Machine Learning Course Syllabus

Course Duration: 10 Days (3.5 hours per day, Total 35 hours)

Course Overview

This intensive 10-day course provides a comprehensive foundation in Machine Learning, designed for AI/ML Students. We will cover essential Python tools, fundamental mathematical concepts, core supervised and unsupervised learning algorithms, model evaluation techniques, and an introduction to neural networks. The course emphasizes hands-on application through daily assignments and culminates in a practical project experience.

Daily Schedule

Day 1: Python Basics for ML (NumPy, Pandas, Visualization)

- **Review of Essential Python Programming Concepts:** Data types, control flow, functions, and object-oriented basics relevant to data manipulation.
- **In-depth Exploration of NumPy:** Efficient numerical operations, array computing, array manipulation, broadcasting, and linear algebra operations with NumPy.
- **Mastering Pandas for Data Structures:** Introduction to Series and DataFrames, data loading (CSV, Excel), indexing, selection, handling missing data, data cleaning, merging, and grouping.
- **Effective Data Visualization Techniques:** Using Matplotlib and Seaborn for creating static, interactive, and aesthetically pleasing plots (scatter plots, histograms, box plots, heatmaps, etc.) for exploratory data analysis.
- **Daily Assignment:** Given a raw CSV dataset (e.g., a dataset on housing prices or customer churn), perform data loading, preliminary cleaning (handling missing values, basic type conversions), simple transformations (e.g., creating a new feature from existing ones) using Pandas, and create at least three insightful visualizations (e.g., distribution of a key variable, relationship between two variables, or a categorical breakdown) using Matplotlib/Seaborn.

Day 2: Introduction to Machine Learning & Essential Math (Linear Algebra)

- **Introduction to Machine Learning:** Definition of ML, types of ML (supervised, unsupervised, reinforcement learning), common ML tasks, and the typical ML workflow (data collection, preprocessing, model training, evaluation, deployment).
- **Understanding Data Types and Preprocessing:** Categorical vs. numerical data, feature scaling (standardization, normalization), encoding categorical features (one-hot encoding, label encoding).
- **Foundations of Linear Algebra for ML:** Scalars, vectors, matrices, and tensors. Basic operations: addition, subtraction, scalar multiplication, dot product, matrix multiplication.
- **Key Linear Algebra Concepts:** Transpose, inverse, determinant, eigenvalues, and eigenvectors (conceptual understanding and their relevance to ML algorithms like PCA).

- **Daily Assignment:**
  1. Implement a function in NumPy to perform matrix multiplication for two given matrices.
  2. Given a small dataset, perform Min-Max Scaling and Standardization on a numerical feature without using sklearn preprocessing.

Day 3: Essential Math for ML (Probability & Calculus)

- **Probability Theory for ML:** Basic probability rules, conditional probability, Bayes' Theorem.
- **Common Probability Distributions:** Bernoulli, Binomial, Poisson, Normal (Gaussian) distributions – understanding their properties and applications in ML.
- **Introduction to Calculus for ML:** Limits, continuity, and derivatives. Understanding the concept of a gradient.
- **Partial Derivatives and Chain Rule:** How they are used in multi-variable functions and for backpropagation in neural networks.
- **Optimization Basics:** Introduction to cost functions and the concept of gradient descent as an optimization algorithm.
- **Daily Assignment:**
  1. Given a dataset of coin flips (e.g., 100 flips with 60 heads, 40 tails), calculate the probability of getting heads and tails. Apply Bayes' Theorem to a simple scenario (e.g., probability of a patient having a disease given a positive test result).
  2. Write Python code to compute the derivative of a simple polynomial function (e.g., f(x)=3x2+2x−5) at a given point. Explain how the concept of gradient descent would use this.

Day 4: Supervised Learning (Regression)

- **Introduction to Regression:** Understanding the goal of regression, types of regression problems.
- **Simple Linear Regression:** Model equation, assumptions, and the Ordinary Least Squares (OLS) method for finding coefficients.
- **Multiple Linear Regression:** Extending simple linear regression to multiple independent variables.
- **Polynomial Regression:** Handling non-linear relationships using polynomial features.
- **Model Training and Prediction:** Using scikit-learn for implementing regression models.
- **Regression Evaluation Metrics:** Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.
- **Daily Assignment:** Given a dataset (e.g., California Housing dataset or a car price prediction dataset), implement a Multiple Linear Regression model using scikit-learn. Split the data into training and testing sets, train the model, make predictions, and evaluate its performance using MAE, MSE, and R-squared.

Day 5: Supervised Learning (Classification)

- **Introduction to Classification:** Understanding the goal of classification, binary vs. multi-class classification.
- **Logistic Regression:** A powerful algorithm for binary classification, understanding the sigmoid function and decision boundary.
- **Decision Trees:** Principles of tree-based models, Gini impurity, entropy, and information gain.
- **Support Vector Machines (SVM):** Understanding hyperplanes, margins, and kernel tricks for non-linear classification.
- **K-Nearest Neighbors (K-NN):** Instance-based learning, distance metrics, and the choice of 'k'.
- **Daily Assignment:** Choose a binary classification dataset (e.g., Iris dataset for binary classification, or a dataset on customer churn). Implement and train a Logistic Regression model using scikit-learn. Evaluate its performance using accuracy, precision, recall, and F1-score. Visualize the decision boundary if applicable (for 2D data).

Day 6: Model Evaluation & Optimization (Metrics & Cross-Validation)

- **Advanced Classification Metrics:**
  - **Confusion Matrix:** True Positives, True Negatives, False Positives, False Negatives.
  - **Precision, Recall, F1-Score:** Detailed understanding and trade-offs.
  - **ROC Curve and AUC:** Evaluating classifier performance across different thresholds.
- **Bias-Variance Trade-off:** Understanding underfitting and overfitting.
- **Cross-Validation Techniques:**
  - **Hold-out Validation:** Simple train-test split.
  - **K-Fold Cross-Validation:** Robust evaluation, reducing variance.
  - **Stratified K-Fold:** Maintaining class distribution in folds.
- **Daily Assignment:** Given a classification dataset (e.g., a credit card fraud detection dataset or a medical diagnosis dataset), train a Decision Tree classifier. Evaluate its performance using a confusion matrix, precision, recall, F1-score, and plot the ROC curve with AUC. Implement K-Fold Cross-Validation to get a more robust estimate of the model's performance.

Day 7: Model Evaluation & Optimization (Hyperparameter Tuning & Regularization)

- **Hyperparameter Tuning:**
  - **Manual Tuning:** The iterative process.
  - **Grid Search:** Exhaustive search over a defined parameter grid.
  - **Random Search:** More efficient exploration of the parameter space.
- **Regularization Techniques:**
  - **L1 Regularization (Lasso):** Feature selection and sparsity.
  - **L2 Regularization (Ridge):** Preventing large coefficients, reducing variance.
  - **Elastic Net:** Combination of L1 and L2.

- **Daily Assignment:** Take the regression model from Day 4 (or a new regression dataset). Apply Grid Search or Random Search to find the optimal hyperparameters for a RandomForestRegressor or GradientBoostingRegressor. Then, implement Ridge or Lasso Regression on the same dataset and compare its performance with the simple Linear Regression from Day 4, highlighting the impact of regularization.

Day 8: Unsupervised Learning (Clustering & Dimensionality Reduction)

- **Introduction to Unsupervised Learning:** Discovering patterns in unlabeled data.
- **Clustering Algorithms:**
  - **K-Means Clustering:** Algorithm steps, choosing 'k' (Elbow method).
  - **Hierarchical Clustering:** Dendrograms, agglomerative vs. divisive methods.
- **Dimensionality Reduction:**
  - **Principal Component Analysis (PCA):** Reducing dimensions while retaining variance, understanding eigenvectors and eigenvalues.
- **Daily Assignment:**
  1. Given an unlabeled dataset (e.g., customer segmentation data or image pixel data), apply K-Means clustering. Use the Elbow method to determine an appropriate number of clusters. Visualize the clusters if possible (e.g., using PCA for 2D projection).
  2. Apply PCA to a high-dimensional dataset (e.g., MNIST digits or a gene expression dataset) to reduce its dimensionality to 2 or 3 components. Visualize the data in the reduced dimension.

Day 9: Neural Networks Basics (Perceptrons, Intro to TensorFlow)

- **Introduction to Neural Networks:** Biological inspiration, basic components (neurons, layers, weights, biases).
- **The Perceptron:** The simplest neural network, its activation function, and learning rule.
- **Multi-Layer Perceptrons (MLPs):** Understanding hidden layers, non-linear activation functions (ReLU, Sigmoid, Tanh).
- **Introduction to TensorFlow/Keras:** Setting up a basic neural network, defining layers, compiling the model, training, and evaluation.
- **Daily Assignment:** Build a simple multi-layer perceptron (MLP) using TensorFlow/Keras for a binary or multi-class classification task (e.g., classify images from the Fashion MNIST dataset or a simple synthetic dataset). Experiment with different activation functions and the number of hidden layers. Train the model and evaluate its accuracy.

Day 10: ML Projects (Regression, Classification, Clustering) Presentation

- **Daily Assignment:** Students choose one project topic from the provided "Mini-Capstone Project Topics" list. For the chosen topic, they must outline their approach, including:
  - Problem Statement and Goal.
  - Anticipated Data Requirements (features, target).

    - Proposed ML Algorithm(s).
    - Key Evaluation Metrics.
    - High-level steps for data preprocessing and model training.

Assessment

- Mini-Capstone Project: A one-day project to apply learned skills to a real-world problem.