

**Sri Sivasubramaniya Nadar College of Engineering, Chennai**  
 (An Autonomous Institution Affiliated to Anna University)

Degree & Branch	B.E. Computer Science & Engineering	Semester VI
Subject Code & Name	UCS2612 – Machine Learning Algorithms Laboratory	
Academic Year	2025–2026 (Even)	Batch 2023–2027

#### **Experiment 4: Binary Classification using Linear and Kernel-Based Models**

## **Objective**

To classify emails as spam or ham using Logistic Regression and Support Vector Machine (SVM) classifiers and to analyze the effect of hyperparameter tuning on classification performance.

## **Dataset**

The **Spambase** dataset contains numerical features extracted from email content and a binary label indicating spam or non-spam (ham).

### **Dataset Links (for reference):**

- Kaggle: <https://www.kaggle.com/datasets/somesh24/spambase>

## **Theory Background**

### **1. Logistic Regression**

Logistic Regression is a probabilistic classification algorithm used for binary classification problems. It models the probability that a sample belongs to a particular class using the sigmoid function:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

A threshold (usually 0.5) is applied to convert probability into class labels.

**Loss Function:** Logistic Regression minimizes the **log-loss (cross-entropy loss)**, which penalizes incorrect predictions.

### **Regularization in Logistic Regression**

Regularization prevents overfitting by adding a penalty to large coefficients.

- **L1 Regularization (Lasso):** Encourages sparsity by shrinking some coefficients exactly to zero. Useful for feature selection.
- **L2 Regularization (Ridge):** Penalizes large weights but keeps all features. Improves model generalization.

## Logistic Regression Hyperparameters

- **C (Inverse Regularization Strength):** Controls the trade-off between model complexity and regularization.
  - Small  $C$ : Strong regularization, simpler model
  - Large  $C$ : Weak regularization, complex model
- **Solver:**
  - `liblinear`: Suitable for small datasets; supports L1 and L2
  - `saga`: Efficient for large datasets; supports L1 and L2

## 2. Support Vector Machine (SVM)

Support Vector Machine is a margin-based classifier that finds an optimal hyperplane separating two classes by maximizing the margin between them.

**Key Idea:** Only a subset of training points (support vectors) define the decision boundary.

### SVM Kernels

- **Linear Kernel:** Suitable for linearly separable data
- **Polynomial Kernel:** Captures polynomial relationships
- **RBF Kernel:** Handles complex, non-linear boundaries
- **Sigmoid Kernel:** Similar to neural network activation

### SVM Hyperparameters

- $C$ : Controls margin vs misclassification
  - Small  $C$ : Wider margin, higher bias
  - Large  $C$ : Narrow margin, lower bias
- $\gamma$ : Controls influence of a single training point

## Hyperparameter Tuning

### Grid Search

Grid Search exhaustively evaluates all combinations of predefined hyperparameters using cross-validation.

### Randomized Search

Randomized Search evaluates randomly sampled hyperparameter combinations and is computationally efficient.

**Note:** Students may use either method. If both are used, results must be compared.

## Implementation Steps

1. Load the dataset.
2. Preprocess the data:
  - Handle missing values
  - Standardize features
3. Perform Exploratory Data Analysis (EDA).
4. Split the dataset into training and testing sets.
5. Train baseline Logistic Regression.
6. Tune Logistic Regression hyperparameters.
7. Train SVM with different kernels.
8. Tune SVM hyperparameters.
9. Evaluate models using standard metrics.
10. Perform 5-Fold Cross-Validation.

## Hyperparameter Search Space

### Logistic Regression

- Regularization: L1, L2
- $C \in \{0.01, 0.1, 1, 10, 100\}$
- Solver: liblinear, saga

### Support Vector Machine

- Kernel: Linear, Polynomial, RBF, Sigmoid
- $C \in \{0.1, 1, 10, 100\}$
- $\gamma \in \{\text{scale}, \text{auto}\}$
- Degree (Polynomial): {2, 3, 4}

## Hyperparameter Tuning Results

Model	Search Method	Best Parameters	Best CV Accuracy
Logistic Regression	Grid / Random		
SVM	Grid / Random		

## Logistic Regression Performance

Metric	Value
Accuracy	
Precision	
Recall	
F1 Score	
Training Time (s)	

## SVM Kernel-wise Performance

Kernel	Accuracy	F1 Score	Training Time (s)
Linear			
Polynomial			
RBF			
Sigmoid			

## K-Fold Cross-Validation Results (K = 5)

Fold	Logistic Regression	SVM
Fold 1		
Fold 2		
Fold 3		
Fold 4		
Fold 5		
Average		

## Comparative Analysis

### Observations

- Identify the best-performing classifier.
- Explain the impact of regularization.
- Discuss kernel behavior in SVM.
- Comment on bias-variance trade-off.

Criterion	Logistic Regression	SVM
Accuracy		
Model Complexity	Low	High
Training Time	Low	High
Interpretability	High	Low

## Learning Outcomes

- Understand probabilistic and margin-based classifiers.
- Apply hyperparameter tuning.
- Evaluate classification models.
- Interpret experimental results.

## References

- Scikit-learn: Logistic Regression
- Scikit-learn: Support Vector Machines
- Scikit-learn: Hyperparameter Optimization
- Spambase Dataset – Kaggle
- UCI ML Repository – Spambase