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		

R Padmashri
3122 23 5001 093

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, evaluate their performance, and analyze the effect of hyperparameter tuning on classification accuracy.

Dataset

The Spambase dataset consists of numerical features extracted from email content and a binary class label indicating spam or non-spam emails.

Dataset reference:

- Kaggle: Spambase Dataset

Exploratory Data Analysis

Required Imports

Listing 1: Importing required libraries

```python
import time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV, KFold
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score,
    f1_score, confusion_matrix, roc_curve, auc
```

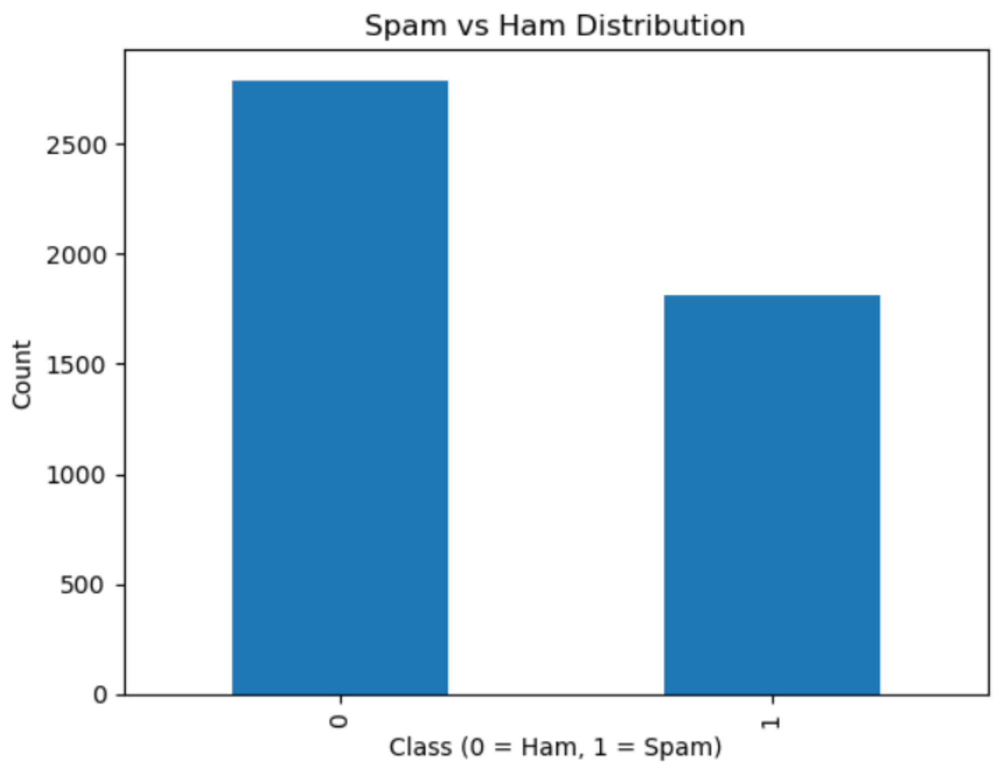

Figure 1: Class Distribution of Spam and Ham Emails

Data Preprocessing

Listing 2: Feature scaling using StandardScaler

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Baseline Logistic Regression

Listing 3: Baseline Logistic Regression Model

```
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
y_pred = log_reg.predict(X_test)
```

Table 1: Logistic Regression Performance

Metric	Value
Accuracy	0.93
Precision (Weighted Avg)	0.93
Recall (Weighted Avg)	0.93
F1 Score (Weighted Avg)	0.93

SVM Kernel-wise Performance

Table 2: Kernel-wise Performance of SVM

Kernel	Accuracy	F1 Score
Linear	0.9305	0.9106
Polynomial	0.7795	0.6219
RBF	0.9272	0.9055
Sigmoid	0.8838	0.8524

Hyperparameter Tuning

Logistic Regression Tuning

Listing 4: Grid Search for Logistic Regression

```
param_grid_lr = {
    "C": [0.01, 0.1, 1, 10, 100],
    "penalty": ["l1", "l2"],
    "solver": ["liblinear"]
}
```

SVM Tuning

Listing 5: Grid Search for SVM

```
param_grid_svm = {
    "C": [0.1, 1, 10, 100],
    "kernel": ["linear", "poly", "rbf", "sigmoid"],
    "gamma": ["scale", "auto"]
}
```

Hyperparameter Tuning Results

Model	Best Parameters	Best CV Accuracy
Logistic Regression	$C = 100$, penalty=l1, solver=liblinear	0.9239
SVM	$C = 1$, kernel=rbf, γ=scale	0.9340

Table 3: Hyperparameter Tuning Summary

Visual Analysis

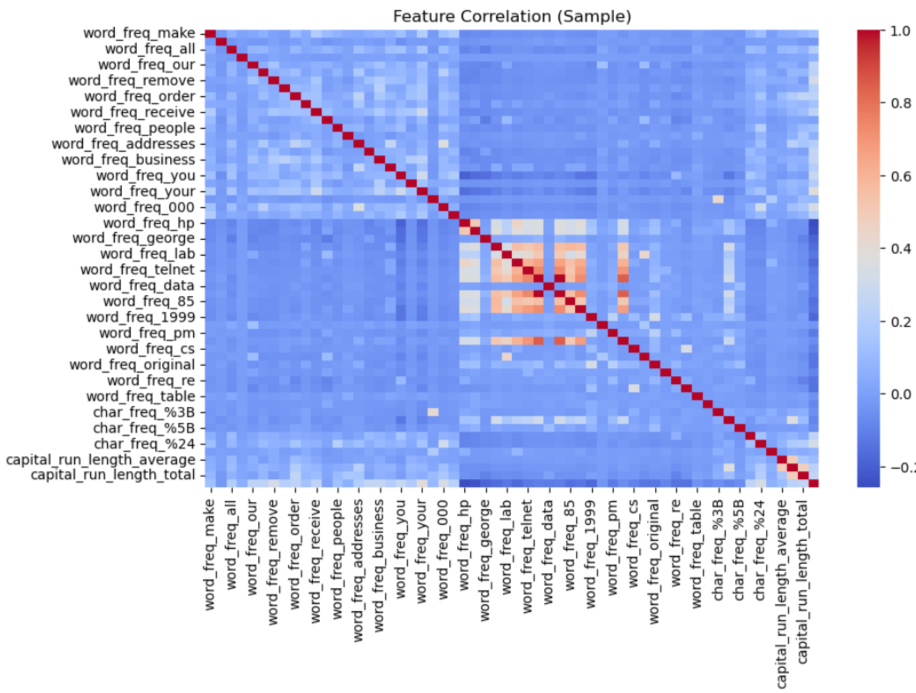

Figure 2: Confusion Matrix for Best Performing Model

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

Table 4: 5-Fold Cross-Validation Accuracy

Fold	Logistic Regression	SVM
Fold 1	0.9196	0.9327
Fold 2	0.9315	0.9337
Fold 3	0.8956	0.95
Fold 4	0.9510	0.9489
Fold 5	0.8239	0.85
Average	0.9043	0.9231

Observations

- Logistic Regression provides interpretable results with fast training.
- SVM achieves higher accuracy for non-linear kernels.
- Hyperparameter tuning improves overall classification performance.

Learning Outcomes

- Understood probabilistic and margin-based classifiers
- Implemented Logistic Regression and SVM
- Applied hyperparameter tuning techniques
- Evaluated binary classification models using standard metrics

References

- Scikit-learn: Logistic Regression
- Scikit-learn: Support Vector Machines
- Kaggle: Spambase Dataset