24CY731 DATA MINING AND MACHINE LEARNING IN CYBER SECURITY
Lab 4 - Classifying Email as Spam or Not Spam using SVM

Karthikeyan G
Roll No: CB.SC.P2.CYS24008

January 29, 2025

1 Classifying Email as Spam or Not Spam using SVM

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score,
    confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv('/kaggle/input/email-spam-classification-dataset-csv/emails.
    csv', index_col='Email No.')  # Setting 'Email No.' as index
print("Preview of Data:\n", data.head())  # Display first few rows

# Separating features (X) and target (y)
features = data.drop(columns=['Prediction'])  # Removing target column from
    features
target = data['Prediction']  # Assigning target variable

# Splitting data into training and testing (80-20 split)
X_train, X_test, y_train, y_test = train_test_split(
    features, target,
    test_size=0.2,  # Keeping 20% for testing
    random_state=42  # Ensuring reproducibility
)

# Standardizing features (important for SVM)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)  # Fit on training data and
    transform
X_test_scaled = scaler.transform(X_test)  # Only transform test data

# Training SVM model with linear kernel
svm_classifier = SVC(kernel='linear', random_state=42)  # Using linear kernel
svm_classifier.fit(X_train_scaled, y_train)  # Training the model

# Making predictions on test data
predictions = svm_classifier.predict(X_test_scaled)

# Evaluating model performance
print("\nConfusion Matrix:")
conf_matrix = confusion_matrix(y_test, predictions)  # Creating confusion matrix
print(conf_matrix)

print("\nClassification Report:")
print(classification_report(y_test, predictions))  # Detailed performance
    metrics

accuracy = accuracy_score(y_test, predictions) * 100  # Calculating accuracy
print("\nModel Accuracy: ", accuracy)

# Visualising confusion matrix using heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Not Spam', 'Spam'],
            yticklabels=['Not Spam', 'Spam'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

```
# Plotting accuracy as a bar graph
plt.figure(figsize=(6, 4))
plt.bar(['Accuracy'], [accuracy], color='skyblue')  # Showing accuracy
    percentage
plt.ylim(0, 100)
plt.title('Model Accuracy (%)')
plt.ylabel('Percentage')
plt.show()

# Plot to show count of spam and non-spam emails
plt.figure(figsize=(6, 4))
sns.countplot(x=target, palette='Set1')  # Counting spam (1) vs not spam (0)
plt.title('Distribution of Spam vs Not Spam')
plt.xlabel('Spam (1) vs Not Spam (0)')
plt.ylabel('Frequency')
plt.show()
```

1.1 Explanation of the Code

- **Previewing the Data**: We display the first few rows of the dataset to get an idea of its structure and contents.
- **Separating Features and Target**: The features (input variables) are separated from the target variable (Prediction). This step is crucial for training the model.
- **Splitting the Data**: The data is split into training and testing sets with an 80-20 ratio. This ensures that the model is trained on a majority of the data but still has enough data to test its performance.
- **Standardizing Features**: The features are standardized using ‘StandardScaler‘. Standardization is important for SVM as it ensures that all features contribute equally to the model.
- **Training the SVM Model**: An SVM model with a linear kernel is trained on the standardized training data. The linear kernel is chosen for its simplicity and effectiveness in this context.
- **Making Predictions**: The trained model makes predictions on the test data. These predictions are then used to evaluate the model’s performance.
- **Evaluating the Model**: The model’s performance is evaluated using a confusion matrix, classification report, and accuracy score. These metrics provide a comprehensive view of how well the model is performing.
- **Visualizing Results**: The confusion matrix is visualized using a heatmap to show the true positive, true negative, false positive, and false negative counts. The model accuracy is plotted as a bar graph, and the distribution of spam vs. not spam emails is shown using a count plot.

```
Preview of Data:
           the  to  ect  and  for  of    a  you  hou  in  ...  connevey  jay  \
Email No.                                                  ...
Email 1      0   0    1    0    0   0    2    0    0   0  ...         0    0
Email 2      8  13   24    6    6   2  102    1   27  18  ...         0    0
Email 3      0   0    1    0    0   0    8    0    0   4  ...         0    0
Email 4      0   5   22    0    5   1   51    2   10   1  ...         0    0
Email 5      7   6   17    1    5   2   57    0    9   3  ...         0    0

           valued  lay  infrastructure  military  allowing  ff  dry  \
Email No.
Email 1         0    0               0         0         0   0    0
Email 2         0    0               0         0         0   1    0
Email 3         0    0               0         0         0   0    0
Email 4         0    0               0         0         0   0    0
Email 5         0    0               0         0         0   1    0

           Prediction
Email No.
Email 1             0
Email 2             0
Email 3             0
Email 4             0
Email 5             0

[5 rows x 3001 columns]
```

Figure 1: Preview of the Dataset: This figure shows the first few rows of the email dataset, providing a glimpse of the features and the target variable.

```
Confusion Matrix:
[[707  32]
 [ 23 273]]

Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.96      0.96       739
           1       0.90      0.92      0.91       296

    accuracy                           0.95      1035
   macro avg       0.93      0.94      0.94      1035
weighted avg       0.95      0.95      0.95      1035


Model Accuracy:  94.68599033816425
```

Figure 2: Model Evaluation: This figure displays the classification report, including precision, recall, F1-score, and support for both spam and not spam classes.

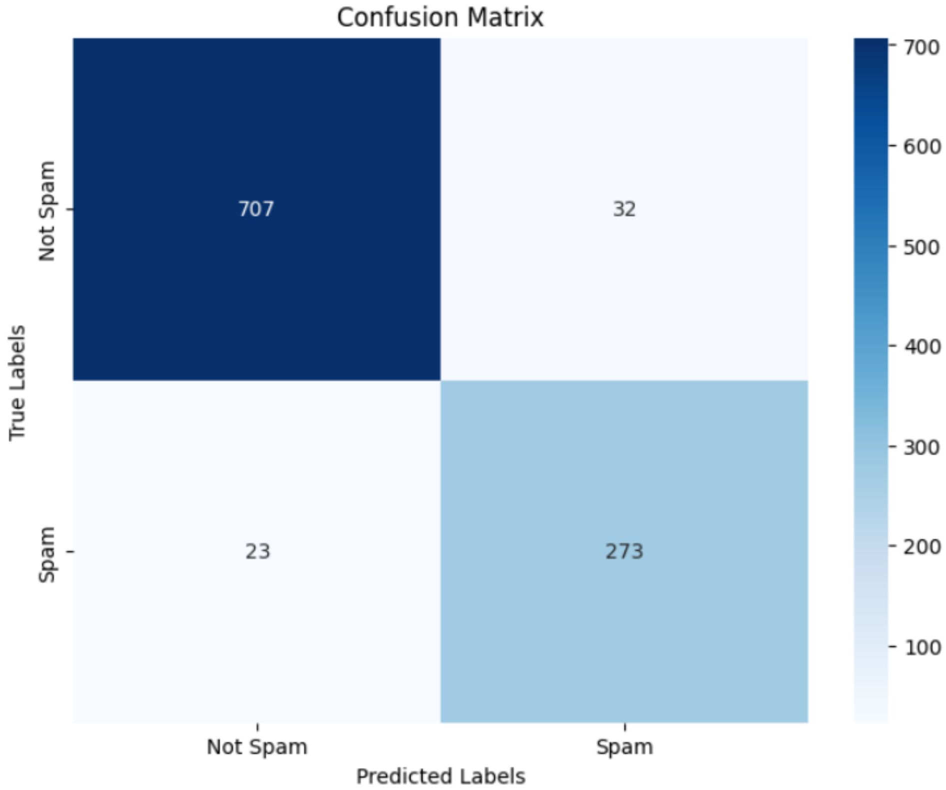

Figure 3: Confusion Matrix Visualization: This heatmap represents the confusion matrix, showing the true positive, true negative, false positive, and false negative counts.

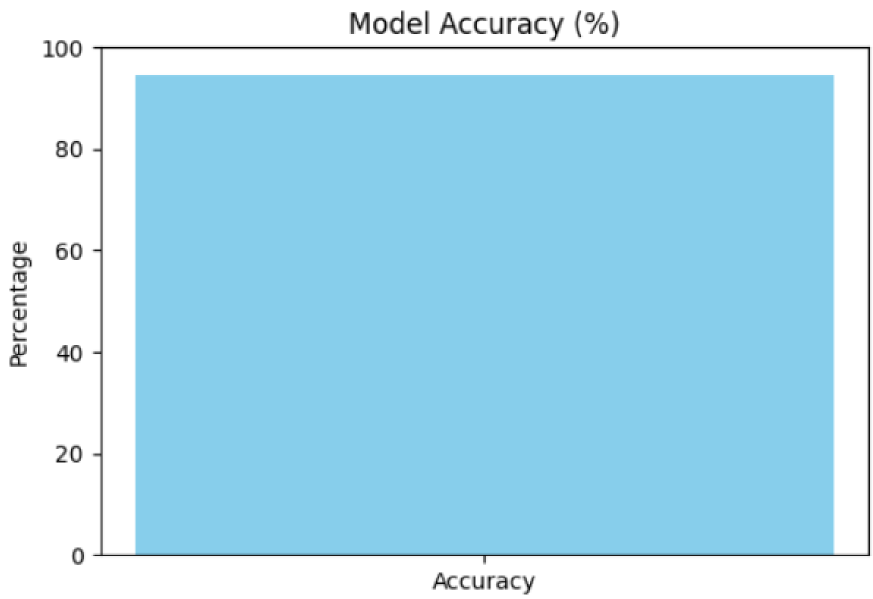

Figure 4: Model Accuracy: This bar graph illustrates the accuracy of the SVM model, indicating the percentage of correct predictions.

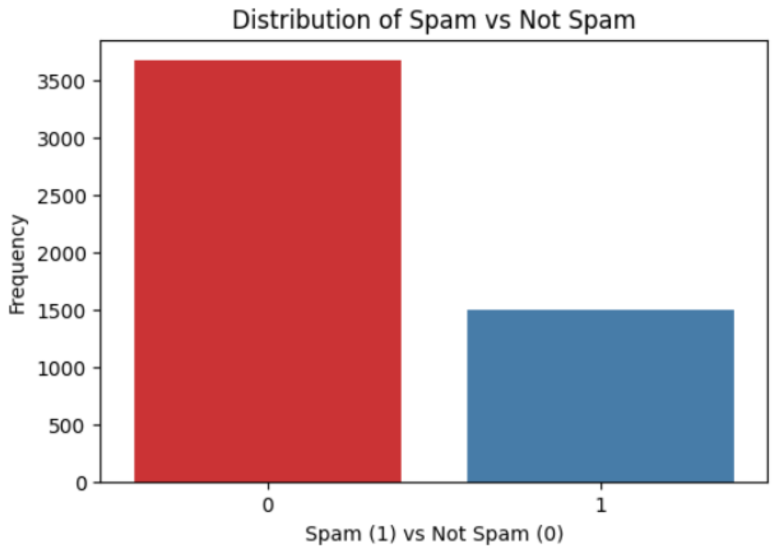

Figure 5: Distribution of Spam and Not Spam in the Dataset: This count plot shows the frequency of spam and not spam emails in the dataset, providing insight into the class distribution.