MACHINE LEARNING

Unit II — Supervised Learning: Regression

Topics Covered in this Unit

Introduction to Regression, Need for Regression & Comprehensive Regression vs. Correlation
Taxonomy: Univariate vs. Multivariate, Linear vs. Non-linear, Simple vs. Multiple Regression
Bias–Variance Tradeoff: Mathematical Error Decomposition, Underfitting vs. Overfitting & Sweet Spot
Simple & Multiple Linear Regression: OLS Closed-Form Math, Assumptions & Gradient Descent
Polynomial Regression: Higher-Degree Non-Linear Feature Expansions & Overfitting Pitfalls
Non-Linear Regression: Decision Tree Regression (Variance Reduction) & Random Forest (OOB Error)
Support Vector Regression (SVR): Epsilon-Insensitive Loss Tube, Slack Variables & Kernel Mapping
Regularization Techniques: Ridge (L2 Penalty / Shrinkage) vs. Lasso (L1 Penalty / Sparsity & Selection)
Evaluation Metrics: Mean Squared Error (MSE), MAE, RMSE, R-squared (R2) & Adjusted R-squared
High-Yield University Exam Question Bank with Model Answers, Proofs & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & Data Science Students

COURSE SYLLABUS & MAPPING

[UNIT II SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit II. Verify with your university curriculum mapping.

Unit II Syllabus Breakdown:

Unit II - Supervised Learning-Regression: Introduction to regression, need of regression, regression vs correlation. Types of regression: Univariate vs Multivariate, Linear vs Nonlinear, Simple vs Multiple, Bias–Variance Tradeoff, Overfitting and Underfitting. Regression Techniques: Simple and Multiple Linear Regression; Polynomial Regression; Decision Tree Regression, Random Forest Regression, Support Vector Regression. Regularization Techniques: Ridge Regression (L2); Lasso Regression (L1). Evaluation Metrics: Mean Squared Error (MSE); Mean Absolute Error (MAE); Root Mean Squared Error (RMSE); R-squared (R^2).

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

2.1 Introduction to Regression & Core Concepts

Definition: Regression Analysis

Regression Analysis is a fundamental supervised machine learning technique used to model, estimate, and understand the mathematical relationship between one or more Independent Predictor Variables (Features X in R^d) and a continuous Real-Valued Dependent Target Variable (Outcome y in R): y = f(X) + epsilon, where epsilon represents random noise/residual error.

Need of Regression in Machine Learning

- **1. Numerical Prediction:** Forecasting future continuous outcomes (e.g., house prices, crop yields, stock valuations, energy grid loads).
- **2. Feature Significance & Inferential Modeling:** Quantifying the relative impact and strength of different predictor variables on the target outcome (determining which features are statistically significant).
- **3. Trend Analysis & Extrapolation:** Tracking longitudinal patterns over time to project future growth trajectories.

Exhaustive Comparison: Regression vs. Correlation

Comparison Parameter	Correlation Analysis (Pearson r)	Regression Analysis (y = f(X))
Core Objective	Measures the degree, strength, and direction of a linear association between two variables.	Models the mathematical functional relationship to predict the value of y given X.
Cause-and-Effect Relationship	Purely symmetric; does not imply or establish causality (Correlation does not equal Causation).	Asymmetric; establishes directional dependency (Predictor X influences Target y).
Scale & Boundary Limits	Dimensionless metric bounded strictly between -1.0 and +1.0.	Unbounded; slopes and intercepts reflect the physical measurement units of X and y.
Variable Symmetry	Symmetric: Correlation(X, Y) = Correlation(Y, X).	Asymmetric: Regressing Y on X produces a different equation than regressing X on Y.
Predictive Capability	Cannot make quantitative predictions for new data points.	Explicitly predicts numerical values: y_hat = beta_0 + beta_1 * x.

Taxonomy and Types of Regression

- **1. Univariate vs. Multivariate Regression:** Univariate regression predicts a single scalar target variable y from features. Multivariate regression simultaneously predicts multiple continuous target variables (y_1, y_2, ..., y_k).
- **2. Simple vs. Multiple Regression:** Simple regression involves exactly ONE independent predictor variable X (e.g., Square Footage -> Price). Multiple regression involves TWO OR MORE independent predictor variables (X_1, X_2, ..., X_p -> Price).
- **3. Linear vs. Non-linear Regression:** Linear regression assumes the target is a linear combination of model parameters (beta coefficients). Non-linear regression models complex curved or piecewise relationships (Polynomial, SVR, Decision Trees).

2.2 Bias–Variance Tradeoff, Overfitting & Underfitting

Mathematical Decomposition of Expected Generalization Error

For any machine learning model f_hat(X) trained on dataset D, the expected prediction error on an unseen test point x is mathematically decomposed into three independent components:

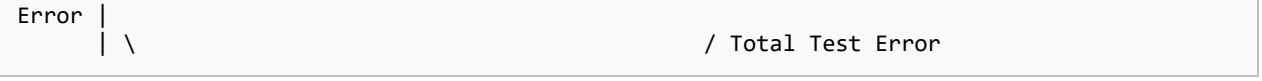
Expected Error = Bias^2 + Variance + Irreducible Error (sigma^2)

- *Bias: Error introduced by erroneous or overly simplistic assumptions in the learning algorithm.*
- *Variance: Error introduced by the model's sensitivity to small random fluctuations in the training set.*
- *Irreducible Error: Inherent noise in the problem domain that cannot be eliminated by any model.*

Underfitting vs. Overfitting

- **1. Underfitting (High Bias, Low Variance):** Occurs when the model is too simple to capture the true underlying structure of the data (e.g., fitting a straight line to a parabolic curve). Results in poor performance on BOTH training data and test data. Solutions: Increase model complexity, engineer polynomial features, reduce regularization penalty lambda.
- **2. Overfitting (Low Bias, High Variance):** Occurs when the model is overly complex and memorizes random noise and idiosyncrasies of the training data (e.g., fitting a 20th-degree polynomial). Results in near-zero training error but terrible test generalization error. Solutions: Apply L1/L2 regularization (Ridge/Lasso), prune decision trees, gather more training data, perform cross-validation.

THE BIAS-VARIANCE TRADEOFF CURVE



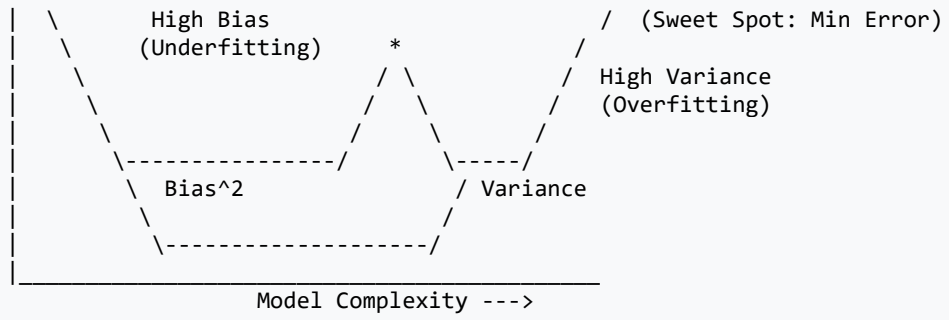


Figure: Optimal model capacity balances Bias^2 and Variance to minimize Total Generalization Error

2.3 Linear Regression: Simple, Multiple & Polynomial

1. Simple Linear Regression (Ordinary Least Squares - OLS)

Simple Linear Regression models the relationship between a single predictor x and continuous target y via the linear equation: y = beta_0 + beta_1 * x + epsilon.

- **Ordinary Least Squares (OLS) Optimization:** Minimizes the Sum of Squared Residuals: RSS = sum((y_i - (beta_0 + beta_1 * x_i))^2).
- **Closed-Form Analytical Solutions (Classic Exam Formulas):**
  - Slope (beta_1): beta_1 = sum((x_i - x_bar) * (y_i - y_bar)) / sum((x_i - x_bar)^2) = Cov(x, y) / Var(x) = r * (sigma_y / sigma_x)
  - Intercept (beta_0): beta_0 = y_bar - beta_1 * x_bar

2. Multiple Linear Regression (Matrix Formulation)

Multiple Linear Regression models y as a linear combination of p features: y = beta_0 + beta_1 * x_1 + beta_2 * x_2 + ... + beta_p * x_p + epsilon.

- **Matrix Representation:** y = X * beta + epsilon, where y is (n x 1), X is design matrix (n x (p+1)) with leading column of 1s, and beta is ((p+1) x 1).
- **The Normal Equation (Closed-Form Solution):** beta = (X^T * X)^(-1) * X^T * y. (Computational complexity: O(p^3) for matrix inversion; computationally expensive when p > 10,000 features).
- **Gradient Descent Optimization:** Iterative weight update rule: beta_j := beta_j - alpha * (1 / n) * sum((f_hat(x_i) - y_i) * x_ij), where alpha is the Learning Rate.

Five Core Assumptions of Classical Linear Regression

- **1. Linearity:** The relationship between predictors X and target y is strictly linear in parameters.

- **2. Independence of Residuals:** Residual errors epsilon_i are mutually independent (no autocorrelation; verified via Durbin-Watson statistic).
- **3. Homoscedasticity:** Residual variance remains constant across all levels of predictor variables (no funnel-shaped fan spreads in residual plots).
- **4. Normality of Residuals:** Errors epsilon are normally distributed with zero mean: epsilon ~ N(0, sigma^2) (verified via Q-Q plots and Shapiro-Wilk test).
- **5. No Multicollinearity:** Predictor features X_i are not linearly correlated with each other (Variance Inflation Factor VIF < 5).

3. Polynomial Regression

Polynomial Regression models non-linear relationships by creating polynomial power combinations of the original features: y = beta_0 + beta_1 * x + beta_2 * x^2 + ... + beta_d * x^d + epsilon. Although it fits a non-linear curve in data space, it remains a LINEAR model mathematically because it is linear with respect to parameters beta.

2.4 Non-Linear Regression Techniques (Trees, Forests, SVR)

1. Decision Tree Regression (CART)

Decision Tree Regression recursively partitions the feature space into orthogonal rectangular regions R_1, R_2, ..., R_M using binary splits. For any query point x falling into region R_m, the prediction is the average of all training responses in R_m: y_hat = (1 / N_m) * sum_{x_i in R_m} y_i.

- **Splitting Criterion (Variance Reduction / Mean Squared Error):** Finds the feature j and split threshold s that minimizes total squared error: min_{j, s} [sum_{x_i in R_L} (y_i - y_bar_L)^2 + sum_{x_i in R_R} (y_i - y_bar_R)^2].
- **Advantages & Limitations:** Non-parametric; naturally models non-linear interactions; requires no feature scaling; highly prone to overfitting (mitigated by Cost-Complexity Pruning via alpha parameter).

2. Random Forest Regression

Random Forest is an ensemble technique that constructs B independent deep, unpruned regression decision trees in parallel using Bootstrap Aggregation (Bagging) and Random Feature Subspace sampling.

- **Prediction Mechanism:** Final prediction is the simple arithmetic mean across all B tree predictions: y_hat = (1 / B) * sum_{b=1}^B T_b(x). Averaging decorrelated trees substantially reduces model variance without increasing bias.

3. Support Vector Regression (SVR)

Unlike ordinary regression which minimizes squared error across all points, SVR uses an epsilon-Insensitive Loss Function (Vapnik, 1995). Errors are penalized ONLY if they exceed a tolerance margin tube of width epsilon around the regression line.

- **SVR Optimization Objective:** min (1 / 2) * ||w||^2 + C * sum(xi_i + xi_i^*) subject to |y_i - (w^T * x_i + b)| <= epsilon + slack. Support vectors are points that lie on or outside the epsilon-tube boundary.
- **Kernel Trick:** Using non-linear kernels (RBF Kernel K(x, z) = exp(-gamma * ||x - z||^2)) enables SVR to fit complex non-linear smooth regression manifolds in high-dimensional spaces.

2.5 Regularization Techniques: Ridge (L2) & Lasso (L1)

Definition: Regularization

Regularization is a technique used to prevent model overfitting and combat multicollinearity by adding an explicit penalty term proportional to model coefficient magnitudes into the cost function, constraining coefficient growth.

1. Ridge Regression (L2 Regularization / Tikhonov)

Ridge regression adds a penalty proportional to the sum of squared weights (L2 norm) to the Ordinary Least Squares loss function.

- **Cost Function:** J(beta) = (1 / n) * sum((y_i - f_hat(x_i))^2) + lambda * sum_{j=1}^p beta_j^2 (where lambda >= 0 is tuning penalty).
- **Closed-Form Solution:** beta_ridge = (X^T * X + lambda * I)^(-1) * X^T * y. (Adding lambda * I guarantees matrix invertibility even under severe multicollinearity!).
- **Behavior:** Smooth coefficient shrinkage toward zero, but coefficients NEVER become exactly zero. Retains all features.

2. Lasso Regression (L1 Regularization — Least Absolute Shrinkage)

Lasso regression adds a penalty proportional to the sum of absolute weight values (L1 norm) to the loss function.

- **Cost Function:** J(beta) = (1 / n) * sum((y_i - f_hat(x_i))^2) + lambda * sum_{j=1}^p |beta_j|.

- **Geometric Intuition & Sparsity:** The L1 constraint region is a diamond-shaped hypercube with sharp corners intersecting coordinate axes. As the OLS elliptical loss contours touch the L1 diamond corners, non-essential feature coefficients are forced to EXACTLY ZERO, performing Automatic Feature Selection!

Master Comparison Table: Ridge vs. Lasso vs. Elastic Net

Comparison Parameter	Ridge Regression (L2)	Lasso Regression (L1)	Elastic Net (L1 + L2)
Penalty Formulation	lambda * sum(beta_j^2) (L2 Norm)	lambda * sum(|beta_j|) (L1 Norm)	lambda_1 * sum(|beta_j|) + lambda_2 * sum(beta_j^2)
Coefficient Shrinkage	Shrinks coefficients smoothly close to 0	Drives non-informative weights to EXACTLY 0	Balances sparsity and smooth shrinkage
Feature Selection	No (Keeps all p features)	Yes (Creates sparse feature subsets)	Yes (Selects groups of correlated features)
Handling Multicollinearity	Excellent (Distributes weights evenly)	Arbitrarily selects one feature, drops others	Best (Groups correlated features together)
Closed-Form Solution	Yes: beta = (X^T X + lambda I)^(-1) X^T y	No analytical solution (Uses Coordinate Descent)	No (Uses Coordinate Descent)

2.6 Evaluation Metrics for Regression

1. Mean Absolute Error (MAE)

Formula: MAE = (1 / n) * sum_{i=1}^n |y_i - y_hat_i|. Measures the average magnitude of absolute errors in the same units as the target. Linear penalty; robust to extreme outliers.

2. Mean Squared Error (MSE) & Root Mean Squared Error (RMSE)

- **Mean Squared Error (MSE):** MSE = (1 / n) * sum_{i=1}^n (y_i - y_hat_i)^2. Squares errors, heavily penalizing large outliers. Differentiable everywhere (ideal cost function for gradient descent).
- **Root Mean Squared Error (RMSE):** RMSE = sqrt(MSE). Retains quadratic penalty on large errors while returning units back to original target scale.

3. Coefficient of Determination (R^2 Score) & Adjusted R^2

- **R-squared (R^2) Score:** Quantifies the proportion of variance in target y explained by features X: R^2 = 1 - [SS_residual / SS_total] = 1 - [sum((y_i - y_hat_i)^2) / sum((y_i - y_bar)^2)] • R^2 = 1.0 indicates perfect prediction; R^2 = 0.0 means model performs no better than simple mean y_bar.
- **Adjusted R-squared (R^2_adj):** Standard R^2 always artificially increases when adding new features, even irrelevant noise. Adjusted R^2 penalizes model complexity: Adjusted R^2 = 1 - [((1 - R^2) * (n - 1)) / (n - p - 1)] (where n is sample count, p is feature count).

2.7 High-Yield University Exam Question Bank

Part A: Short Answer Questions (2 to 5 Marks)

Q1. Explain the Bias-Variance tradeoff with mathematical decomposition. [3-4 Marks]

- **Answer:** Expected Generalization Error = Bias^2 + Variance + Irreducible Error. High Bias leads to Underfitting (overly simple model); High Variance leads to Overfitting (model captures training noise). The optimal model minimizes total error at the sweet spot.

Q2. Why is Adjusted R^2 preferred over standard R^2 in multiple regression? [3 Marks]

- **Answer:** Standard R^2 monotonically increases with every added predictor feature, regardless of relevance. Adjusted R^2 incorporates a penalty for feature count p: R^2_adj = 1 - [(1-R^2)(n-1)/(n-p-1)]. It increases only if the new feature improves the model beyond chance.

Q3. Differentiate between Ridge (L2) and Lasso (L1) regression. [4 Marks]

- **Answer:** Ridge adds L2 squared penalty lambda * sum(beta^2), shrinking weights smoothly near zero without feature elimination. Lasso adds L1 absolute penalty lambda * sum(|beta|), forcing non-essential weights to exactly zero, performing automatic feature selection.

Part B: Long Answer Questions (10 to 15 Marks Outline Guides)

- **Q4. Derive the Ordinary Least Squares (OLS) closed-form solutions for slope beta_1 and intercept beta_0 in Simple Linear Regression. [12 Marks]**
- **Q5. Explain Multiple Linear Regression, its matrix formulation, Normal Equation derivation, and the 5 core assumptions. [12 Marks]**
- **Q6. Discuss Regularization techniques (Ridge, Lasso, Elastic Net) with cost functions, geometric contour visualizations, and comparative analysis. [10 Marks]**
- **Q7. Explain Support Vector Regression (SVR) with epsilon-insensitive tube diagrams, slack variables, and kernel functions. [10 Marks]**

2.8 Unit Summary & Key Takeaways

- Regression models continuous numerical mappings y = f(X) using supervised training data.
- Total Error = Bias^2 (Underfitting) + Variance (Overfitting) + Irreducible Noise.
- OLS Simple Linear Regression minimizes RSS, yielding slope beta_1 = Cov(x, y) / Var(x).
- Multiple Regression Normal Equation: beta = (X^T X)^(-1) X^T y.
- Non-linear models include Polynomial, Decision Tree Regression (variance reduction), and SVR (epsilon-tube loss).
- Ridge (L2) smoothly shrinks weights; Lasso (L1) drives coefficients to zero for sparse feature selection.
- Evaluation metrics include MAE (robust), MSE/RMSE (penalizes outliers), and R^2 / Adjusted R^2.