SPAM EMAIL CLASSIFICATION

SOHINI MANDAL

1. Introduction

Spam email classification is a fundamental application of Natural Language Processing (NLP) and supervised machine learning, aimed at automatically identifying unsolicited or malicious emails and separating them from legitimate communication. With the rapid growth of digital communication, effective spam filtering systems are essential to improve user experience and reduce security risks such as phishing and fraud.

This project focuses on building an end-to-end spam email classification pipeline using classical machine learning algorithms. The workflow includes data cleaning, exploratory data analysis, text preprocessing, feature extraction using TF-IDF, training multiple classification models with hyperparameter tuning, and evaluating their performance using appropriate metrics. Ensemble techniques such as Voting and Stacking classifiers are also explored to improve predictive performance.

2. Data in Hand

Dataset Description

- **Dataset Name:** Spam Email Dataset
- **Source:** Kaggle
- **Target Variable:**
    - 1 → Spam
    - 0 → Not Spam (Ham)

Initial Columns

- **text:** Raw email content
- **spam:** Spam label

Data Cleaning Process

The following data cleaning steps were performed:

1. Only the text and spam columns were retained.
2. Rows containing missing values were removed, resulting in 5,728 records.
3. Duplicate rows were removed, reducing the dataset to 5,695 records.
4. Two rows containing textual values in the spam column were removed.
5. The final dataset consists of 5,693 email records.

6. The spam column was converted into numeric binary labels using LabelEncoder.

Class Distribution

The dataset is imbalanced, with ham emails significantly outnumbering spam emails. This imbalance influenced the selection of evaluation metrics and model comparison strategies.

3. Exploratory Data Analysis (EDA)

To understand the structural characteristics of the email data, several length-based features were engineered:

- Number of characters
- Number of words
- Number of sentences

Key Observations

- Spam and ham emails exhibit positively skewed distributions across all length-based features.
- Significant overlap exists between spam and ham emails in the short-to-medium length range.
- Pair plots reveal strong correlations between character count and word count, indicating redundancy among length-based features.
- Correlation analysis shows almost no linear relationship between length-based features and the spam label.

These observations indicate that email length alone is insufficient for effective spam classification, motivating the use of content-based text representations.

4. Text Preprocessing

Each email message was preprocessed using the following steps:

1. Removal of subject prefixes
2. Conversion to lowercase
3. Tokenization using NLTK
4. Removal of non-alphanumeric tokens
5. Removal of stopwords and punctuation
6. Stemming using the Porter Stemmer

The processed output was stored in a new column named transformed_text.

Text Analysis

- Word clouds were generated separately for spam and ham emails.
- Frequency analysis of the top 30 most common words revealed distinct vocabulary usage between spam and ham emails.

This analysis provided descriptive insight into word usage patterns across classes.

5. Methodology

Feature Extraction

- TF-IDF Vectorization
    - Maximum number of features: 3,000
    - The TF-IDF vectorizer was fitted only on the training data to prevent data leakage.
- Scaling
    - Min–Max scaling was applied to the TF-IDF feature matrices to ensure compatibility with distance-based and gradient-based models.

Train–Test Split

- The dataset was split into 80% training and 20% testing sets.
- Stratified sampling was used to preserve class proportions.
- A fixed random state ensured reproducibility.

Cross-Validation and Evaluation Metric

- Stratified K-Fold cross-validation (k = 4) was used during hyperparameter tuning.
- Precision was selected as the primary scoring metric.

Precision was prioritized because, in spam filtering systems, false positives are more costly than false negatives. Incorrectly classifying a legitimate email as spam may result in the loss of important information, whereas allowing a limited number of spam emails into the inbox is comparatively less harmful. Therefore, maximizing precision ensures that emails classified as spam are highly likely to be truly spam.

6. Model Building and Evaluation

Models Evaluated

The following machine learning models were trained and evaluated:

- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Bernoulli Naive Bayes
- Complement Naive Bayes
- Logistic Regression
- Support Vector Classifier

- K-Nearest Neighbors
- Decision Tree
- Random Forest
- AdaBoost
- Bagging
- Extra Trees
- Gradient Boosting
- XGBoost

Hyperparameters were optimized using GridSearchCV for each model**.**

Key Results

- Multinomial Naive Bayes achieved the highest precision among Naive Bayes variants.
- SVC, KNN, Logistic Regression, Random Forest, and Extra Trees achieved very high precision but suffered from poor recall, resulting in low F1-scores.
- Gradient Boosting and XGBoost provided a more balanced trade-off between precision and recall.

Due to class imbalance, accuracy alone was found to be misleading; therefore, precision and F1-score were emphasized for model comparison.

7. Ensemble Learning

Voting Classifier

A soft voting classifier was constructed using:

- Multinomial Naive Bayes
- Gradient Boosting
- XGBoost

Performance:

- Accuracy: 0.9886
- Precision: 0.9711
- F1-Score: 0.9764

Stacking Classifier

A stacking classifier was built using:

- Base learners: Multinomial Naive Bayes, Gradient Boosting, XGBoost
- Meta-learner: Logistic Regression

Performance:

- Accuracy: 0.9895
- Precision: 0.9712
- F1-Score: 0.9783

ROC Curve Analysis

The ROC curve for the stacking classifier lies close to the top-left corner, with an AUC value close to 1.00, indicating strong discriminative capability between spam and ham emails.

8. Conclusion

Among all the evaluated models, the Stacking Classifier, which integrates Multinomial Naive Bayes, Gradient Boosting, and XGBoost with a Logistic Regression meta-learner, achieved the best overall performance.

The model attained a test accuracy of **0.9895**, a precision of **0.9712,** and an F1-score of **0.9783**, demonstrating a strong balance between accurately identifying spam emails and minimizing false positives.

The current approach relies on TF-IDF-based feature representation. While effective, TF-IDF ignores word order and semantic context, limiting its ability to capture meaning conveyed through phrases, sarcasm, or sentence structure. Additionally, the model depends on a fixed vocabulary learned during training, which may reduce its ability to adapt to evolving spam patterns and newly emerging terms (concept drift). The approach is also limited to English-language emails and does not explicitly incorporate important metadata such as email headers, URLs, or sender information.

Future work can explore broader hyperparameter tuning, deep learning-based approaches, alternative word embedding techniques, class imbalance handling methods, and cost-sensitive or adaptive learning strategies to further improve the robustness and generalization of the spam classification system.