📘 Week 23: Introduction to Machine Learning + Linear Regression

🔹 Theory

1. Define Machine Learning. Differentiate between Supervised, Unsupervised, Semi-supervised, and Reinforcement Learning.
2. What are the challenges in designing ML systems? Discuss with examples.
3. Compare Batch vs. Online Learning with suitable use cases.
4. Explain the ML life cycle from data collection to model deployment.

🔹 Derivation

5. Derive the Normal Equation for Simple Linear Regression.
6. Prove that the least squares solution minimizes the sum of squared errors.

🔹 Numerical

7. Given a small dataset, compute slope and intercept of the best-fit line manually.
8. Calculate MAE, MSE, RMSE, and R^2 for the given prediction vs. actual values.

📘 Week 24: Gradient Descent

🔹 Theory

1. What are the differences between Batch, Stochastic, and Mini-Batch Gradient Descent?
2. Explain the impact of learning rate in gradient descent.

🔹 Derivation

3. Derive the update rule for parameters in Gradient Descent for linear regression.
4. Show how gradient descent behaves on convex and non-convex loss functions using plots.

🔹 Numerical

5. Perform 3 manual iterations of Gradient Descent for a simple linear regression setup with θ initialized to 0.
6. Analyze convergence with two learning rates (e.g., 0.01 and 0.1).

📘 Week 25: Regression Analysis

🔹 Theory

1. Explain the difference between prediction and inference in regression.
2. Discuss the assumptions of linear regression and their diagnostic checks.

🔹 Derivation

3. Derive the F-statistic formula using ESS, RSS, and TSS.
4. Prove that $R^2 = 1 - \frac{RSS}{TSS}$.

🔹 Numerical

5. Given TSS and RSS, compute R^2, F-statistic, and Adjusted R^2.
6. Interpret p-values and confidence intervals for regression coefficients.

📘 Week 26: Feature Selection

🔹 Theory

1. Differentiate between Filter, Wrapper, and Embedded methods.
2. What are the limitations of correlation-based feature selection?

🔹 Numerical

3. Compute ANOVA F-values for feature selection given dataset split by class.
4. Use VIF values to identify multicollinearity and suggest remedy.

📘 Week 27: Regularization

🔹 Theory

1. Explain bias-variance tradeoff with a diagram.
2. Compare Ridge, Lasso, and ElasticNet regularization methods.

🔹 Derivation

3. Derive the Ridge Regression cost function and its parameter update formula.
4. Show how L1 regularization leads to sparsity.

🔹 Numerical

5. Compute Ridge coefficients manually for given data and λ.
6. Plot coefficient paths for Lasso over varying α values (conceptual/numerical).

📘 Week 28: K-Nearest Neighbors

🔹 Theory

1. How does the choice of K affect bias and variance in KNN?
2. Explain distance metrics used in KNN and their influence on performance.

🔹 Numerical

3. Classify a point using K = 3 from given labeled data.
4. Compute weighted KNN prediction for a test sample.

📘 Week 29: PCA

🔹 Theory

1. What is the curse of dimensionality? How does PCA address it?
2. Compare PCA and SVD with respect to dimensionality reduction.

🔹 Derivation

3. Derive the objective function of PCA using variance maximization.
4. Explain the role of eigenvectors and eigenvalues in PCA.

🔹 Numerical

5. Perform PCA manually on a 2D dataset: standardize, compute covariance, eigen decomposition.
6. Project the original data onto the first principal component.

📘 Week 30: Model Evaluation & Selection

🔹 Theory

1. Explain ROC-AUC curve and its interpretation.
2. Define precision, recall, F1-score and their trade-offs.

🔹 Numerical

3. Given a confusion matrix, compute accuracy, precision, recall, and F1-score.
4. Perform K-fold cross-validation manually on a toy dataset.

📘 Week 31: Naive Bayes

🔹 Theory

1. State and explain the Naive Bayes assumption.
2. Compare Gaussian, Multinomial, and Bernoulli Naive Bayes.

🔹 Derivation

3. Derive Bayes' Theorem and apply it in classification context.

4. Show how log probabilities are used to avoid underflow.

🔹 Numerical

5. Compute class probabilities and predict label using Naive Bayes (with Laplace smoothing).
6. Build spam classifier using word frequency data.

📘 Week 32: Logistic Regression

🔹 Theory

1. Explain the sigmoid function and its role in logistic regression.
2. Interpret coefficients in terms of odds and log-odds.

🔹 Derivation

3. Derive the log-likelihood function for binary logistic regression.
4. Explain how MLE is applied in logistic regression.

🔹 Numerical

5. Perform one iteration of gradient descent for logistic regression.
6. Calculate log loss for a given prediction set.

📘 Week 33: SVM

🔹 Theory

1. Differentiate between hard-margin and soft-margin SVMs.
2. What are kernel functions? Give examples.

🔹 Derivation

3. Derive the optimization problem for hard-margin SVM.
4. Explain the role of KKT conditions in SVM.

🔹 Numerical

5. Identify support vectors from a dataset.
6. Classify a point using polynomial/RBF kernel value computation.

📘 Week 34: Decision Trees

🔹 Theory

1. Explain how CART splits are made using Gini Impurity.
2. What is pruning in decision trees? Why is it important?

🔹 Numerical

3. Calculate Gini Index and Information Gain for a binary split.
4. Build a depth-2 decision tree from scratch.

📘 Week 35: Ensemble Methods

🔹 Theory

1. Explain Bagging and Boosting with real-life analogies.
2. Compare Random Forest and Gradient Boosting.

🔹 Numerical

3. Build a small random forest manually (3 trees).
4. Trace a sample through AdaBoost rounds with weights and errors.

📘 Week 36: Gradient Boosting & XGBoost

🔹 Theory

1. What is the intuition behind gradient boosting?
2. How does XGBoost differ from traditional GBM?

🔹 Derivation

3. Derive the function update rule for classification boosting using log loss.
4. Explain Taylor approximation in XGBoost's objective function.

🔹 Numerical

5. Perform one iteration of gradient boosting using residuals.
6. Compute similarity score and gain for a single XGBoost split.

📦 Clustering (KMeans, DBSCAN, Hierarchical)

🔹 Theory

1. Explain the difference between Partitional, Hierarchical, and Density-based clustering.
2. What is the Elbow Method? How does it help choose K?

Derivation

3. Derive update rules for Lloyd's Algorithm in KMeans.

Numerical

4. Perform 2 iterations of KMeans on a small dataset.
5. Classify core and border points using DBSCAN with ε and MinPts.
6. Build dendrogram and identify optimal number of clusters.