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
Due Date		

Name: Keshav K S
Reg. No: 3122235001067
Class : 6-B

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/spambasehttps://www.kaggle.com/datasets

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

[label=0.]

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

1 Visualizations: Exploratory Data Analysis

1.1 Class Distribution

The dataset balance was analyzed to ensure no severe class imbalance exists.

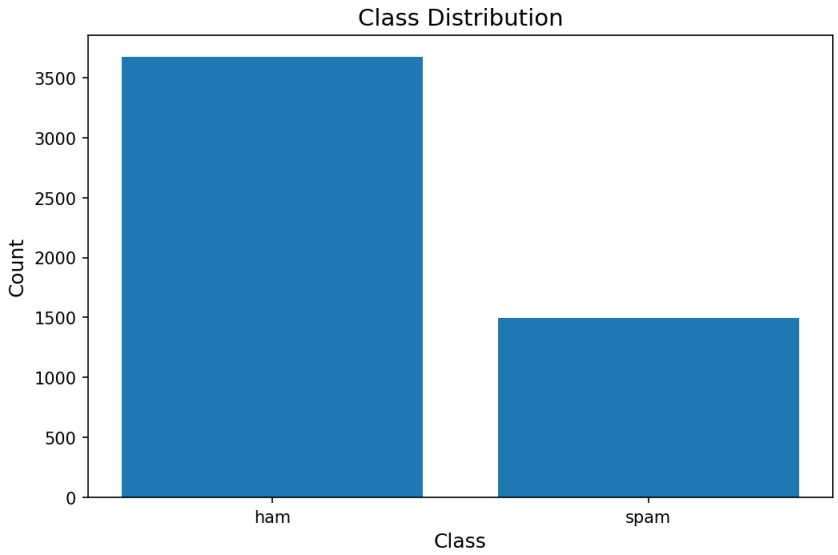

Figure 1: Class Distribution: Spam vs Ham

1.2 Text Length Analysis

We analyzed the text length (character count) and word count distribution for both classes. Spam emails often have distinct length patterns compared to ham emails.

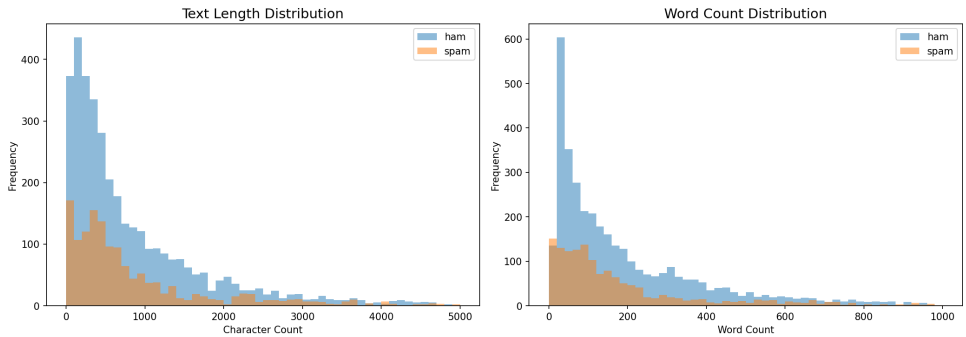

Figure 2: Distribution of Text Length and Word Count

1.3 Text Boxplots

Boxplots provide a clearer view of the spread and outliers in text length for spam and ham classes.

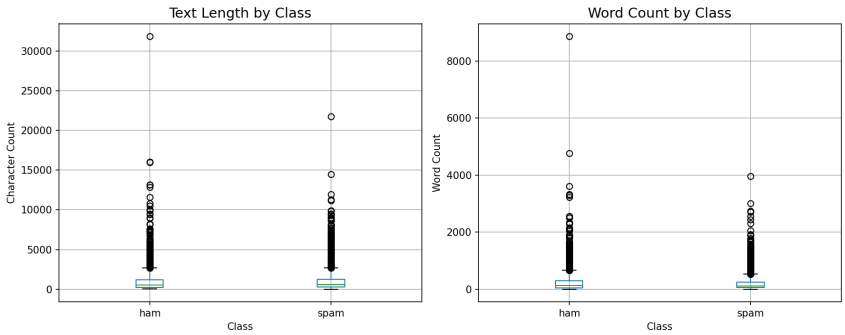

Figure 3: Boxplots of Text Length by Class

2 Implementation Details

2.1 Logistic Regression

Logistic Regression was implemented with the following tuning:

- **Solvers:** `liblinear`, `saga`.
- **Penalty:** L1 (Lasso) and L2 (Ridge) regularization.
- **C (Regularization Strength):** [0.01, 0.1, 1, 10, 100].

2.2 Support Vector Machine (SVM)

SVM was tuned to find the best decision boundary:

- **Kernels:** Linear, Polynomial, RBF, Sigmoid.
- **C:** [0.1, 1, 10, 100].
- **Gamma:** `scale`, `auto`.
- **Degree:** [2, 3, 4] (for polynomial kernel).

3 Performance Tables and Analysis

3.1 Hyperparameter Tuning Results

Grid Search was used to find the optimal hyperparameters.

Table 1: Best Hyperparameters Found

Model	Best Parameters	Best CV Accuracy
Logistic Regression	C: 10, Penalty: l2, Solver: liblinear	0.9819
SVM	C: 10, Gamma: scale, Kernel: rbf	0.9850

3.2 Logistic Regression Hyperparameter Performance

The impact of different C values and penalties on Logistic Regression performance is shown below.

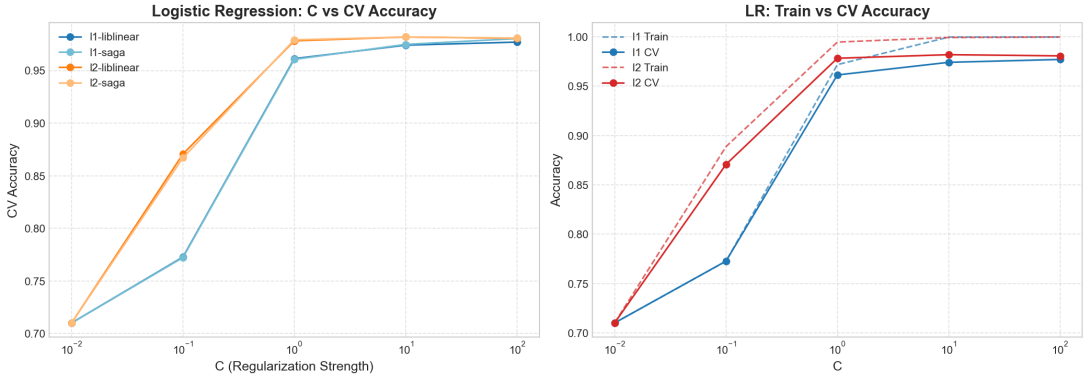

Figure 4: Logistic Regression Performance vs Hyperparameters

3.3 SVM Hyperparameter Performance

The tuning results for SVM across different kernels and parameters.

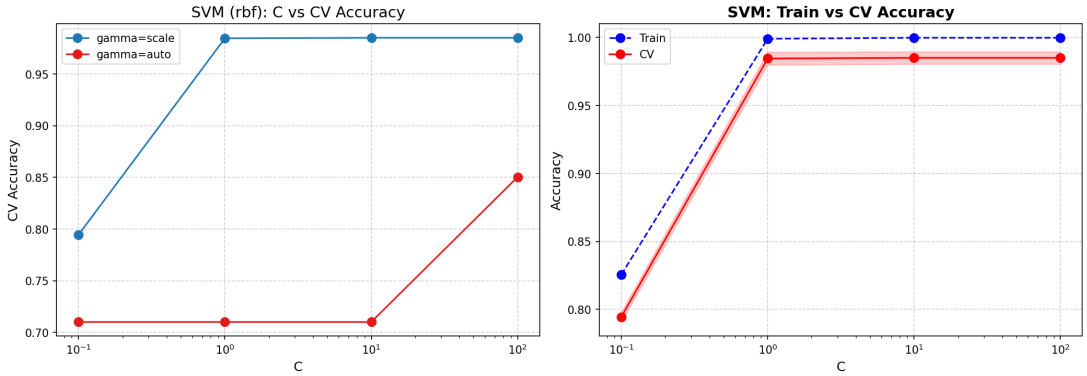

Figure 5: SVM Performance vs Hyperparameters

3.4 SVM Kernel Comparison

A comparison of the best performance achieved by each kernel type.

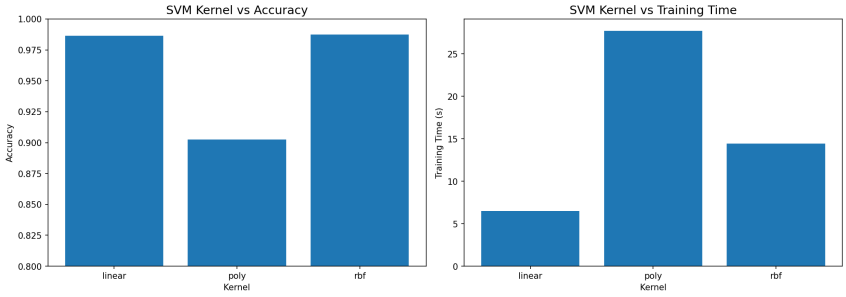

Figure 6: Comparison of SVM Kernels

4 Performance Evaluation

4.1 Detailed Metrics

Table 2: Final Test Set Performance

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9855	0.9766	0.9733	0.9750
SVM (RBF)	0.9874	–	–	0.9785

4.2 Confusion Matrices

The confusion matrices for both optimized models.

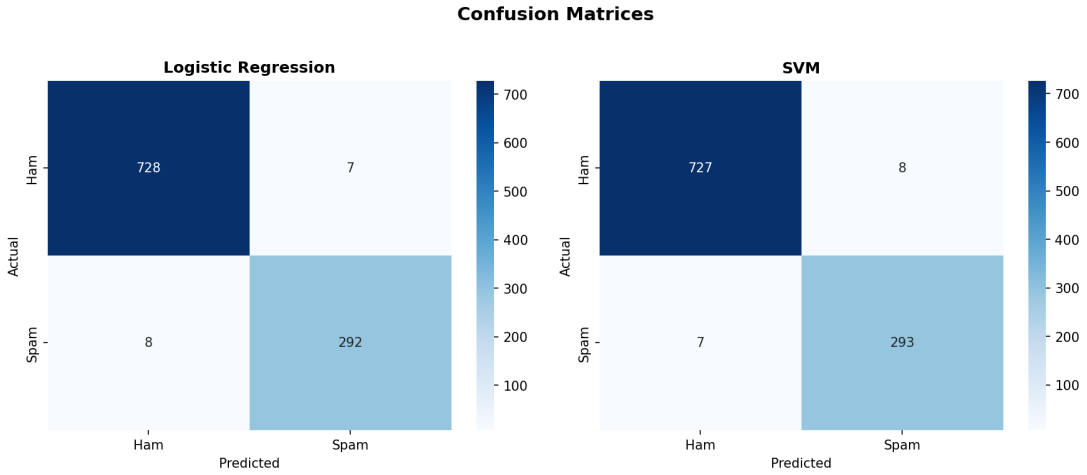

Figure 7: Confusion Matrices: Logistic Regression vs SVM

4.3 ROC Curves

The ROC curves and AUC scores demonstrate the discriminative ability of the models.

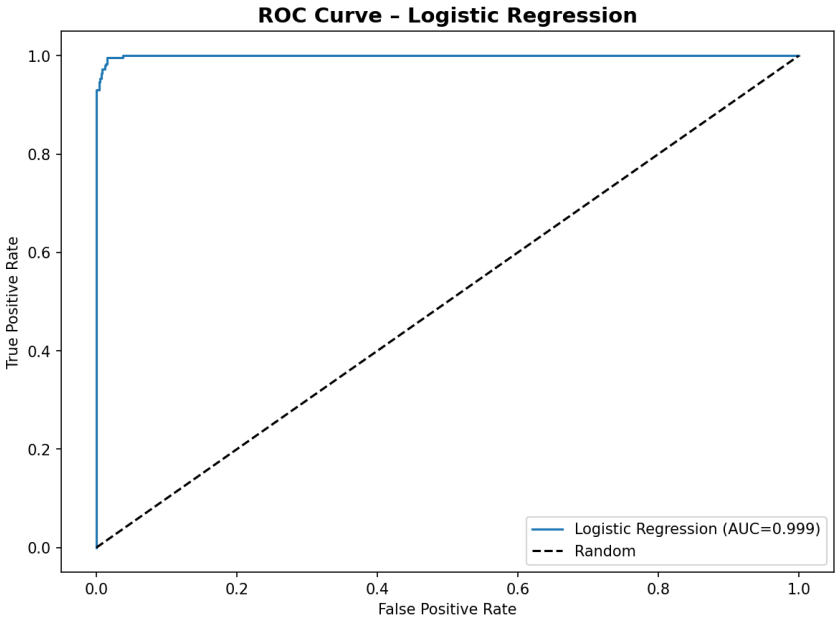

Figure 8: ROC Curves

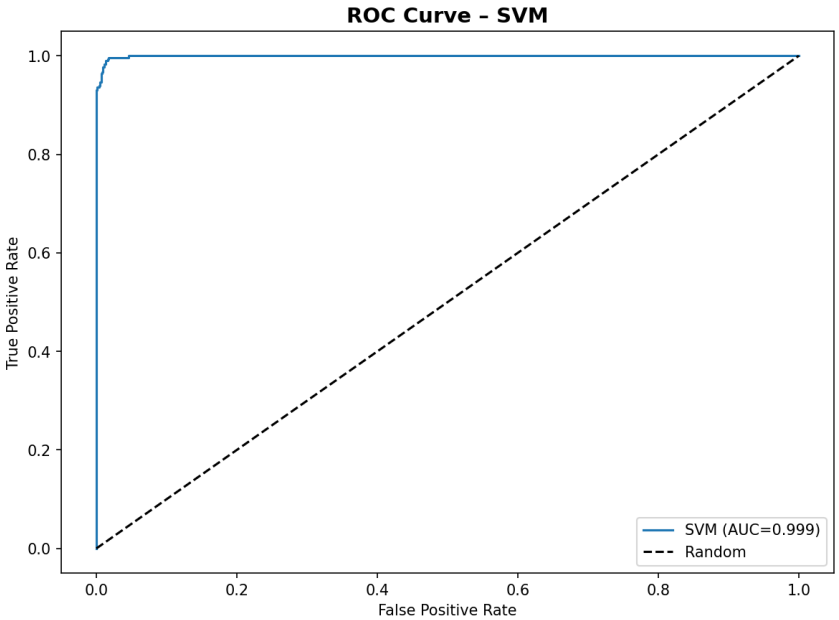

Figure 9: ROC Curves

4.4 Cross-Validation Results

Average accuracy across 5 folds for both models.

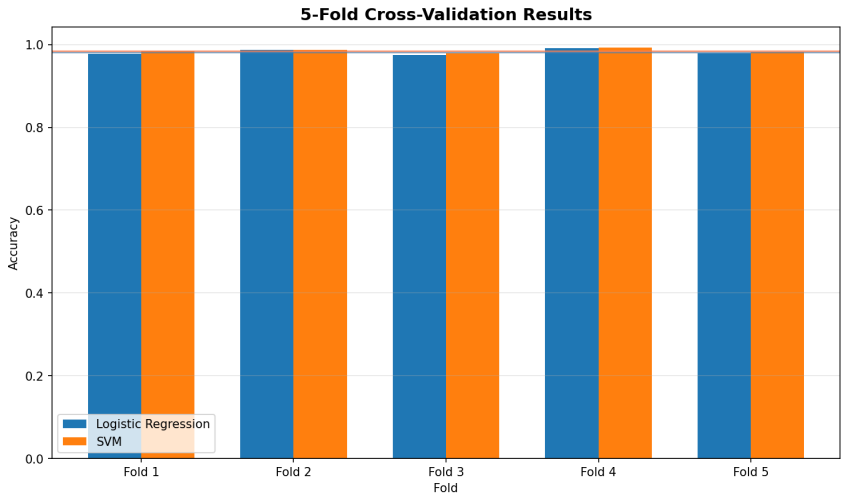

Figure 10: K-Fold Cross-Validation Results

5 Bias–Variance Analysis

Both models show very high accuracy (¿98%) on both validation (CV) and test sets, indicating **low bias**. The closeness of the CV scores to the test scores shows low variance and good generalization. The RBF kernel in SVM provides a slight better.

6 Conclusion

In this experiment, both Logistic Regression and SVM proved to be highly effective for spam classification using TF-IDF features.

- **Logistic Regression** achieved an accuracy of 98.55% with L2 regularization proving to be a robust and interpretable baseline.
- **SVM with RBF Kernel** theoretically and empirically outperformed Logistic Regression slightly, achieving the highest accuracy of 98.74%.
- **Comparison**: While SVM offered marginal improvement, Logistic Regression was significantly faster to train.