MACHINE LEARNING

Unit III — Supervised Learning: Classification

Topics Covered in this Unit

Introduction to Classification: Core Concepts, Discrete Output Spaces & Real-World Use Cases
Taxonomy: Binary vs. Multiclass, Balanced vs. Imbalanced Problems (SMOTE & Cost-Sensitive Learning)
Binary Classification: Linear Classification Models, Hyperplanes & Optimal Decision Boundaries
Performance Evaluation: Confusion Matrix, Accuracy, Precision, Recall, Specificity & F1-Score (ROC-AUC)
Multiclass Classification Strategies: One-vs-Rest (OvR) & One-vs-One (OvO) Decomposition Schemes
Multiclass Evaluation: Per-Class Metrics, Macro-Averaging, Micro-Averaging & Weighted-Averaging
K-Nearest Neighbors (KNN): Distance Metrics, Choice of k Parameter & Lazy Learning Trade-offs
Support Vector Machines (SVM): Hard Margin (Maximum Margin 2/||w||) vs. Soft Margin (Slack Variables C)
Non-Linear SVM & The Kernel Trick: RBF (Gaussian) Kernel, Polynomial Kernel & Sigmoid Kernel
High-Yield University Exam Question Bank with Step-by-Step Numerical Calculations & 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 III SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

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

Unit III Syllabus Breakdown:

Unit III - Supervised Learning-Classification: Introduction to classification, need of classification. Types of Classification: Binary and Multiclass, Binary vs. Multiclass Classification, Balanced and Imbalanced Classification Problems. Binary Classification: Linear classification model, decision boundary. Performance Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1-Score. Multiclass Classification: One-vs-One and One-vs-All classification techniques, multiclass confusion matrix; Per-Class Precision and Per-Class Recall; Macro, Micro and Weighted Averaging Methods. Classification Algorithms: K-Nearest Neighbors (KNN), Linear Support Vector Machine (SVM), Soft Margin SVM. Kernel Functions in SVM: Radial Basis Function (RBF/Gaussian) Kernel, Polynomial Kernel, Sigmoid Kernel.

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

3.1 Introduction to Classification & Taxonomies

Definition: Classification

Classification is a supervised machine learning task where the target output variable y is a discrete categorical category (Class Label): f: X in R^d -> y in {C_1, C_2, ..., C_K}. The algorithm learns an optimal Decision Boundary (separating surface) that partitions the feature space into distinct decision regions corresponding to each class.

Types of Classification Problems

- **1. Binary Classification:** The target variable has exactly TWO mutually exclusive classes (e.g., y in {0, 1} or {-1, +1}). Examples: Email Spam vs Ham, Loan Approval (Yes/No), Disease Diagnosis (Positive/Negative).
- **2. Multiclass Classification:** The target variable has THREE OR MORE mutually exclusive classes (e.g., y in {0, 1, ..., K-1}). Each sample belongs to exactly one class. Examples: Handwritten Digit Recognition (0 to 9), Vehicle Type (Car, Truck, Motorcycle, Bus).
- **3. Balanced vs. Imbalanced Classification:**
  - Balanced Datasets: All target classes have roughly equal representation in the training data (e.g., 50% Class A, 50% Class B).
  - Imbalanced Datasets: Severe class disparity (e.g., Credit Card Fraud where 99.9% of transactions are legitimate and 0.1% are fraudulent). Traditional accuracy fails (Accuracy Paradox: a dummy model predicting 'Legitimate' achieves 99.9% accuracy but catches 0% of fraud!).
  - Mitigation Strategies: (a) Resampling: SMOTE (Synthetic Minority Over-sampling Technique) generating synthetic samples, (b) Cost-Sensitive Learning: Assigning higher loss penalties to minority class misclassifications, (c) Evaluating with F1-Score / PR-AUC instead of accuracy.

Binary Linear Classification & Decision Boundaries

In a binary linear classification setting, a Hyperplane in d-dimensional space serves as the decision boundary: w^T * x + b = 0 (where w in R^d is the normal weight vector and b is the bias scalar).

- **Decision Rule:** y_hat = sign(w^T * x + b) = +1 if (w^T * x + b >= 0), else -1 if (w^T * x + b < 0).

3.2 Performance Evaluation of Classification Models

1. The Confusion Matrix

The Confusion Matrix is a tabular layout that cross-tabulates predicted class labels against true ground-truth labels for a binary classification problem:

Confusion Matrix	Predicted POSITIVE (+1)	Predicted NEGATIVE (-1)
Actual POSITIVE (+1)	True Positive (TP) — Correct hit	False Negative (FN) — Miss / Type II Error
Actual NEGATIVE (-1)	False Positive (FP) — False alarm / Type I Error	True Negative (TN) — Correct rejection

2. Core Performance Evaluation Metrics (Formulas & Real-World Trade-Offs)

- **1. Classification Accuracy:** Accuracy = (TP + TN) / (TP + TN + FP + FN). Measures overall percentage of correct predictions. Misleading under severe class imbalance.
- **2. Precision (Positive Predictive Value):** Precision = TP / (TP + FP). Out of all samples predicted as Positive, what fraction was truly Positive? High precision is critical when False Positives are costly (e.g., Spam Filter: marking an important client email as spam is unacceptable).
- **3. Recall (Sensitivity / True Positive Rate - TPR):** Recall = TP / (TP + FN). Out of all actual Positive samples in reality, what fraction did the model successfully detect? High recall is critical when False Negatives are fatal (e.g., Cancer Detection / Fraud: missing a malignant tumor has disastrous consequences).
- **4. Specificity (True Negative Rate - TNR):** Specificity = TN / (TN + FP). Out of all actual Negatives, what fraction was correctly rejected?
- **5. F1-Score (Harmonic Mean of Precision & Recall):** F1-Score = 2 * ((Precision * Recall) / (Precision + Recall)) = (2 * TP) / (2 * TP + FP + FN). Gives balanced metric when precision and recall must be traded off. (Harmonic mean severely penalizes extreme imbalances, e.g., Precision=1.0, Recall=0.0 -> F1=0.0).

3.3 Multiclass Classification & Averaging Methods

1. Decomposition Techniques: One-vs-Rest vs. One-vs-One

Strategy	Number of Binary Classifiers Trained	Training Methodology	Prediction Decision Rule
One-vs-Rest (OvR / One-vs-All)	K classifiers (where K = number of classes)	Classifier i is trained to separate Class i (Positive) from ALL other classes combined (Negative).	y_hat = argmax_i f_i(x) (Class with highest confidence/probability score).
One-vs-One (OvO)	K * (K - 1) / 2 classifiers	A dedicated binary classifier is trained for	Majority Voting: Each classifier votes for one class;

Strategy	Number of Binary Classifiers Trained	Training Methodology	Prediction Decision Rule
		every distinct pair of classes (Class i vs Class j).	the class with the most votes wins.

2. Multiclass Averaging Schemes (Macro, Micro, Weighted)

- **Per-Class Metrics:** For Class i in a K-class problem: Precision_i = TP_i / (TP_i + FP_i), Recall_i = TP_i / (TP_i + FN_i).
- **1. Macro-Averaging:** Computes the unweighted arithmetic mean of per-class metrics: Macro-Precision = (1 / K) * sum_{i=1}^K Precision_i. Treats every class equally; highlights performance on rare minority classes.
- **2. Micro-Averaging:** Globally aggregates all true positives, false positives, and false negatives across all classes before calculating the metric: Micro-Precision = sum(TP_i) / [sum(TP_i) + sum(FP_i)]. Dominated by majority classes.
- **3. Weighted-Averaging:** Weights each per-class metric by its actual class support (number of ground-truth instances N_i): Weighted-Precision = sum_{i=1}^K ((N_i / N_total) * Precision_i).

3.4 K-Nearest Neighbors (KNN) Classifier

Definition: K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based, Lazy Learning algorithm. It makes no explicit assumptions about the underlying data distribution and performs zero explicit training; all computation is deferred until inference time when classifying a query test instance.

The KNN Algorithm Mechanics

- **1. Distance Calculation:** Compute distance between query point x and ALL training points in dataset using Euclidean distance d(x, x_i) = sqrt(sum((x_j - x_ij)^2)).
- **2. Neighbor Selection:** Identify and sort the k training points with smallest distances to x.
- **3. Voting Rule:**
  - Simple Majority Voting: Assign x to the most frequent class among the k neighbors.
  - Distance-Weighted Voting: Weight neighbor votes inversely by distance w_i = 1 / d(x, x_i)^2 (closer neighbors have greater influence).

Impact of Hyperparameter k

- • **Small k (e.g., k = 1):** Extremely complex, jagged decision boundaries. Low bias, High variance (overfitting; highly sensitive to outlier noise).
- • **Large k (e.g., k = 100):** Very smooth, rigid decision boundaries. High bias, Low variance (underfitting; minority classes swamped by majority classes).
- • **Optimal k:** Chosen via Cross-Validation (typically an odd integer to break ties in binary classification: k = 3, 5, 7).

3.5 Support Vector Machines (SVM) & Kernel Functions

Definition: Support Vector Machine (SVM)
Support Vector Machine (SVM) is a powerful geometric classification algorithm based on Statistical Learning Theory (Vapnik). It seeks the Optimal Separating Hyperplane that maximizes the geometric Margin (distance between the decision boundary and the closest training points of each class).

1. Linear Hard Margin SVM (Linearly Separable Data)

Given linearly separable training data y_i in {-1, +1}, the two bounding margin hyperplanes are defined as: w^T * x + b = +1 (for Class +1) and w^T * x + b = -1 (for Class -1).

- **Margin Width:** The total geometric distance between the two margin hyperplanes is: Margin = 2 / ||w||.
- **Optimization Problem:** Maximizing Margin (2 / ||w||) is mathematically equivalent to minimizing ||w||^2 / 2: min_{w, b} (1 / 2) * ||w||^2 subject to y_i * (w^T * x_i + b) >= 1 for all i = 1, ..., n.
- **Support Vectors:** The critical subset of training points lying exactly on the margin hyperplanes (where y_i * (w^T x_i + b) = 1). Removing all other data points leaves the decision boundary completely unchanged!

2. Soft Margin SVM (Linearly Non-Separable Data)

When real-world data contains noise or overlap, a hard margin is impossible. Soft Margin SVM (Cortes & Vapnik, 1995) introduces Slack Variables xi_i >= 0 allowing controlled margin violations.

- **Soft Margin Optimization Formulation:** min_{w, b, xi} (1 / 2) * ||w||^2 + C * sum_{i=1}^n xi_i subject to y_i * (w^T * x_i + b) >= 1 - xi_i.
- **Role of Hyperparameter C (Regularization Parameter):**
  - Large C: Imposes massive penalty on errors. Strict margin, fewer violations; risk of overfitting.

- Small C: Tolerates more misclassifications and margin violations in exchange for a wider, smoother margin; risk of underfitting.

3. Kernel Functions in SVM (Non-Linear Classification)

When data is non-linearly separable in the original input space R^d, the Kernel Trick maps input vectors into a higher-dimensional Hilbert feature space H: phi: R^d -> H where the data becomes linearly separable. The beauty of the kernel trick is that the inner product K(x, z) = <phi(x), phi(z)> is computed directly in low-dimensional space without explicitly evaluating phi(x)!

Kernel Name	Mathematical Formula K(x, z)	Key Hyperparameters & Characteristics
Linear Kernel	K(x, z) = x^T * z + c	Used when data is already linearly separable or feature count d >> sample count n (e.g., Text Classification).
Polynomial Kernel	K(x, z) = (gamma * x^T * z + c)^d	Degree d controls non-linear curve flexibility. Models feature cross-product interactions.
Radial Basis Function (RBF / Gaussian)	K(x, z) = exp(-gamma * ||x - z||^2)	Most popular general-purpose kernel. Maps into infinite-dimensional space. Gamma controls radius of influence (High gamma = tight boundary, overfitting).
Sigmoid Kernel	K(x, z) = tanh(alpha * x^T * z + c)	Derived from neural network activation functions. Models multi-layer perceptron behavior.

3.6 High-Yield University Exam Question Bank

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

Q1. What is the difference between Precision and Recall? When is Precision prioritized over Recall? [3 Marks]

- **Answer:** Precision = TP/(TP+FP) measures exactness (fraction of predicted positives that are true). Recall = TP/(TP+FN) measures completeness (fraction of actual positives detected). Precision is prioritized when False Positives carry severe costs (e.g., spam filtering, fraud accusation).

Q2. Explain the Kernel Trick in Support Vector Machines. [3 Marks]

- **Answer:** The Kernel Trick allows SVM to operate in an infinite or high-dimensional feature space without explicitly computing coordinates phi(x). It replaces dot products with a kernel function K(x, z) = <phi(x), phi(z)>, enabling non-linear classification with low computational complexity.

Q3. Differentiate between One-vs-Rest (OvR) and One-vs-One (OvO) multiclass strategies. [3 Marks]

- **Answer:** For K classes, OvR trains K binary classifiers (Class i vs all others combined) and predicts via max confidence. OvO trains K(K-1)/2 binary classifiers (Class i vs Class j for every pair) and predicts via majority voting.

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

- **Q4. Given a Confusion Matrix [TP=80, FN=20, FP=10, TN=90], calculate Accuracy, Precision, Recall, Specificity, and F1-Score step-by-step. [10 Marks]**
- **Q5. Formulate Linear Hard Margin and Soft Margin SVM optimization problems. Explain the roles of margin width 2/||w||, slack variables xi_i, and penalty parameter C. [12 Marks]**
- **Q6. Explain the K-Nearest Neighbors (KNN) algorithm in detail. Discuss distance metrics, the effect of k on bias-variance, and distance-weighted voting. [10 Marks]**
- **Q7. Discuss Kernel Functions in SVM (Linear, Polynomial, RBF, Sigmoid) with mathematical formulas and geometric interpretation of decision boundaries. [10 Marks]**

3.7 Unit Summary & Key Takeaways

- Classification maps feature vectors X to discrete categorical class labels y.
- Performance evaluation relies on the Confusion Matrix: Precision (exactness), Recall (completeness), and F1-Score (harmonic mean).
- Multiclass classification uses OvR (K classifiers) or OvO (K(K-1)/2 classifiers) with Macro/Micro/Weighted averaging.
- KNN is a non-parametric lazy learner classifying via majority vote of k nearest metric neighbors.
- SVM finds the Maximum Margin Hyperplane (Margin = 2/||w||) supported by boundary Support Vectors.
- Soft Margin SVM introduces slack variables and regularization parameter C for non-separable data.
- The Kernel Trick (RBF, Polynomial, Sigmoid) projects non-linear data into high-dimensional space where linear separation is possible.