 2:** Object Detection Concepts: Understanding the core difference between classification, localization, and bounding boxes.

Module 5: Capstone Implementation

Sprint 19: Capstone Project Scoping & Design

- **Lesson 1:** Ideation workshop, structuring robust data loaders, establishing performance benchmarks, and selecting appropriate metrics.
- **Lesson 2:** Mentored Development Lab: Setting up PyTorch code architecture, checking shape alignment, and executing baseline model training.

Sprint 20: Optimization & Final Showcase

- **Lesson 1:** Hyperparameter tuning, identifying bottleneck layers, debugging convergence problems, and verifying final metrics.
- **Lesson 2:** Technical Presentation Showcase: Demonstrating model design choices, training logs, results, and structural engineering takeaways.