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 Laboratory		
Academic year	2025-2026 (Even)	Batch:2023-2027	**Due date: 27-01-2026**

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

Aim: 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:

#	Column	Non-Null Count	Dtype
0	word_freq_make	4601	float64
1	word_freq_address	4601	float64
2	word_freq_all	4601	float64
3	word_freq_3d	4601	float64
4	word_freq_our	4601	float64
5	word_freq_over	4601	float64
6	word_freq_remove	4601	float64
7	word_freq_internet	4601	float64
8	word_freq_order	4601	float64
9	word_freq_mail	4601	float64
...	...	...	...
56	capital_run_length_total	4601	int64
57	class	4601	int64

Table 1: Spam-base Dataset Description

Libraries used:

- **Pandas & NumPy:** For data manipulation, dataframe operations, and numerical analysis.
- **Matplotlib & Seaborn:** For data visualization, including statistical plots and model evaluation graphs.
- **Time:** For measuring the computational execution time of model training and testing.
- **Scikit-Learn (Preprocessing):** For feature scaling and normalization (`StandardScaler`).
- **Scikit-Learn (Model Selection):** For splitting datasets (`train_test_split`), cross-validation (`KFold`), and exhaustive hyperparameter tuning (`GridSearchCV`).
- **Scikit-Learn (Algorithms):** For implementing classification models, including Logistic Regression (`LogisticRegression`) and Support Vector Classification (`SVC`).
- **Scikit-Learn (Metrics):** For evaluating classification performance using Accuracy, Recall, Precision, and F1 Score, as well as visualizing results with Confusion Matrices (`ConfusionMatrixDisplay`) and ROC Curves (`RocCurveDisplay`).

Mathematical/theoretical description of the algorithm/objective performed:

- **Models:**
  - **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.
    - **Logistic Regression Hyperparameters:**
      - **C (Inverse Regularization Strength):** Controls the trade-off between model complexity and regularization.
      - **Solver:**
      - `liblinear`: Suitable for small datasets
      - `saga`: Efficient for large datasets
      - **Penality Regularization:**
      - `L1 Regularization:` Encourages sparsity by shrinking some coefficients exactly to zero.
      - `L2 Regularization:` Penalizes large weights but keeps all features.
  - **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.
    - **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
      - γ: Controls influence of a single training point
- **Preprocessing steps:**
  - **Handling null values:** Dataset doesn't have any null values
  - **Transformation:** We observe columns with outliers causing a heavily right skewed distribution like `capital_run_length_average`, `capital_run_length_longest`, `capital_run_length_total`. We apply a log transform on these features to make them more distributed and reduce the effect of outliers.

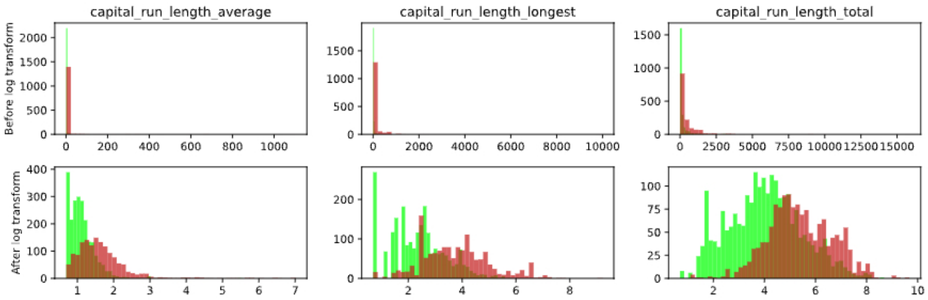

Figure 1: Analysis of skew

- **Scaling:** Used `StandardScaler` to scale parameters. It transforms parameters based on the mean and variance of the dataset, resulting in a distribution with mean 0 and variance 1. This is a good Scaler for regression tasks

$$x_{scaled} = \frac{x_{original} - \mu_x}{\sigma_x}$$

- **Exploratory Data Analysis:**

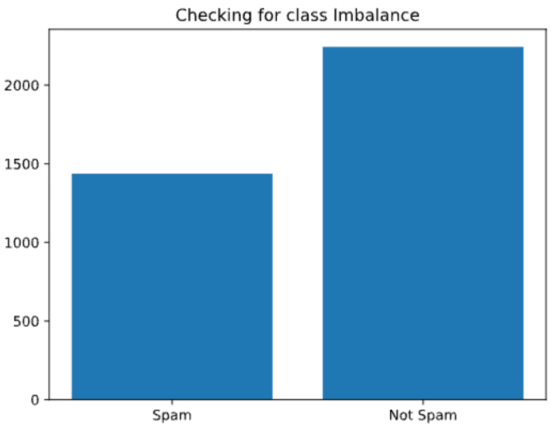

Figure 2: Distribution

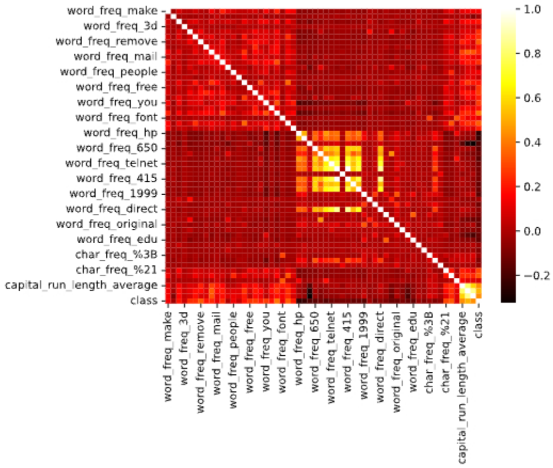

Figure 3: Heatmap

- From the Heatmap, we observe that none of the input features are highly correlated to each other, we also not high correlation between target class and capital_run_length features. From the bar graph, we observe that the distribution of classes spam/not spam are balanced.

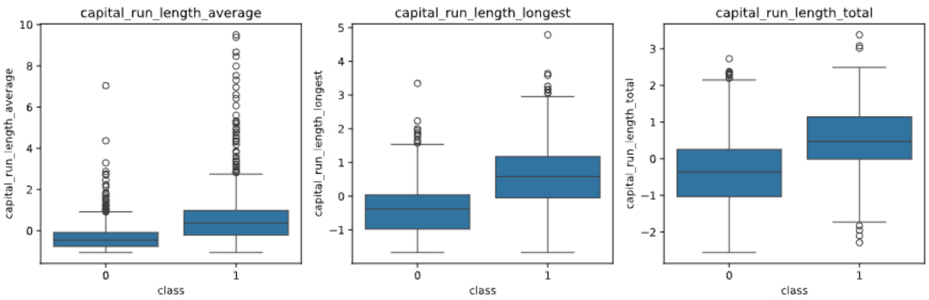

Figure 4: Analysis of distribution of capital_run_length variables

- From the box plots we confirm the significance of capital_run_length features in distinguishing the two classes.

- **Performance Metrics:**
  - Accuracy:
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
  - Precision:
$$Precision = \frac{TP}{TP + FP}$$
  - Recall:
$$Recall = \frac{TP}{TP + FN}$$
  - F1 Score:
$$F1Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Results and Discussions:

- **Visualizations:**

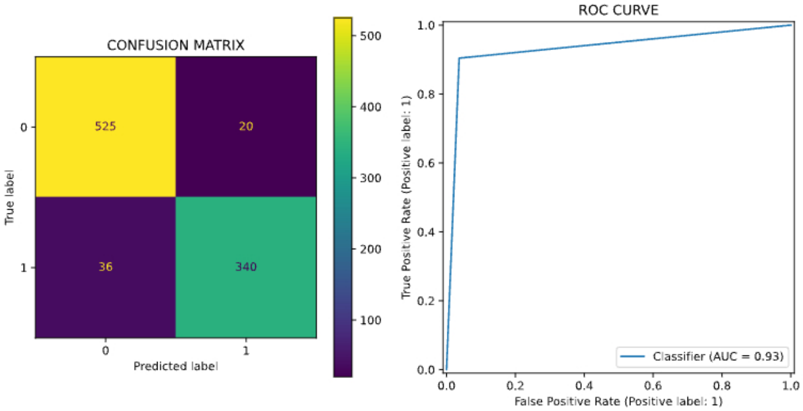

Figure 5: Baseline Logistic Regression results

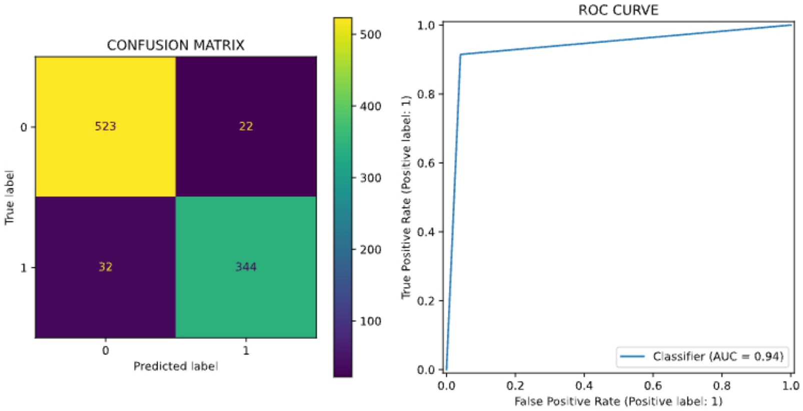

Figure 6: Finetuned Logistic regression results

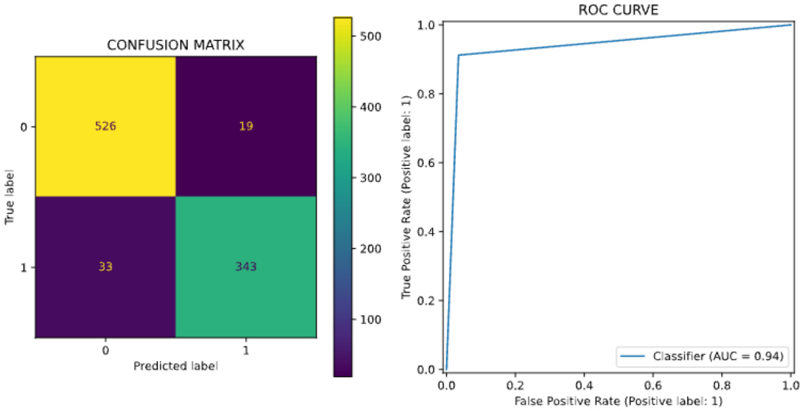

Figure 7: Linear SVC results

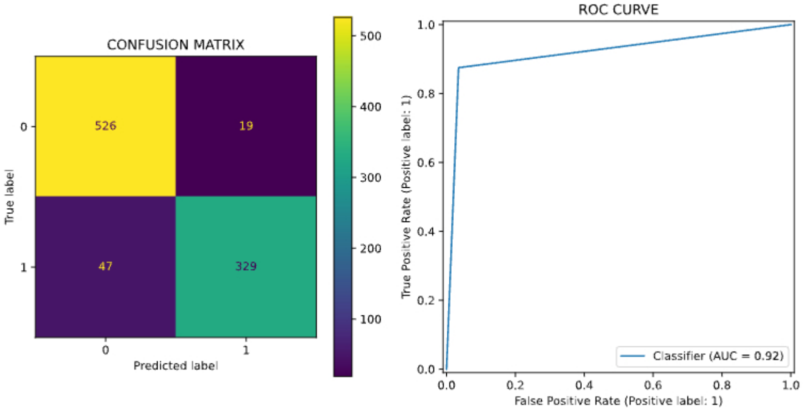

Figure 8: Polynomial SVC results

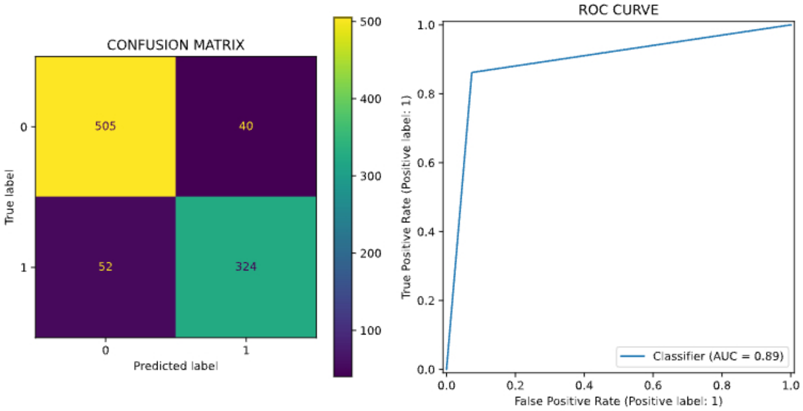

Figure 9: Sigmoid SVC results

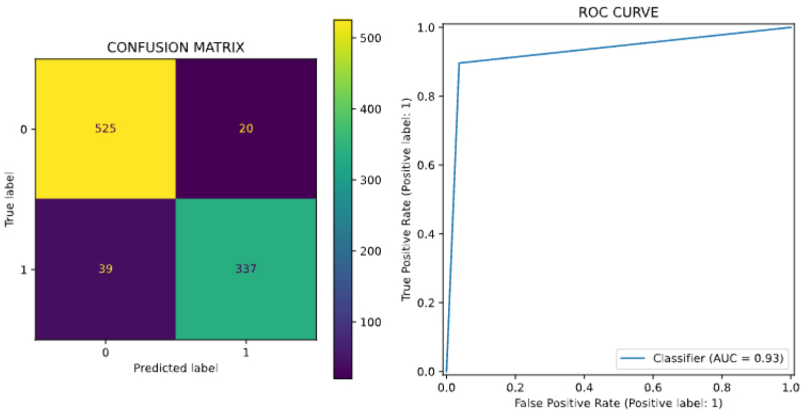

Figure 10: RBF SVC results

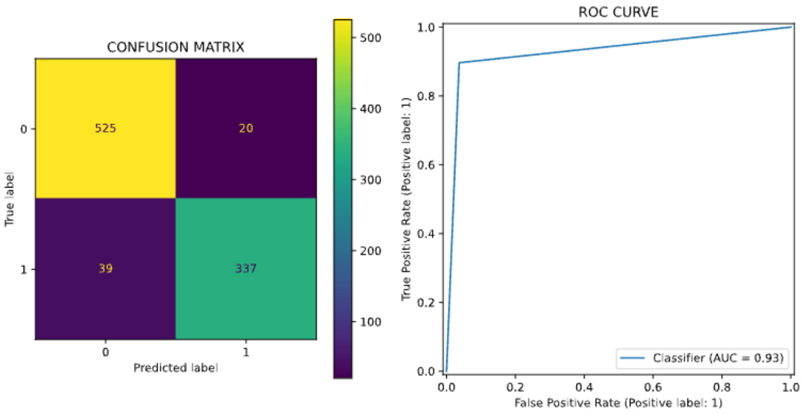

Figure 11: Finetuned SVC results

- **Hyperparameter Tuning Results**

Model	Search Method	Best Parameters	Best CV Accuracy

Logistic Regression	Grid	C=100
		penalty=l2
		solver=liblinear
			0.9118

SVM	Grid	C=10
		gamma=scale
		kernel=rbf
			0.9183

- **Logistic Regression Performance**

Metric	Value
Accuracy	0.9414
Precision	0.9399
Recall	0.9149
F1 Score	0.9272
Training Time (ms)	1.7ms

- **SVM Kernel-wise Performance**

Kernel	Accuracy	F1 Score	Training Time (ms)
Linear	0.9435	0.9295	23.9
Polynomial	0.9283	0.9088	45
RBF	0.9359	0.9195	122.2
Sigmoid	0.9001	0.8757	48.0

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

Fold	Logistic Regression	SVM
Fold 1	0.9457	0.9470
Fold 2	0.9212	0.9280
Fold 3	0.9293	0.9361
Fold 4	0.9375	0.9348
Fold 5	0.9255	0.9416
Average	0.9312	0.9375

- **Comparative Analysis**

Criterion	Logistic Regression	SVM
Accuracy	0.9414	0.9359
Model Complexity	Low	High
Training Time	Low	High
Interpretability	High	Low

Learning Practices:

- Learned pre-processing and EDA steps for a classification task.
- Created Logistic Regression model for classification and performed hyperparameter tuning on the same.
- Created SVM classifier with linear, polynomial, sigmoid and rbf kernels and performed hyperparameter tuning on the same.