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
Name	Piranow C	Register No.	3122235001096
Due Date	27.01.2026		

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

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.

2. Dataset Description

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: spambase

3. Preprocessing Steps

- The dataset was loaded into a Pandas DataFrame for preprocessing.
- Input features and target labels were separated for modeling.
- A stratified train–test split was performed to preserve class distribution.
- Feature scaling was applied using the StandardScaler technique.

4. Implementation Details

- Implemented baseline Logistic Regression classifier
- Tuned Logistic Regression hyperparameters using RandomizedSearchCV
- Implemented Support Vector Machine classifiers with different kernels
- Tuned SVM hyperparameters using RandomizedSearchCV
- Compared linear, polynomial, RBF, and sigmoid kernels

5. Visualizations

5.1 Class Distribution

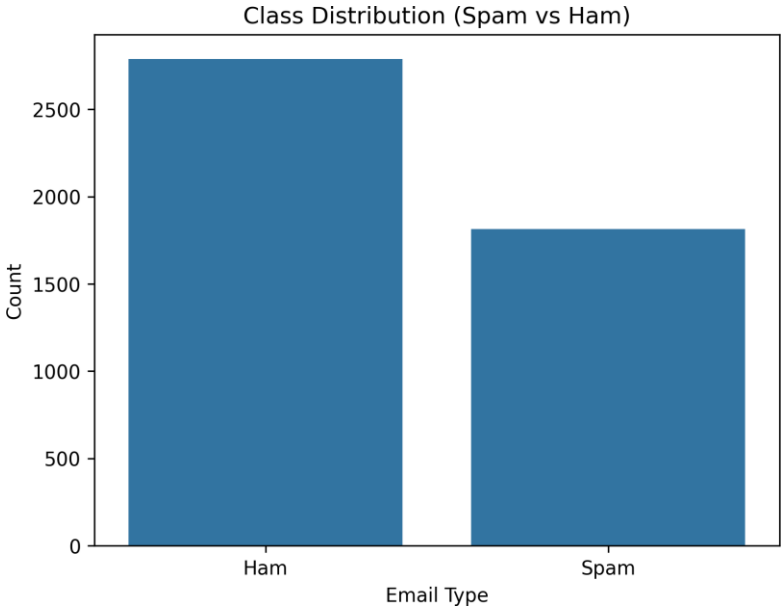


Figure 1: Class Distribution of Spam and Non-Spam Emails

5.2 Features Distribution

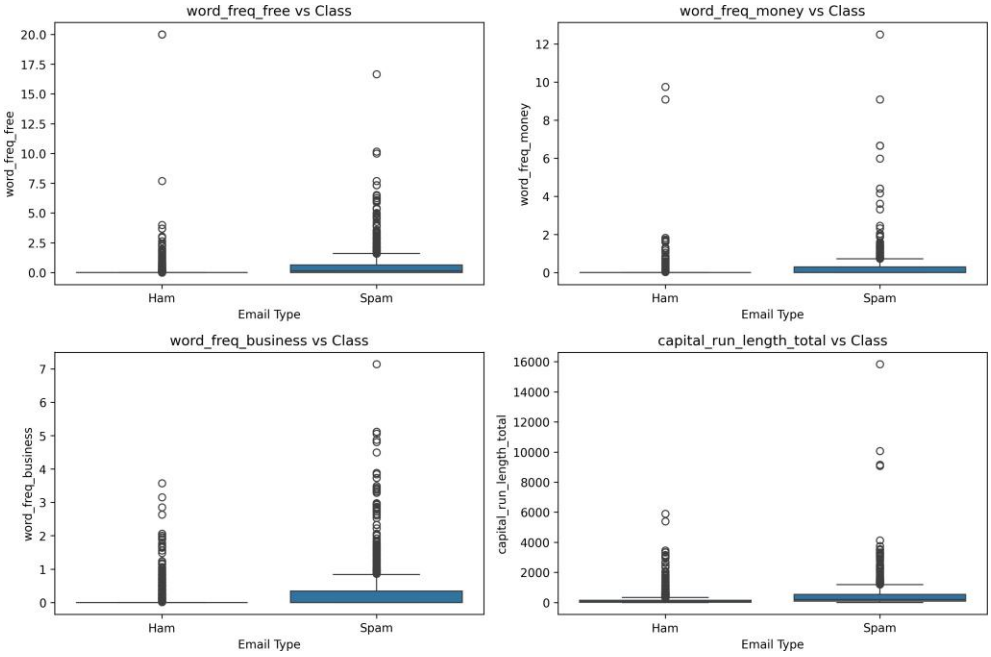
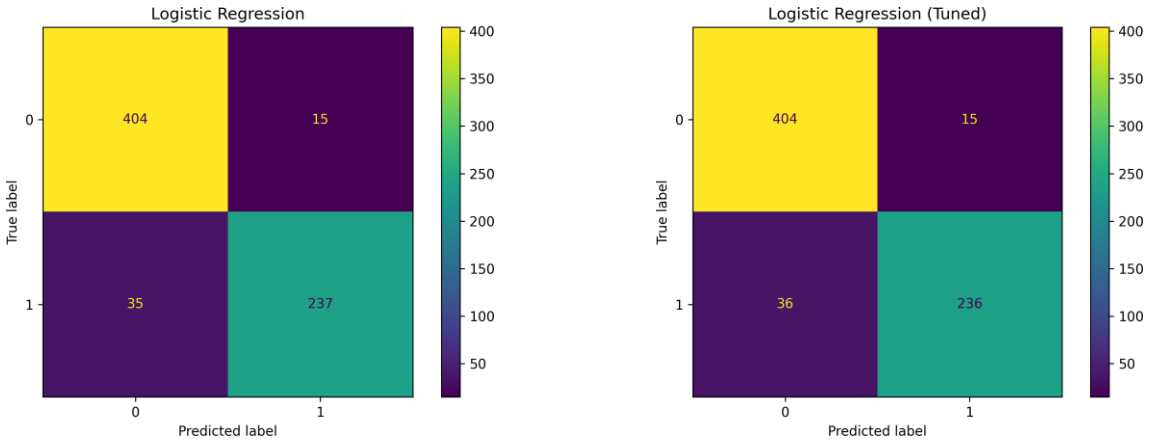


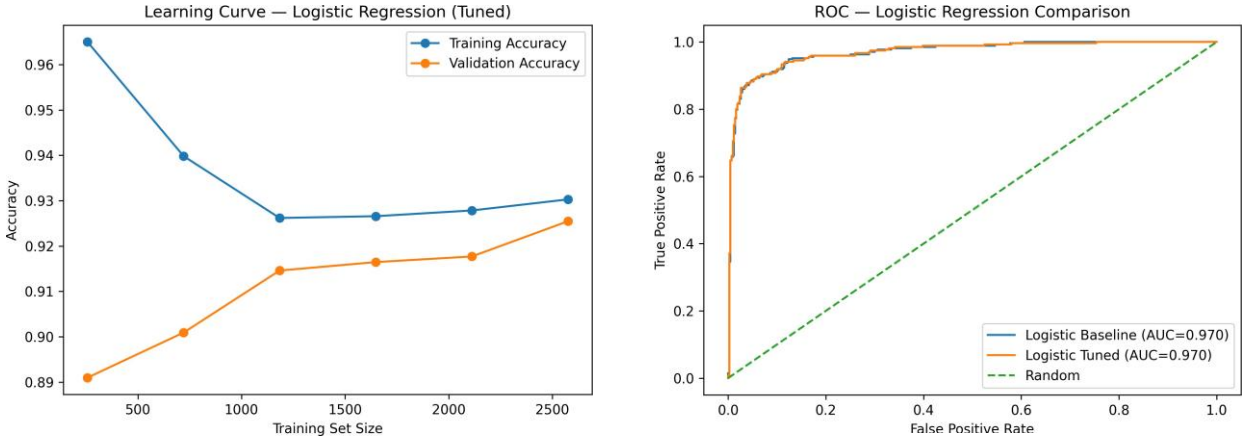
Figure 2: Box-Plots for important features

5.3 Logistic Regression Results



a. Logistic Regression b. Tuned Logistic Regression (Randomized)

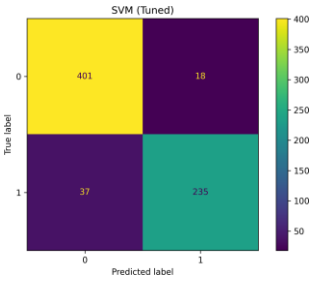
Figure 3: Linear Regression Confusion Matrices



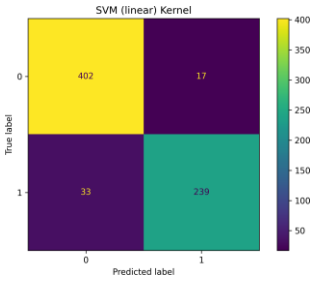
a. Learning curve for Logistic Regression b. ROC for Logistic Regression

Figure 4: Linear Regression Plots

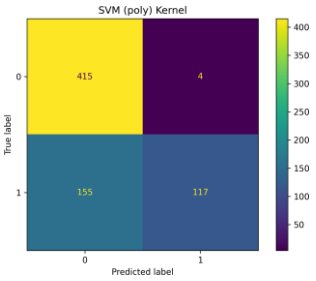
5.4 Support Vector Machine Results



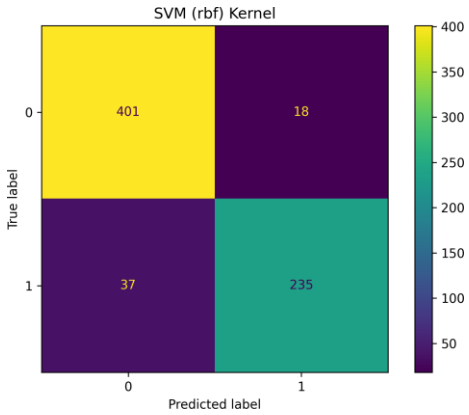
a. SVM (Tuned)



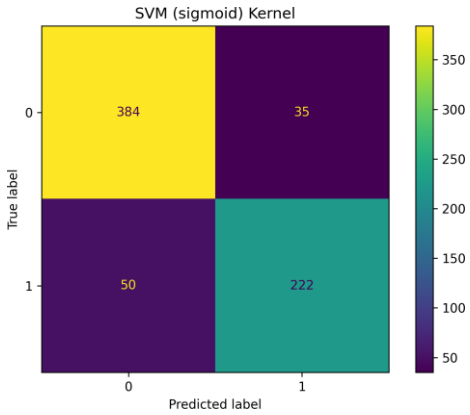
b. SVM (Linear)



c. SVM (Polynomial)

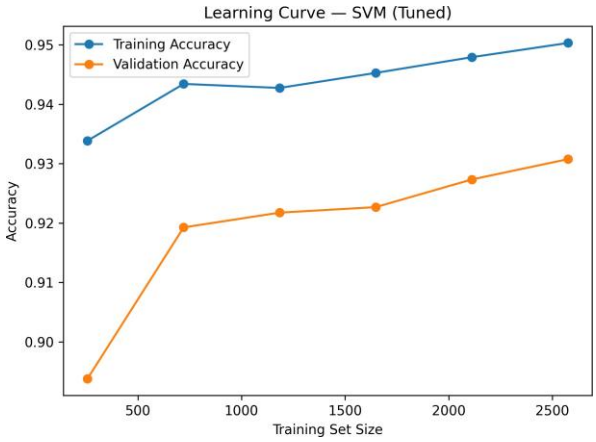


d. SVM (RBF)

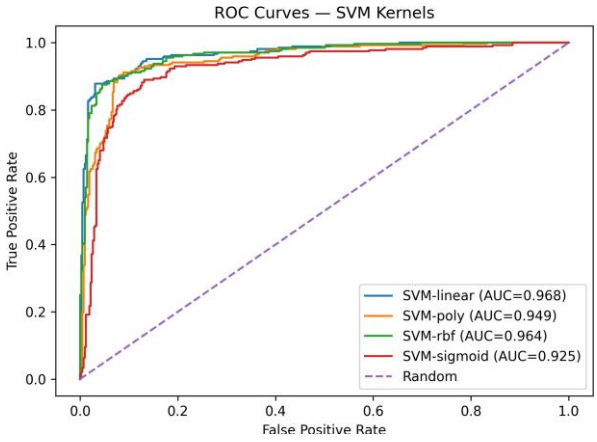


e. SVM (Sigmoid)

Figure 5: Confusion Matrices for Support Vector Machine Models



a. Learning curve for SVM



b. ROC for SVM

Figure 6: Linear Regression Plots

6. Performance Tables

6.1 Hyperparameter Tuning Results

Model	Search Method	Best Parameters	Best CV Accuracy
Logistic Regression	Random	C=100, penalty=L1, solver=saga	0.9255
SVM	Random	kernel=rbf, C=1, gamma=scale, degree=4	0.9307

6.2 Logistic Regression Performance

Metric	Value
Accuracy	0.9290
Precision	0.9055
Recall	0.9154
F1 Score	0.9104
Training Time (s)	0.0360

6.3 SVM Kernel-wise Performance

Kernel	Accuracy	F1 Score	Training Time (s)
Linear	0.9304	0.9130	0.4329
Polynomial	0.7710	0.6089	0.4180
RBF	0.9348	0.9168	0.2860
Sigmoid	0.8913	0.8644	0.2630

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

Fold	Logistic Regression	SVM
Fold 1	0.9161	0.9208
Fold 2	0.9286	0.9425
Fold 3	0.9270	0.9208
Fold 4	0.9239	0.9348
Fold 5	0.9317	0.9348
Average	0.9255	0.9307

6.5 Comparative Analysis

Criterion	Logistic Regression	SVM
Accuracy	0.9255	0.9307
Model Complexity	Low	High
Training Time	Low	High
Interpretability	High	Low

7. Observations and conclusion

The RBF-kernel Support Vector Machine was the best-performing classifier, achieving the highest test and cross-validation accuracies, slightly exceeding both baseline and tuned Logistic Regression models. Logistic Regression benefited from reduced regularization (high C) and L1 penalty, which allowed the model to learn more complex decision boundaries and marginally improved performance over the baseline. SVM performance depended strongly on kernel choice: linear and RBF kernels worked well, while the polynomial kernel performed poorly and the sigmoid kernel produced moderate results. From a bias–variance perspective, tuning reduced the bias in Logistic Regression, and the RBF SVM provided the most balanced trade-off, as reflected in its stable cross-validation scores.

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