Binary Classification of Spam Emails using Logistic Regression and Support Vector Machines

Ramlath Nisha A
B.E. Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering, Chennai
Email: ramlathnisha2310611@ssn.edu.in

***Abstract*—This work presents a binary classification framework for detecting spam emails using Logistic Regression and Support Vector Machine (SVM) classifiers on the Spambase dataset. The numerical features extracted from email content are standardized and used to train baseline models, which are subsequently improved through hyperparameter tuning with grid search and 5-fold cross-validation. Logistic Regression is evaluated with L1 and L2 regularization, while SVM is studied with linear, polynomial, radial basis function (RBF), and sigmoid kernels. Experimental results show that the tuned RBF SVM achieves the best overall performance with an accuracy of approximately 93.49% and F1-score of 0.921 on the test set, outperforming both baseline and tuned Logistic Regression. A detailed comparative analysis highlights the impact of regularization strength, kernel choice, and hyperparameter settings on accuracy, complexity, and bias–variance trade-off.**

***Index Terms*—Binary classification, Logistic Regression, Support Vector Machines, Hyperparameter tuning, Spam detection, Spambase.**

I. Introduction

Email remains a primary mode of communication, which makes spam detection a crucial real-world classification problem. Manually filtering spam is infeasible at scale, motivating the use of machine learning models that automatically distinguish spam from legitimate (ham) emails.

This experiment is conducted as part of the course UCS2612 – Machine Learning Algorithms Laboratory, B.E. Computer Science and Engineering, Semester VI, at Sri Sivasubramaniya Nadar College of Engineering. The objective is to implement and compare probabilistic and margin-based classifiers for binary classification, analyze the effect of regularization and kernel choice, and apply hyperparameter tuning and cross-validation to improve classification performance.

The specific goals of this experiment are:

- To classify emails as spam or ham using Logistic Regression and SVM classifiers.
- To compare linear and kernel-based decision boundaries using different SVM kernels.
- To apply grid search based hyperparameter tuning for both models.
- To evaluate models using standard metrics and 5-fold cross-validation.
- To interpret the effect of regularization, kernel behavior, and bias–variance trade-off.

II. Dataset Description

The experiments use the Spambase dataset, which contains 4601 email samples with 57 numerical features and a binary class label. Each feature captures word frequencies, character frequencies, or patterns related to capital letters in the email.

A. Dataset Characteristics

- Number of instances: 4601
- Number of features: 57 numeric predictor variables
- Target label: `class` (1 = spam, 0 = ham)
- Feature types:
  - Word frequency features such as `word_freq_make`, `word_freq_free`, `word_freq_email`.
  - Character frequency features such as `char_freq_%21`, `char_freq_%24`.
  - Capital-run features such as `capital_run_length_average`, `capital_run_length_longest`, `capital_run_length_total`.
- Missing values: None (all features have zero missing entries as per the initial analysis).

B. Exploratory Data Analysis

A basic exploratory data analysis (EDA) was performed to understand the class distribution and feature relationships.

1) Class Distribution: The proportion of spam versus ham emails was visualized using a count plot on the `class` variable.

2) Correlation Structure: The correlation matrix of all features was computed and visualized using a heatmap to identify groups of correlated predictors and potential redundancies.

III. Theory Background

A. Logistic Regression

Logistic Regression is a probabilistic model used for binary classification. Given a feature vector $x \in \mathbb{R}^d$, the model computes the probability that the label $y \in \{0, 1\}$ equals 1 using the sigmoid (logistic) function

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w^T x + b)}}, \tag{1}$$

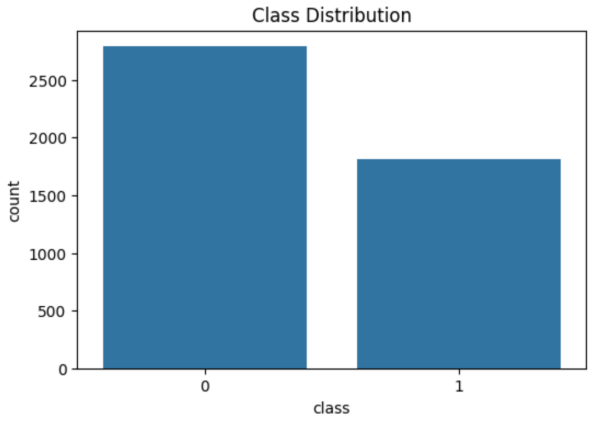

Fig. 1. Class distribution of spam and ham emails.

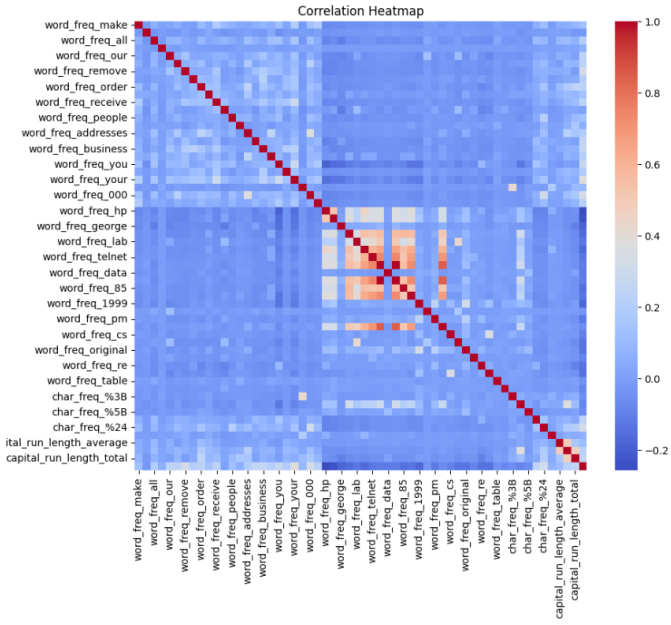

Fig. 2. Correlation heatmap of Spambase features.

where $w \in \mathbb{R}^d$ is the weight vector and $b \in \mathbb{R}$ is the bias. A threshold (typically 0.5) is used to convert probabilities into class labels.

The model is trained by minimizing the logistic (cross-entropy) loss

$$\mathcal{L}(w,b) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right], \quad (2)$$

where N is the number of samples and $\hat{y}_i$ is the predicted probability for sample i.

1) Regularization: To prevent overfitting, regularization adds a penalty term on the magnitude of the weights.

L1 regularization (Lasso) encourages sparsity and can perform implicit feature selection:

$$\mathcal{L}_{\text{L1}} = \mathcal{L} + \lambda\|w\|_1, \quad (3)$$

where $\lambda > 0$ controls the regularization strength.

L2 regularization (Ridge) penalizes large weights but typically keeps all features:

$$\mathcal{L}_{\text{L2}} = \mathcal{L} + \lambda\|w\|_2^2. \quad (4)$$

The inverse regularization strength $C = 1/\lambda$ is used in scikit-learn; smaller C corresponds to stronger regularization.

2) Hyperparameters: Key hyperparameters for Logistic Regression include:

- Regularization type: L1 or L2.
- $C \in \{0.01, 0.1, 1, 10, 100\}$.
- Solver:
  - `liblinear`: suitable for small to medium-sized datasets; supports L1 and L2.
  - `saga`: efficient for large datasets; supports L1 and L2.

B. Support Vector Machine

Support Vector Machine (SVM) is a margin-based classifier that seeks an optimal hyperplane separating classes with maximum margin. In the linear case, SVM solves

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad (5)$$

subject to

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad (6)$$

where $\phi(x)$ is a feature mapping induced by a kernel function, and C controls the trade-off between margin width and misclassification penalties.

1) Kernels: Kernels define similarity in a transformed feature space without explicitly computing $\phi(x)$. Common kernels used in this experiment are:

- Linear: $K(x,x') = x^Tx'$.
- Polynomial: $K(x,x') = (\gamma x^Tx' + r)^d$.
- RBF: $K(x,x') = \exp(-\gamma\|x-x'\|^2)$.
- Sigmoid: $K(x,x') = \tanh(\gamma x^Tx' + r)$.

2) Hyperparameters: Key hyperparameters for SVM in this experiment are:

- Kernel: `linear`, `poly`, `rbf`, `sigmoid`.
- $C \in \{0.1, 1, 10, 100\}$.
- $\gamma \in \{\texttt{scale}, \texttt{auto}\}$.
- Degree $d \in \{2, 3, 4\}$ for polynomial kernels.

C. Hyperparameter Tuning

Hyperparameter tuning was performed using grid search. Grid Search systematically evaluates all combinations of hyperparameter values from a predefined search space using cross-validation, selecting the configuration that maximizes a chosen performance metric (here, accuracy). Randomized search is an alternative that samples combinations randomly; in this experiment, grid search was used for both Logistic Regression and SVM.

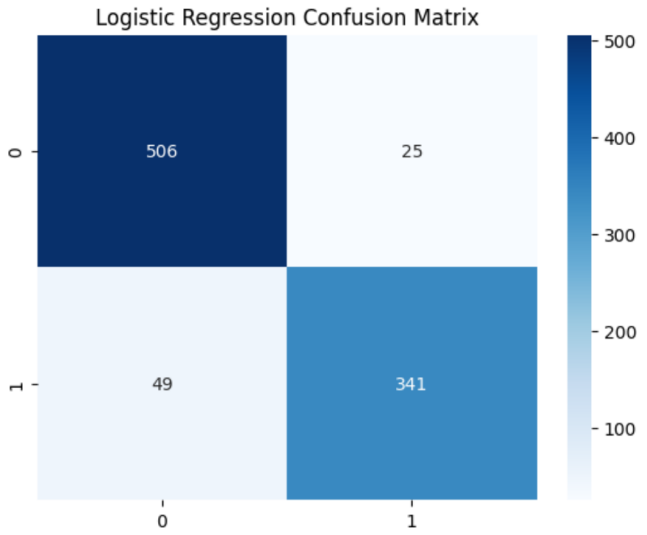

Fig. 3. Confusion matrix for baseline Logistic Regression on test set.

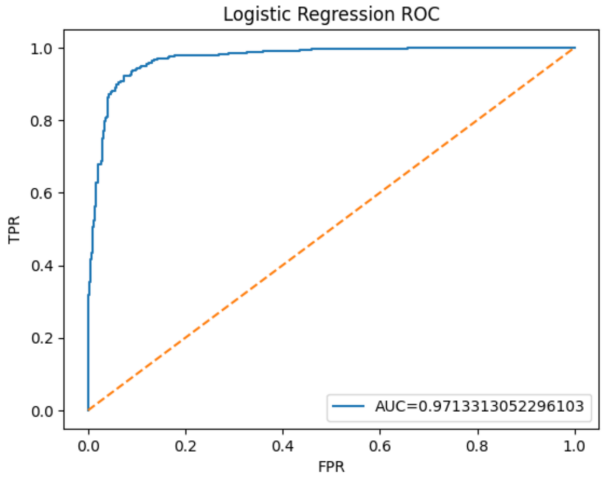

Fig. 4. ROC curve for baseline Logistic Regression with AUC value.

IV. Implementation Methodology

A. Preprocessing

The implementation follows these steps:

1) Load the Spambase CSV file into a pandas DataFrame.
2) Inspect the first few rows, dataset shape, and missing values to confirm data quality.
3) Separate features and target:
   - X: all columns except `class`.
   - y: `class` label (0 or 1).
4) Standardize features using `StandardScaler`. The standardized features X_{scaled} have zero mean and unit variance.
5) Split the dataset into training and testing sets with an 80:20 ratio using `train_test_split` and a fixed random seed for reproducibility.

B. Baseline Logistic Regression

A baseline Logistic Regression model is trained without extensive tuning:

- Model: `LogisticRegression()` with default parameters.
- Training time is recorded using wall-clock time.
- Predictions are made on the test set.
- Evaluation metrics include accuracy, precision, recall, and F1-score.
- A confusion matrix and ROC curve are generated to visualize performance.

C. Hyperparameter Tuning for Logistic Regression

Grid Search with 5-fold cross-validation is used to tune Logistic Regression over the search space:

- $C \in \{0.01, 0.1, 1, 10, 100\}$.
- Penalty $\in \{\texttt{l1}, \texttt{l2}\}$.
- Solver $\in \{\texttt{liblinear}, \texttt{saga}\}$.

The model with the best cross-validation accuracy is selected as the tuned Logistic Regression classifier.

D. SVM with Different Kernels

To analyze kernel behavior, SVM classifiers with four kernels (`linear`, `poly`, `rbf`, `sigmoid`) are trained with default hyperparameters:

- Each kernel-specific SVM is trained on the standardized training data.
- For each kernel, training time, accuracy, and F1-score on the test set are recorded.

E. Hyperparameter Tuning for SVM

SVM hyperparameters are tuned with Grid Search and 5-fold cross-validation using the search space:

- Kernel $\in \{\texttt{linear}, \texttt{poly}, \texttt{rbf}, \texttt{sigmoid}\}$.
- $C \in \{0.1, 1, 10, 100\}$.
- $\gamma \in \{\texttt{scale}, \texttt{auto}\}$.
- Degree $d \in \{2, 3, 4\}$ for polynomial kernels.

The best-performing SVM model is selected and evaluated on the test set.

F. Cross-Validation

To obtain robust estimates of generalization performance, 5-fold cross-validation is performed for the tuned Logistic Regression and tuned SVM models on the entire standardized dataset:

- 5-fold cross-validation scores are computed using `cross_val_score`.
- The mean cross-validation accuracy is reported for each model.

V. Results and Discussion

A. Baseline Logistic Regression Performance

Table I summarizes the baseline Logistic Regression performance on the test set.

TABLE I
BASELINE LOGISTIC REGRESSION PERFORMANCE ON TEST SET

Metric	Value
Accuracy	0.920
Precision	0.932
Recall	0.874
F1-score	0.902
Training time (s)	0.029

TABLE II
HYPERPARAMETER TUNING RESULTS (5-FOLD CV ACCURACY)

Model	Best Parameters	Best CV Accuracy
Logistic	$C = 10$, penalty = L1, solver = liblinear	0.928
SVM	$C = 1$, kernel = RBF, degree = 2, $\gamma =$ scale	0.931

TABLE III
SVM KERNEL-WISE PERFORMANCE ON TEST SET

Kernel	Accuracy	F1-score	Time (s)
Linear	0.925	0.909	0.347
Polynomial	0.764	0.629	0.300
RBF	0.935	0.921	0.294
Sigmoid	0.889	0.866	0.322

The baseline model already achieves high accuracy and F1-score, indicating that Logistic Regression is well-suited for this dataset when combined with feature standardization.

B. Hyperparameter Tuning Results

Grid Search hyperparameter tuning yields the best configurations shown in Table II.

The tuned Logistic Regression model prefers L1 regularization with a relatively high C, suggesting that a sparse model with weaker regularization yields the best cross-validation performance. The best SVM configuration uses an RBF kernel with moderate C and default scaling for γ, indicating that a non-linear decision boundary captures the underlying structure of the data better than linear or polynomial kernels.

C. SVM Kernel-wise Performance

Table III reports the test set performance of SVM with different kernels using default hyperparameters as implemented in the notebook.

The RBF kernel achieves the highest accuracy and F1-score, followed closely by the linear kernel. The polynomial kernel performs significantly worse, likely due to overfitting or poor alignment between polynomial decision boundaries and the data distribution. The sigmoid kernel performs moderately better than polynomial but worse than linear and RBF.

D. Final Tuned Model Performance

The tuned Logistic Regression and tuned SVM models are evaluated on the held-out test set. The resulting metrics are shown in Table IV.

TABLE IV
PERFORMANCE OF TUNED MODELS ON TEST SET

Model	Accuracy	Precision	Recall	F1-score
Logistic (tuned)	0.915	0.917	0.879	0.898
SVM (tuned RBF)	0.935	0.951	0.892	0.921

TABLE V
5-FOLD CROSS-VALIDATION ACCURACIES

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Logistic	0.920	0.932	0.896	0.950	0.824	0.904
SVM	0.933	0.934	0.950	0.949	0.850	0.923

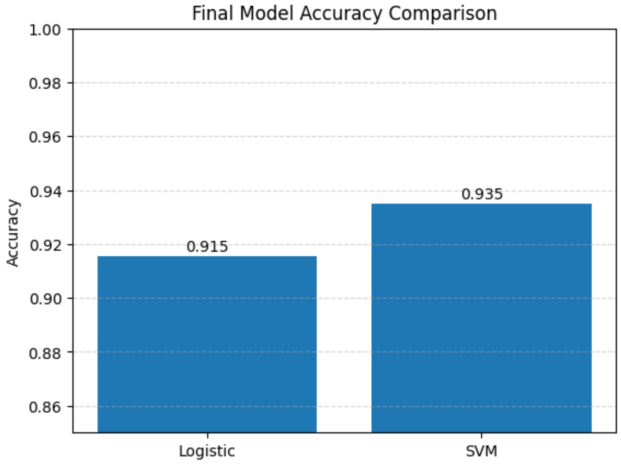

Fig. 5. Accuracy comparison between tuned Logistic Regression and tuned SVM.

The tuned RBF SVM achieves the best test performance among all models, improving slightly over the baseline Logistic Regression, particularly in terms of precision and F1-score. The tuned Logistic Regression shows similar recall but slightly lower precision and accuracy compared to the tuned SVM.

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

The 5-fold cross-validation accuracies for the tuned models on the full standardized dataset are:

- Logistic Regression (tuned): [0.9197, 0.9315, 0.8957, 0.9500, 0.8239], average 0.9041.
- SVM (tuned RBF): [0.9327, 0.9337, 0.9500, 0.9489, 0.8500], average approximately 0.9231.

The cross-validation results confirm that the tuned SVM generalizes better than the tuned Logistic Regression on average, with higher mean accuracy and more consistent performance across folds.

F. Overall Accuracy Comparison

A bar chart was created to compare the accuracy of the final tuned models.

TABLE VI
COMPARATIVE ANALYSIS OF LOGISTIC REGRESSION AND SVM

Criterion	Logistic Regression	SVM (RBF)
Accuracy (test)	Moderate to high	Highest
Model complexity	Low (linear boundary)	High (non-linear boundary)
Training time	Low	Higher but acceptable
Interpretability	High (weights analyzable)	Low (kernel-based decision)
Bias–variance trade-off	Higher bias, lower variance	Lower bias, higher variance

G. Comparative Analysis

Table VI summarizes a qualitative comparison between Logistic Regression and SVM.

Regarding the impact of regularization, the experiments show that an L1-regularized Logistic Regression with higher C balances model complexity and generalization. Stronger regularization (smaller C) may underfit, while too large C can lead to overfitting. L1 regularization also encourages sparse coefficients, effectively selecting influential features for spam detection.

For SVM, kernel behavior is critical. The linear kernel approximates a linear decision boundary and performs well when the data is nearly linearly separable after scaling. The RBF kernel introduces non-linear decision boundaries that capture more complex patterns, yielding the best performance. The polynomial and sigmoid kernels exhibit poorer performance on this dataset, likely due to mismatched model complexity or sensitivity to hyperparameter settings.

The bias–variance trade-off is evident in these results. Logistic Regression tends to have higher bias (simpler linear model) but lower variance, while RBF SVM has lower bias and higher variance. With appropriate regularization and hyperparameter tuning, the RBF SVM achieves a better balance, resulting in superior accuracy and F1-score.

VI. CONCLUSION

This experiment implemented and compared Logistic Regression and Support Vector Machine classifiers for spam email detection using the Spambase dataset. After standardizing the input features and performing EDA, baseline models were trained and evaluated. Logistic Regression with L1 regularization and an RBF-kernel SVM were tuned using grid search and 5-fold cross-validation.

The tuned RBF SVM achieved the best overall performance with a test accuracy of approximately 93.49% and F1-score of 0.921, outperforming both the baseline and tuned Logistic Regression models. Logistic Regression still offers competitive performance with simpler models and better interpretability, making it attractive when model transparency is important.

The study highlights the importance of feature scaling, regularization, kernel choice, and systematic hyperparameter tuning in building effective binary classifiers. The bias–variance trade-off observed between Logistic Regression and SVM emphasizes that the best model choice depends on both performance requirements and interpretability constraints.

VII. LEARNING OUTCOMES

Through this experiment, the following learning outcomes were achieved:

- Understanding of probabilistic classifiers (Logistic Regression) and margin-based classifiers (SVM).
- Practical experience in preprocessing numerical features and applying standardization.
- Ability to configure and execute grid search based hyperparameter tuning using cross-validation.
- Competence in evaluating classification models using metrics such as accuracy, precision, recall, F1-score, confusion matrices, ROC curves, and cross-validation scores.
- Insight into the impact of regularization and kernel selection on model performance and bias–variance trade-offs.