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

Name: Mariya Joevita
Reg.No: 3122235001077
Class: CSE-B

1. Aim and 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 Description

The Spambase dataset contains email message features used to classify emails as spam or non-spam. It consists of word frequency, character frequency, and capital run-length features.

- Total Features: 57
- Target Class: Binary (0 – Non-Spam, 1 – Spam)

Dataset Source: Kaggle – Spambase Dataset

Preprocessing Steps

Before training the machine learning models, the dataset was preprocessed to improve data quality and ensure compatibility with the algorithms.

Loading the Dataset

The dataset contains a total of 4601 email samples. Each sample is represented using 57 numerical features and belongs to one of two classes:

- Spam (1)
- Non-spam (0)

Checking Missing Values

All features were checked for missing values.

$$\text{Number of missing values} = 0$$

Since no missing values were present, no data cleaning was required.

Separating Features and Target

The dataset was divided into input features and output labels.

$$X = \text{All columns except the class label}$$

$$y = \text{Class label column}$$

Here, X represents the feature matrix and y represents the target vector.

Train–Test Split

The dataset was split into training and testing sets to evaluate model performance.

- Training set: 80% of the data (3680 samples)
- Testing set: 20% of the data (921 samples)

Stratified sampling was used to maintain the same class distribution in both sets.

Feature Scaling

Feature scaling was applied to ensure that all features contribute equally to the model.

For each feature, standardization was performed as:

$$\text{Scaled value} = \frac{\text{Value} - \text{Mean}}{\text{Standard Deviation}}$$

After scaling:

- Mean = 0
- Standard deviation = 1

Result of Preprocessing

After completing the preprocessing steps, the dataset was clean, well-structured, and ready for model training.

Exploratory Data Analysis (EDA)

EDA was performed to understand the dataset characteristics:

- Dataset shape, data types, and statistical summary
- Class distribution analysis
- Feature distribution visualization for word frequency features

Model Implementation

This section describes the implementation of the classification models used for spam detection on the Spambase dataset. The dataset consists of numerical features extracted from email content along with a binary class label indicating spam or non-spam (ham). Prior to model training, the dataset was split into training and testing sets, and feature scaling was performed using standardization to ensure uniform feature ranges.

Logistic Regression

Logistic Regression was implemented as a baseline linear classification model. It estimates the probability of an email being spam using a sigmoid activation function and applies a decision threshold of 0.5 for class prediction. To prevent overfitting and improve generalization, regularization was incorporated into the model.

Hyperparameter tuning was performed using Grid Search with 5-fold cross-validation. Both L1 (Lasso) and L2 (Ridge) regularization techniques were evaluated. The inverse regularization strength parameter $C \in \{0.01, 0.1, 1, 10, 100\}$ was tuned, and the `liblinear` and `saga` solvers were used as they support both L1 and L2 penalties. Model selection was based on cross-validated accuracy.

Support Vector Machine

Support Vector Machine (SVM) was implemented as a margin-based classifier to capture both linear and non-linear decision boundaries. Multiple kernel functions were explored, including Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid kernels.

Hyperparameter tuning for SVM was carried out using Grid Search with 5-fold cross-validation. The regularization parameter $C \in \{0.1, 1, 10, 100\}$ was tuned across all kernels. For non-linear kernels (RBF, Polynomial, and Sigmoid), the kernel coefficient $\gamma \in \{\text{scale}, \text{auto}\}$ was evaluated. Additionally, the degree parameter for the Polynomial kernel was tuned over $\{2, 3, 4\}$. The best model was selected based on cross-validation accuracy.

Model Selection

For both Logistic Regression and SVM, the optimal hyperparameters were selected using Grid Search with cross-validation. The best-performing models were further evaluated using standard classification metrics on the test dataset. This approach ensured fair comparison and robust estimation of model performance.

Visualizations

This section presents the confusion matrices for Logistic Regression and various SVM kernels, providing a detailed view of classification performance in terms of true positives, true negatives, false positives, and false negatives.

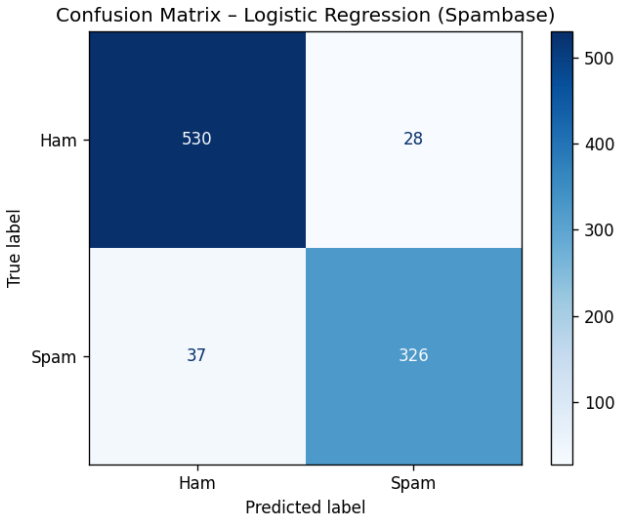

Figure 1: Confusion Matrix for Logistic Regression

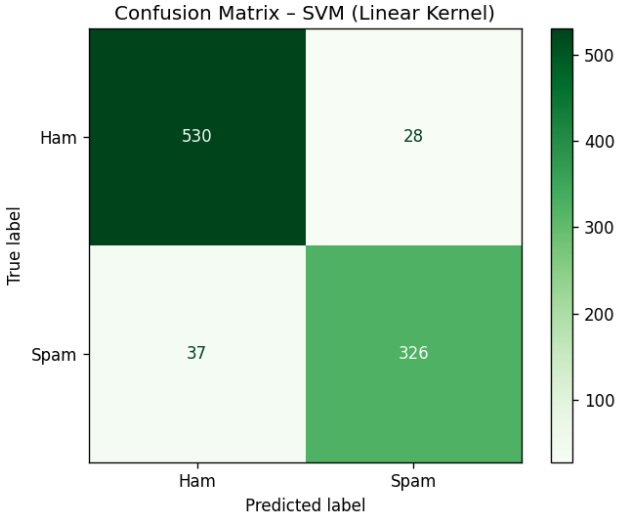

Figure 2: Confusion Matrix for SVM with Linear Kernel

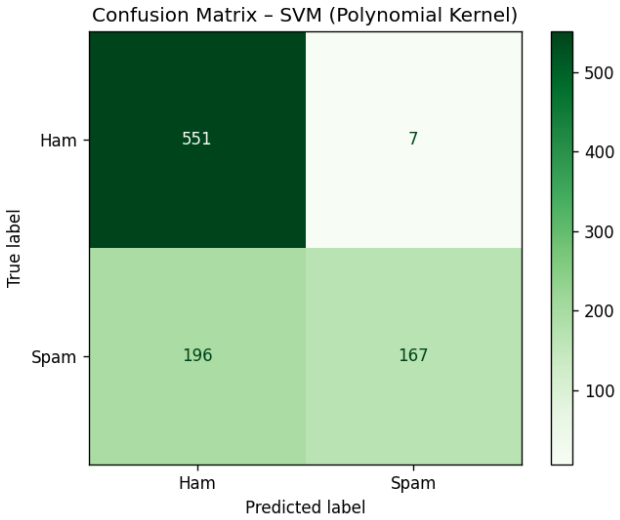

Figure 3: Confusion Matrix for SVM with Polynomial Kernel

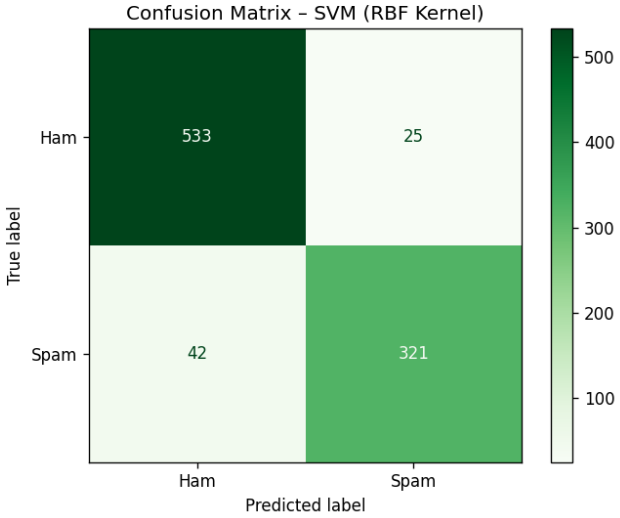

Figure 4: Confusion Matrix for SVM with RBF Kernel

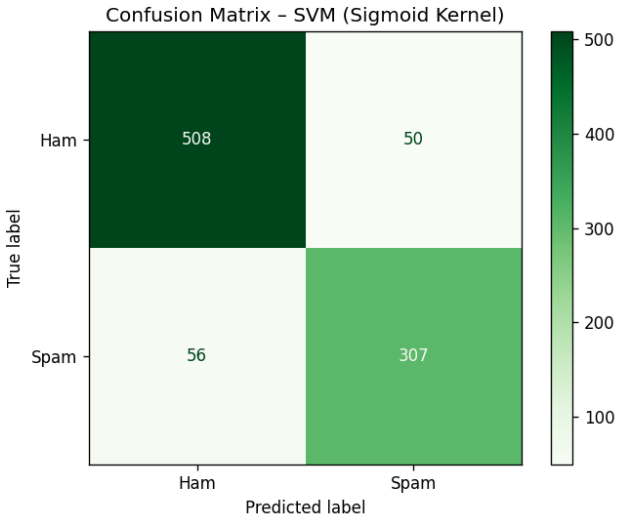

Figure 5: Confusion Matrix for SVM with Sigmoid Kernel

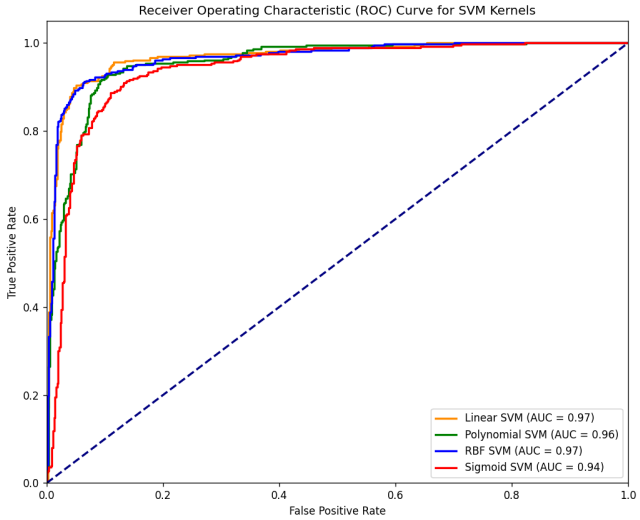

Figure 6: Confusion Matrix for Tuned SVM Model

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	Penalty = L1, $C = 10$, Solver = saga	0.9239
SVM	Grid Search	Kernel = RBF, $C = 10$, γ = scale	0.9332

Logistic Regression Performance

Metric	Value
Accuracy	0.9294
Precision	0.9209
Recall	0.8981
F1 Score	0.9093
Training Time (s)	0.0561

SVM Kernel-wise Performance

Kernel	Accuracy	F1 Score	Training Time (s)
Linear	0.9294	0.9093	0.8765
Polynomial	0.7796	0.6220	0.6792
RBF	0.9273	0.9055	0.3438
Sigmoid	0.8849	0.8528	0.4013

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	0.9144	0.9334
Fold 4	0.9198	0.9212
Fold 5	0.9266	0.9361
Average	**0.9239**	**0.9332**

Comparative Analysis

Criterion	Logistic Regression	SVM
Accuracy	0.9239	0.9332
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