Model Evaluation Report: Spam Email Classification using SVM with RBF Kernel

---

Problem Statement:

The task is to build a machine learning model capable of classifying emails into two categories: **Spam** and **Not Spam**. The model was trained using various features extracted from email content, such as subject length, body length, presence of links, and specific keywords that are commonly associated with spam emails. The goal was to predict whether a given email is spam or not based on its characteristics.

Approach:

1. **Dataset Preparation:**
    - **Feature Extraction**: The dataset included various features, such as:
        - **Subject Length**: Length of the email subject line.
        - **Body Length**: Length of the email body.
        - **Number of Links**: Count of hyperlinks within the email body.
        - **Has Attachments**: Whether the email contains any attachments.
        - **Spam Keywords Count**: Number of spam-related keywords (e.g., "gift card", "free", "promotion").
        - **Contains Suspicious Words**: Presence of words commonly associated with phishing or spam (e.g., "click here", "limited time").
        - **Sender Reputation Score**: An arbitrary score based on the sender's history.
        - **Time of Day**: The time the email was sent (morning/afternoon).
        - **Email Domain**: Domain of the sender's email address.
        - **Recipient Address Count**: Number of recipients.
        - **Unsubscribe Link Present**: Whether the email contains an "unsubscribe" link.
    - **Textual Features**: The body of the email was processed using the **TF-IDF Vectorizer** to extract text-based features, which were later added to the model as part of the feature set.
2. **Data Preprocessing:**
    - **Scaling**: The numerical features were scaled using the **StandardScaler** to ensure all features had the same scale for the SVM to perform effectively.

    - **Feature Combination**: A combination of the extracted features (both numerical and textual) was used to form the final input data for the model.
3. **Model Training:**
    - **Support Vector Machine (SVM)** with the **Radial Basis Function (RBF) Kernel** was chosen as the classifier.
    - The model was trained using the training dataset and evaluated using a separate test dataset.
4. **Model Evaluation**: The trained model was evaluated using accuracy and various classification metrics such as precision, recall, and F1-score.

Results:

Best RBF Kernel Accuracy:

- **Accuracy**: **0.505**
    - The model's overall accuracy is **50.5%**, which indicates that the model's predictions are not much better than random guessing.

Classification Report:

The detailed classification report is as follows:

Metric	Class 0 (Not Spam)	Class 1 (Spam)	Overall
Precision	0.57	0.45	0.51
Recall	0.50	0.51	0.51
F1-Score	0.53	0.48	0.50
Support	112	88	200

Key Insights:

1. **Precision**:
    - **Class 0 (Not Spam)**: The model has a **57% precision**, meaning that when it predicts an email as **not spam**, it is correct **57% of the time**.
    - **Class 1 (Spam)**: The model has a **45% precision**, meaning it correctly identifies spam emails **45% of the time**. This is lower than the precision for non-spam emails.
2. **Recall**:

    - **Class 0 (Not Spam)**: The model correctly identifies **50% of the actual non-spam emails**, with a **50% recall**.
    - **Class 1 (Spam)**: The model correctly identifies **51% of the actual spam emails**, with a **51% recall**.
3. **F1-Score**:
    - **Class 0 (Not Spam)**: The **F1-score** for non-spam emails is **0.53**, which indicates that the model is moderately effective in identifying non-spam emails but still leaves room for improvement.
    - **Class 1 (Spam)**: The **F1-score** for spam emails is **0.48**, showing that the model performs less well on identifying spam emails compared to non-spam.
4. **Overall Accuracy**:
    - The overall **accuracy** of **51%** indicates that the model is only slightly better than random guessing. This suggests that the model needs significant improvements in terms of feature engineering, data handling, and model selection.

Macro Average:

- The **macro average** of precision, recall, and F1-score is **0.51**, meaning the model performs similarly for both spam and non-spam classes, but the overall performance is subpar.

Weighted Average:

- The **weighted average** for precision, recall, and F1-score is **0.51**, accounting for the class imbalance and showing that the model is not effectively distinguishing between spam and non-spam emails.

Challenges Faced:

1. **Feature Engineering**:
    - Despite adding several meaningful features, the model performance is still low. Some features, such as **Spam Keywords Count** and **Contains Suspicious Words**, may not have had enough discriminative power to improve model performance significantly.
2. **Data Imbalance**:
    - There could be an imbalance between the number of spam and non-spam emails in the dataset, leading to biased predictions. This

might have affected the performance of the model, especially in predicting the minority class (spam).

3. **Model Complexity**:
    - The **SVM with RBF kernel** might not have been the best model for this problem. The RBF kernel, while effective for certain types of data, might not have captured the underlying patterns in the email data effectively.
4. **Hyperparameter Tuning**:
    - The performance of the SVM model could be improved by tuning its hyperparameters, such as **C** (penalty) and **gamma** (kernel coefficient), which control the decision boundary of the model.

Future Work and Improvements:

1. **Improved Feature Selection and Engineering**:
    - Add more informative features, such as the **sender's reputation**, **sender's domain**, **email metadata**, or even external features like social media activity of the sender.
    - Investigate other types of textual features, like word embeddings (e.g., **Word2Vec** or **GloVe**) or more advanced text preprocessing.
2. **Addressing Data Imbalance**:
    - Use **SMOTE** (Synthetic Minority Over-sampling Technique) or **class weights** in the SVM to deal with class imbalance.
    - Consider **undersampling** the majority class or **oversampling** the minority class during training.
3. **Model Experimentation**:
    - Experiment with other machine learning algorithms like **Random Forests**, **Gradient Boosting**, or **Logistic Regression**.
    - Explore **Deep Learning** models like **LSTM** or **CNN** for text classification, which are known to perform well with unstructured text data.
4. **Hyperparameter Tuning**:
    - Perform **Grid Search** or **Random Search** to tune the SVM hyperparameters (e.g., **C**, **gamma**).
5. **Evaluation Metrics**:
    - Evaluate using additional metrics such as **ROC-AUC** to assess the model's ability to distinguish between spam and non-spam emails.
6. **Cross-Validation**:
    - Implement **K-fold cross-validation** to assess the robustness of the model and mitigate the risk of overfitting.

Conclusion:

While the SVM model with the RBF kernel provided a starting point for classifying spam emails, the results show that the model is underperforming, with an accuracy of only **51%**. Future improvements should focus on enhancing feature engineering, addressing data imbalance, and experimenting with more advanced models and hyperparameter tuning. With these adjustments, we can improve the model's ability to accurately classify spam and non-spam emails.