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:	Nanda Kumar B	Roll No:	3122235001701
Due Date	27.01.2026		

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

1. Aim & 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

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 (spambase_csv_Kaggle.csv) was loaded into a Pandas DataFrame.
- The dataset was split into the feature matrix (X) and the target vector (y).
- The StandardScaler from the Scikit-Learn library was applied to normalize the feature values.
- The data was divided into training and testing sets for the evaluation of model performance (X_train & X_test & y_train and y_test).

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. Visualization

5.1 Class Distribution

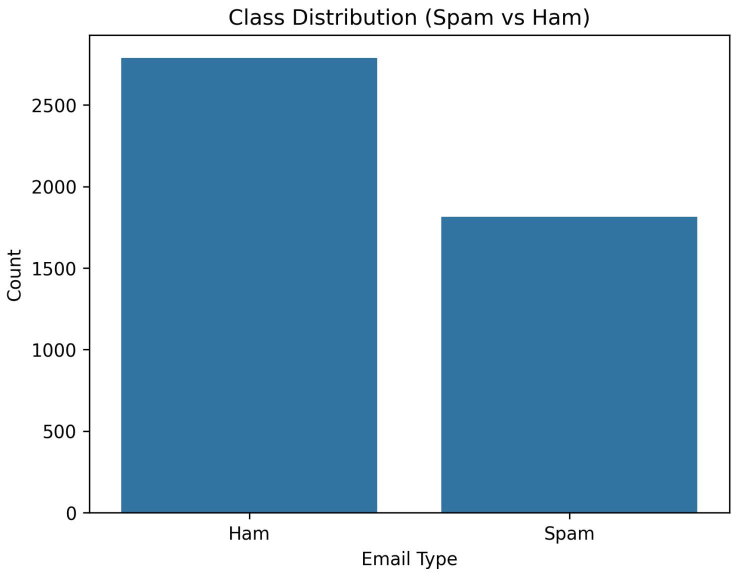

Figure 1: Class Distribution of Spam and Non-Spam Emails

5.2 Feature Distribution

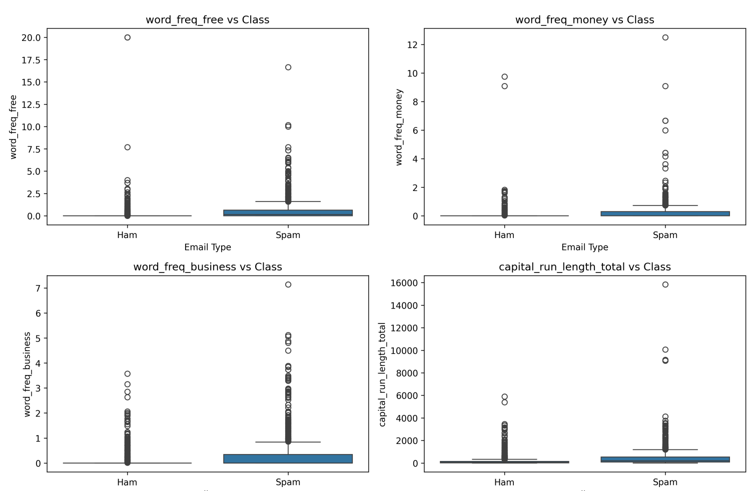

Figure 2: Box Plot

5.3 Logistic Regression Result

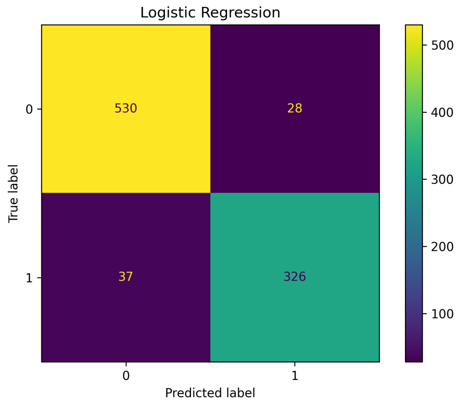

Figure 3: Logistic Regression

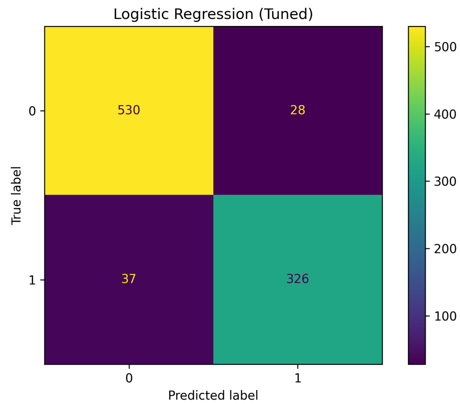

Figure 4: Hyperparameter tuned - Logistic regression

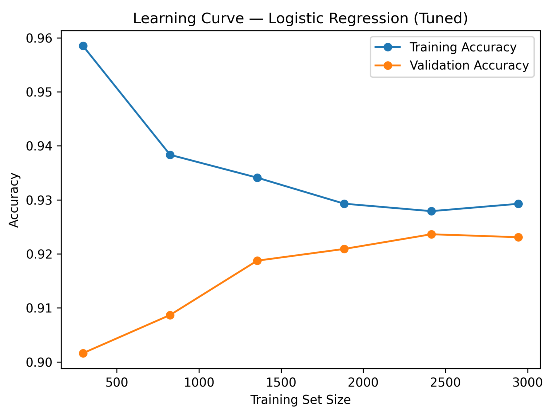

Figure 5: Learning Curve - Logistic Regression

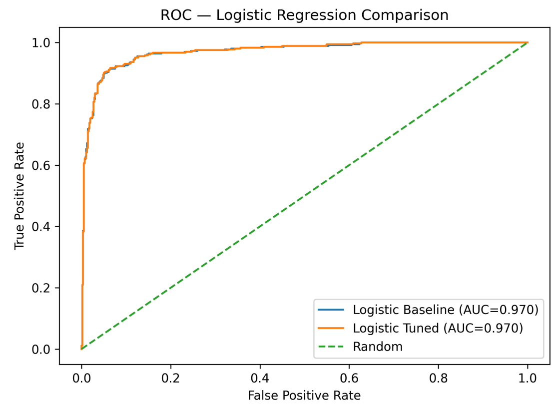

Figure 6: ROC Curve for Logistic Regression

5.4 Support Vector Machine Results

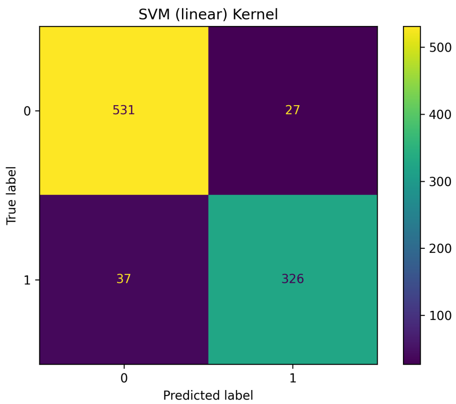

Figure 7: Confusion Matrix - SVM Linear

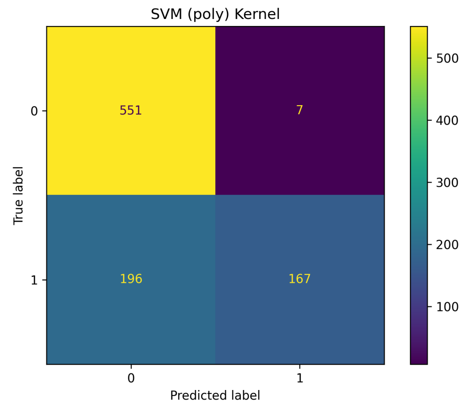

Figure 8: Confusion Matrix - SVM Poly

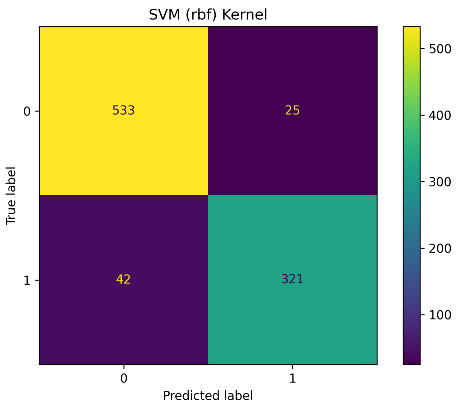

Figure 9: Confusion Matrix - SVM RBF

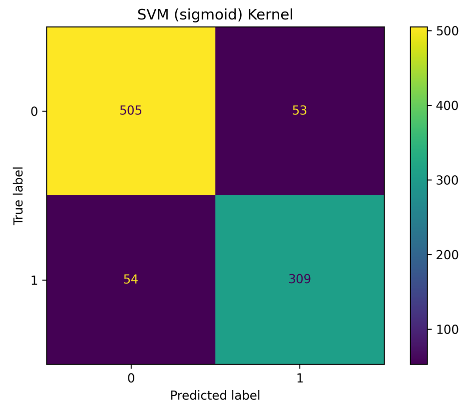

Figure 10: Confusion Matrix - SVM Sigmoid

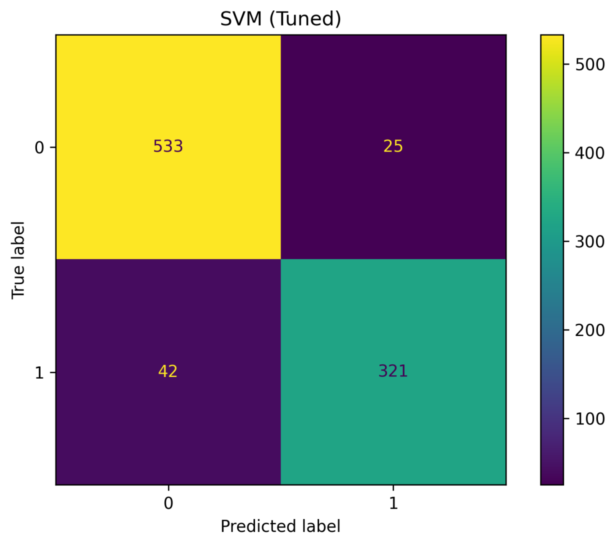

Figure 11: Confusion Matrix - SVM Hyperparameter tuned

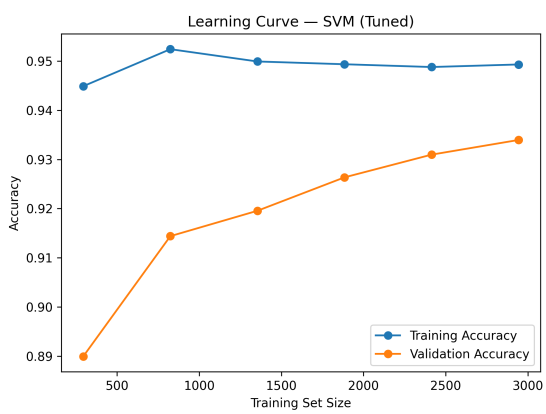

Figure 12: Learning Curve - SVM

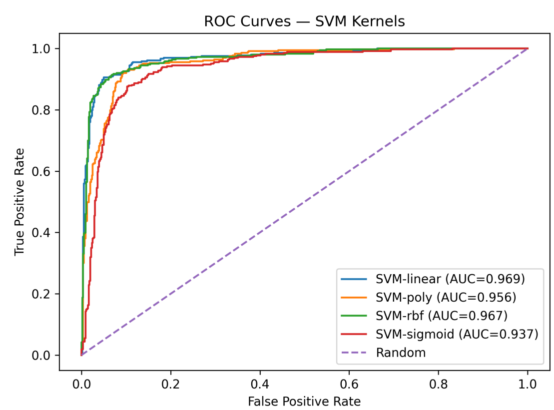

Figure 13: ROC Curve for Logistic Regression

6. Performance Table

6.1 Hyperparameter Tuning Summary

Model	Search Method	Best Parameters	Best CV Accuracy
Logistic Regression	Grid / Random	$solver$: $saga$, $penalty$: $l2$, C: 1	0.929
SVM	Grid / Random	$kernel$: rbf, $gamma$: $scale$, $degree$: 4, C: 1	0.933

6.2 Logistic Regression Performance

Metric	Value
Accuracy	0.9294
Precision	0.9209
Recall	0.8980
F1 Score	0.9093
Training Time (s)	10.9785

6.3 SVM Kernel-wise Performance

Kernel	Accuracy	F1 Score	Training Time (s)
Linear	0.9305	0.9106	0.6709
Polynomial	0.7795	0.6219	0.5335
RBF	0.9272	0.9055	0.3566
Sigmoid	0.8838	0.8524	0.3397

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

Fold	Logistic Regression	SVM
Fold 1	0.9375	0.9457
Fold 2	0.9144	0.9334
Fold 3	0.9211	0.9280
Fold 4	0.9157	0.9226
Fold 5	0.9266	0.9402
Average	0.9230	0.9339

6.5 Comparative Analysis

Criterion	Logistic Regression	SVM
Accuracy	0.93	0.93
Model Complexity	Low	High
Training Time	Low	High
Interpretability	High	Low

7. Observations

- 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.
- Different kernels map data into different feature spaces
  - Linear Kernel
    * Works well for near-linear boundaries.
    * Fast and Performed almost as well as RBF → dataset is fairly linearly separable.
  - Polynomial Kernel
    * Adds complex curved boundaries.
    * Poor performance, likely overfitting.
  - RBF Kernel
    * Non-linear, flexible.
    * Captures complex patterns. Best overall after tuning.
  - Sigmoid Kernel
    * Moderate results.

8. Learning Outcomes

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

9. References

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