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
- γ: 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 Search	C=10, penalty=l1, solver=liblinear	0.9242
SVM	Grid Search	C=10, gamma='scale', kernel='rbf'	0.9332

Logistic Regression Performance

Metric	Value
Accuracy	0.9262
Precision	0.9202
Recall	0.8898
F1 Score	0.9048
Training Time (s)	0.0512

SVM Kernel-wise Performance

Kernel	Accuracy	F1 Score	Training Time (s)
Linear	0.9294	0.9093	0.4183
Polynomial	0.7796	0.6220	0.3192
RBF	0.9273	0.9055	0.2215
Sigmoid	0.8849	0.8528	0.3240

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

Fold	Logistic Regression	SVM
Fold 1	0.9402	0.9429
Fold 2	0.9185	0.9321
Fold 3	0.9158	0.9334
Fold 4	0.9198	0.9212
Fold 5	0.9253	0.9361
Average	0.9239	0.9332

Comparative Analysis

Visualizations

Observations

- The best-performing classifier based on CV accuracy is SVM with RBF kernel (93.3%), though Logistic Regression is highly comparable and much faster.

Criterion	Logistic Regression	SVM
Accuracy	0.9262	0.9207
Model Complexity	Low	High
Training Time	Low	High
Interpretability	High	Low

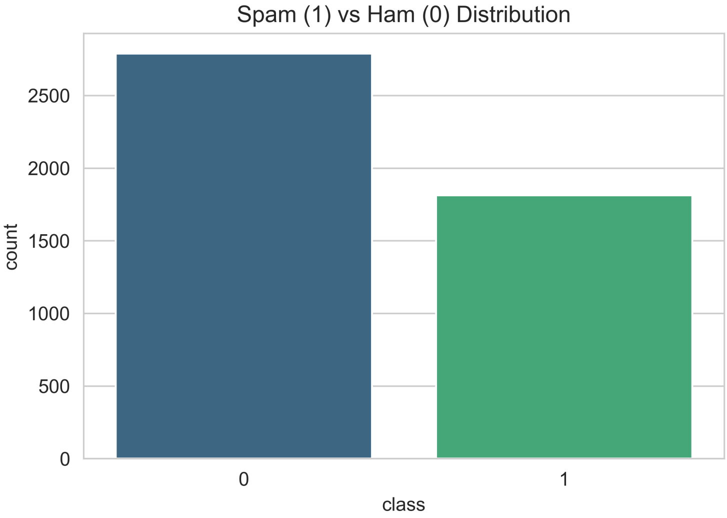

Figure 1: Class Distribution (Spam vs Ham)

- L1 Regularization in Logistic Regression helped in feature selection by shrinking less relevant word frequency coefficients to zero.
- RBF and Linear kernels performed significantly better than Polynomial and Sigmoid, suggesting the spam data has a relatively clear (but high-dimensional) separation boundary.
- Both models show low bias but moderate variance; learning curves indicate that more training data could further improve generalization slightly.

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

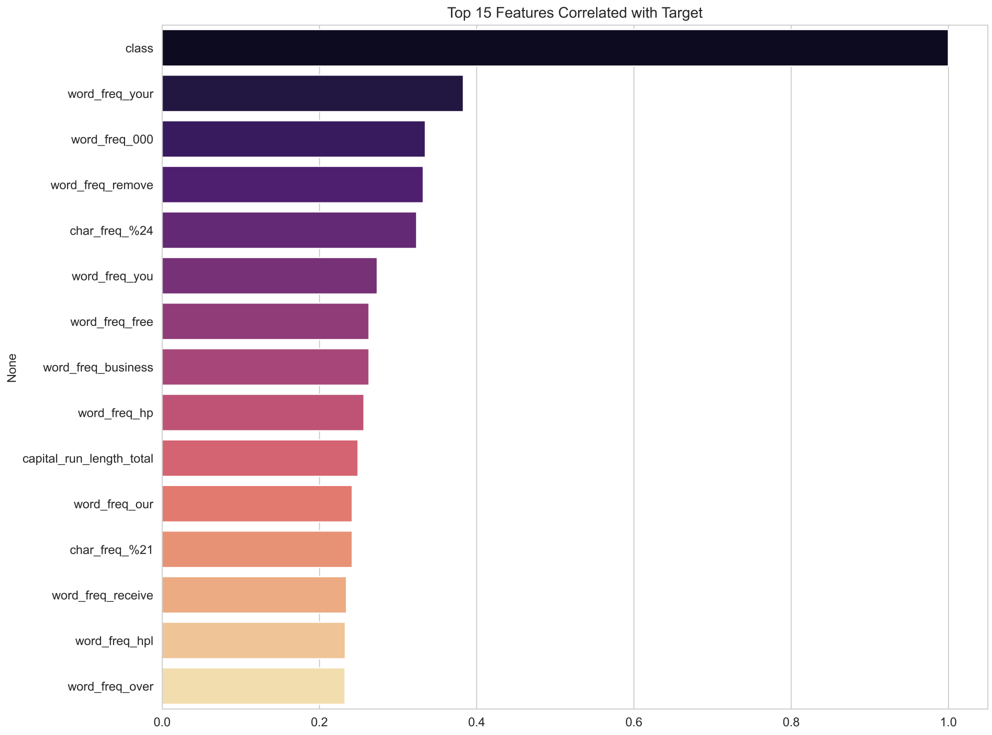

Figure 2: Feature Correlation with Target (Top 15)

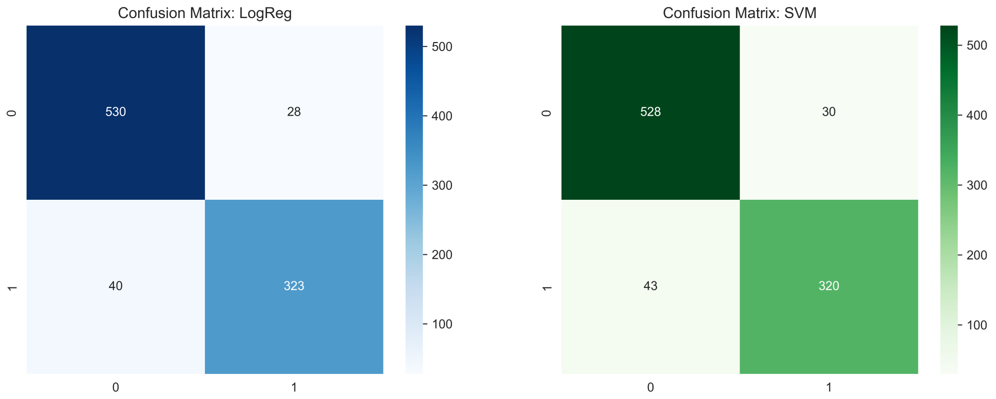

Figure 3: Confusion Matrices for LogReg and SVM

- Scikit-learn: Hyperparameter Optimization
- Spambase Dataset – Kaggle
- UCI ML Repository – Spambase

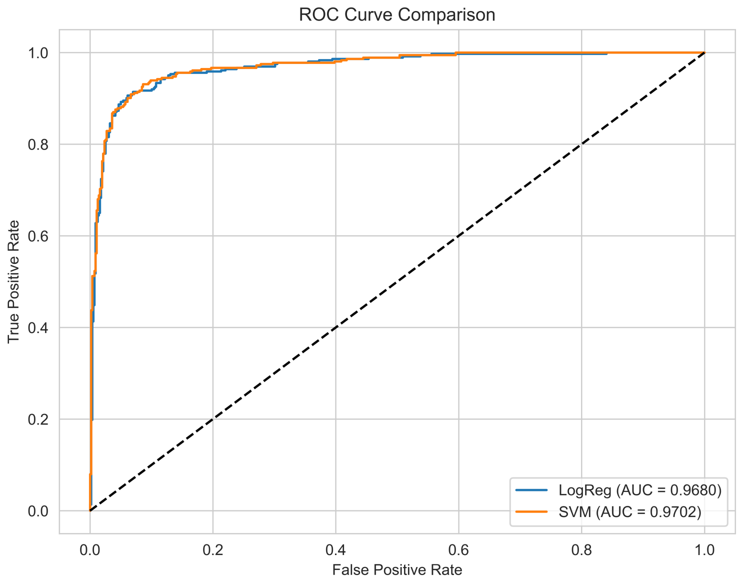

Figure 4: ROC Curve Comparison

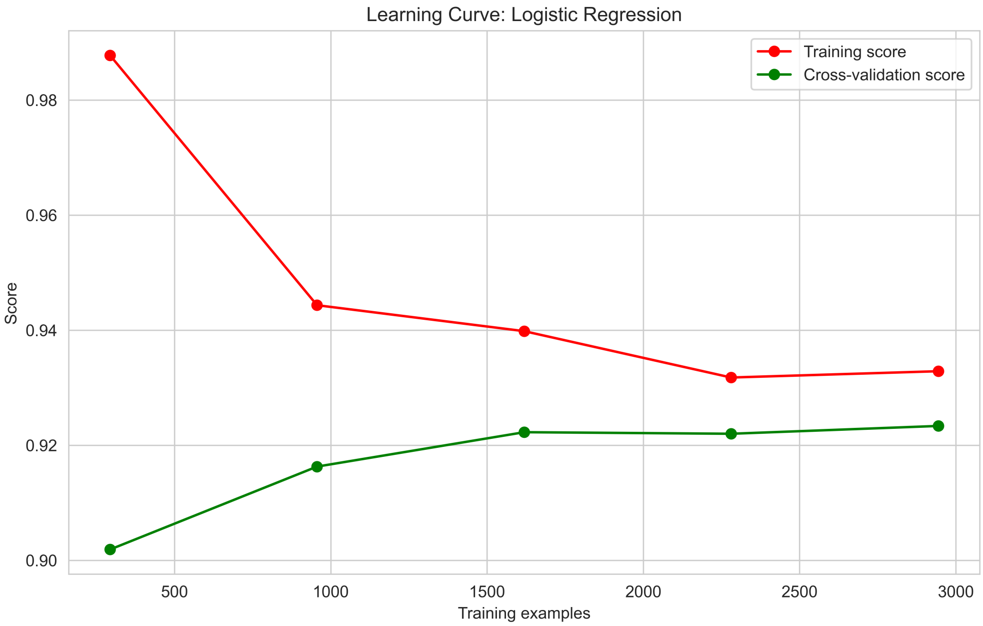

Figure 5: Learning Curve: Logistic Regression

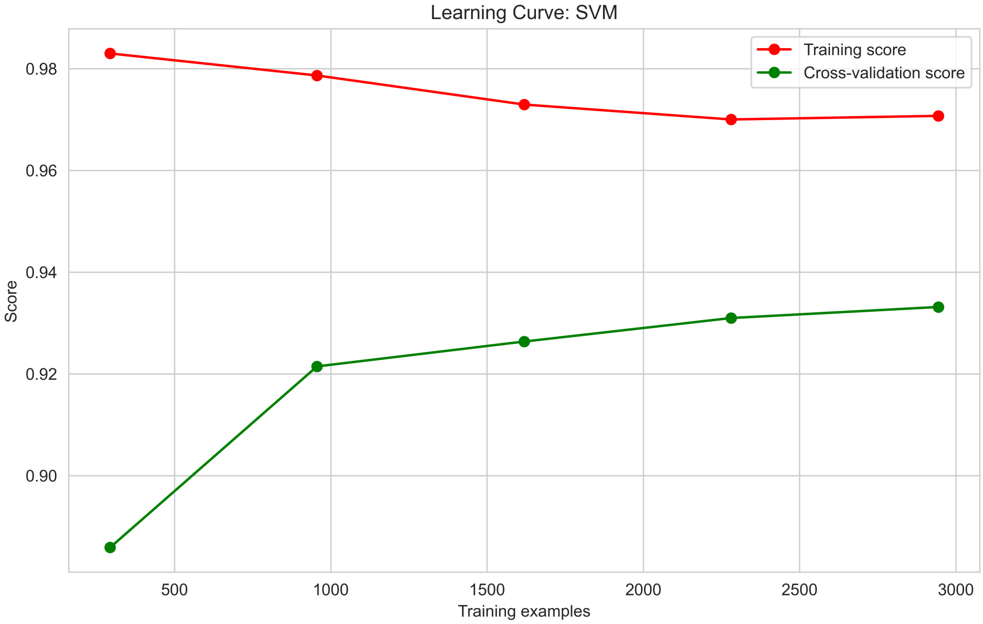

Figure 6: Learning Curve: SVM (RBF)