Message Spam Classifier Machine Learning Model

Aditya Dighe, Yuvraj Rasal, Amartya Mishra

Contributing authors: ardighe7@gmail.com; yuvraj8918@gmail.com;
amartyamishra2503@gmail.com;

Abstract

In the realm of message spam detection, the proliferation of various communication channels has amplified the need for robust classifiers to discern between genuine messages and spam. This project presents an innovative approach to tackle this challenge by leveraging advanced techniques including tokenization, stashing, and ensemble learning through voting classifiers.

Initially, the messages are tokenized to extract meaningful features, enabling the model to discern patterns indicative of spam. Stashing, a novel concept introduced in this work, involves strategically withholding a portion of the training data to serve as a validation set, enhancing model generalization. The ensemble of classifiers, facilitated by voting mechanisms, harnesses the collective intelligence of diverse algorithms to improve classification accuracy and resilience to diverse spam tactics.

The trained model, encapsulated into a pickle file (pkl), is seamlessly integrated into a web interface. Users can input messages through a text area on the website, where the model swiftly processes the text and classifies it as spam or genuine. This end-to-end solution provides an efficient and effective means of spam detection, empowering users to mitigate the onslaught of unwanted messages across various communication platforms.

1 Introduction

In our digitally interconnected world, the incessant barrage of unsolicited messages has become an unavoidable facet of online communication. From email inboxes flooded with promotional offers to social media feeds cluttered with spammy comments, the pervasiveness of spam poses significant challenges for both individuals and organizations alike. Beyond mere annoyance, spam presents real risks, including privacy breaches, financial scams, and even malware distribution. As such, the need for effective spam detection systems has never been more critical. Traditional methods, such as

rule-based filters and keyword matching, are often insufficient to accurately distinguish between genuine messages and spam, highlighting the necessity for more sophisticated solutions.

In response to the growing demand for spam detection, the market has witnessed a proliferation of products leveraging machine learning (ML) algorithms to combat spam. Major email service providers have integrated ML-powered spam filters into their platforms, continually refining these algorithms based on user feedback to enhance detection accuracy. Social media platforms have also adopted ML techniques to identify and filter out spammy content, preserving the integrity of user interactions and safeguarding against malicious activities. Additionally, standalone software solutions and APIs offer customizable spam detection capabilities, catering to businesses and individuals seeking tailored solutions for diverse communication channels.

While existing products have made significant strides in spam detection, several challenges persist. The dynamic nature of spam tactics necessitates continuous adaptation and innovation in spam classification techniques. Moreover, the proliferation of diverse communication channels—from emails to messaging apps—underscores the need for versatile spam detection solutions capable of addressing varying contexts and formats. Furthermore, the inherent trade-off between detection accuracy and false positive rates remains a key consideration, highlighting the importance of fine-tuning ML models to strike the right balance.

In light of these challenges, this project proposes an innovative approach to spam classification, incorporating advanced techniques such as tokenization, stashing, and ensemble learning. By tokenizing messages to extract meaningful features, the model gains deeper insights into the underlying patterns indicative of spam. The concept of stashing introduces a novel mechanism for validation data withholding, enhancing model generalization and mitigating overfitting. Ensemble learning, facilitated through voting classifiers, harnesses the collective intelligence of diverse algorithms to improve classification accuracy and resilience to evolving spam tactics. The integration of these techniques into a web-based interface, allowing users to seamlessly classify messages in real-time, underscores the practical applicability and user-centric design of the proposed solution.

2 Methodology

2.1 Tokenization

Tokenization is the initial step in our approach, involving the segmentation of text data into individual tokens or words. This process enables the extraction of meaningful features essential for spam classification. We will employ Python libraries such as NLTK or spaCy for tokenization, ensuring efficient processing of text data. Additionally, custom tokenization algorithms will be implemented to handle various linguistic nuances, including word boundaries, punctuation, and special characters.

2.2 Voting Classifiers

Voting classifiers will be utilized as part of ensemble learning techniques to harness the collective intelligence of multiple base classifiers for improved spam classification accuracy. Diverse base classifiers, such as Random Forest, Support Vector Machines (SVM), and Naive Bayes, will be selected to capture different aspects of the data and mitigate individual classifier biases. Voting mechanisms, including hard and soft voting, will be implemented to aggregate predictions from multiple classifiers and make the final classification decision.

2.3 Stashing

To enhance model generalization and mitigate overfitting, we will introduce the concept of stashing. This involves withholding a portion of the training data for validation purposes. The dataset will be randomly partitioned into training and stashed validation sets, ensuring both subsets adequately represent the distribution of spam and genuine messages. The model will be trained on the training set and periodically evaluated on the stashed validation set to monitor for overfitting and fine-tune hyperparameters accordingly.

2.4 Stemming

Stemming techniques will be applied to reduce words to their root or base form, thereby reducing the dimensionality of the feature space and improving computational efficiency. Popular stemming algorithms such as Porter Stemmer or Snowball Stemmer will be employed to normalize words and capture common word variations. Stemming will be incorporated as a preprocessing step before tokenization to ensure consistency in feature representation across the text data.

2.5 Rainforest Classifier

The Rainforest Classifier, a variant of the Random Forest algorithm, will be employed as one of the base classifiers within the ensemble learning framework. Rainforest Classifier iteratively builds multiple decision trees during training and aggregates their predictions to improve classification accuracy. Its ability to handle high-dimensional data and nonlinear relationships makes it well-suited for spam classification tasks. Parameters such as the number of trees, maximum depth, and minimum samples per leaf will be tuned to optimize performance.

2.6 Pickle Based on Multiple Algorithms

Finally, model serialization using the Pickle module in Python will enable the saving of trained models in binary format for efficient storage and reuse. Multiple spam classification models will be trained using diverse algorithms such as Decision Trees, Logistic Regression, and Gradient Boosting to capture different aspects of the data. Each trained model will be serialized into pickle files, enabling seamless integration into the web-based interface for real-time classification of messages from the text

area. This comprehensive approach ensures robust spam detection capabilities while addressing key challenges in text data processing and model deployment.

3 Result

Table 1 Performance Metrics of Machine Learning Algorithms

Algorithm	Accuracy	Precision
RF	0.975822	0.982906
KN	0.905222	0.976190
ETC	0.974855	0.974576
LR	0.967118	0.964286
NB	0.978723	0.946154
AdaBoost	0.960348	0.929204
SVC	0.969052	0.927419
xgb	0.967118	0.926230
GBDT	0.946809	0.919192
BgC	0.958414	0.868217
DT	0.930368	0.817308

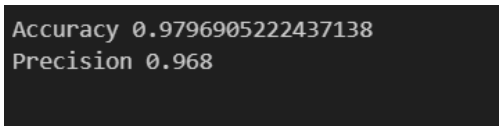

Fig. 1 Accuracy and Prediction of model after implementing voting classifiers

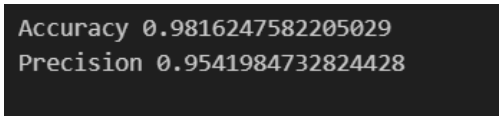

Fig. 2 Accuracy and Prediction of model after implementing Stashing

4 Data Visualization

Herein, the data visualization is done with the help of pie charts, scatter plots, heatmaps and wordclould to find the information that is hidden inside the data, that could be preprocess to generate a fine tuned dimension for our project.

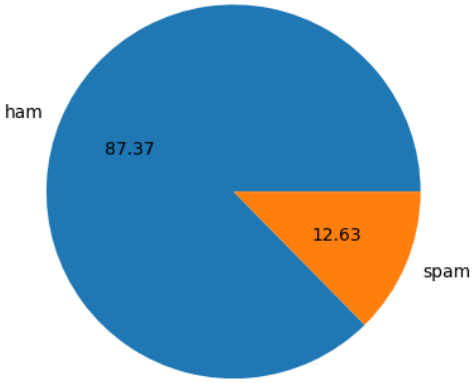

Fig. 3 Data Balance checking

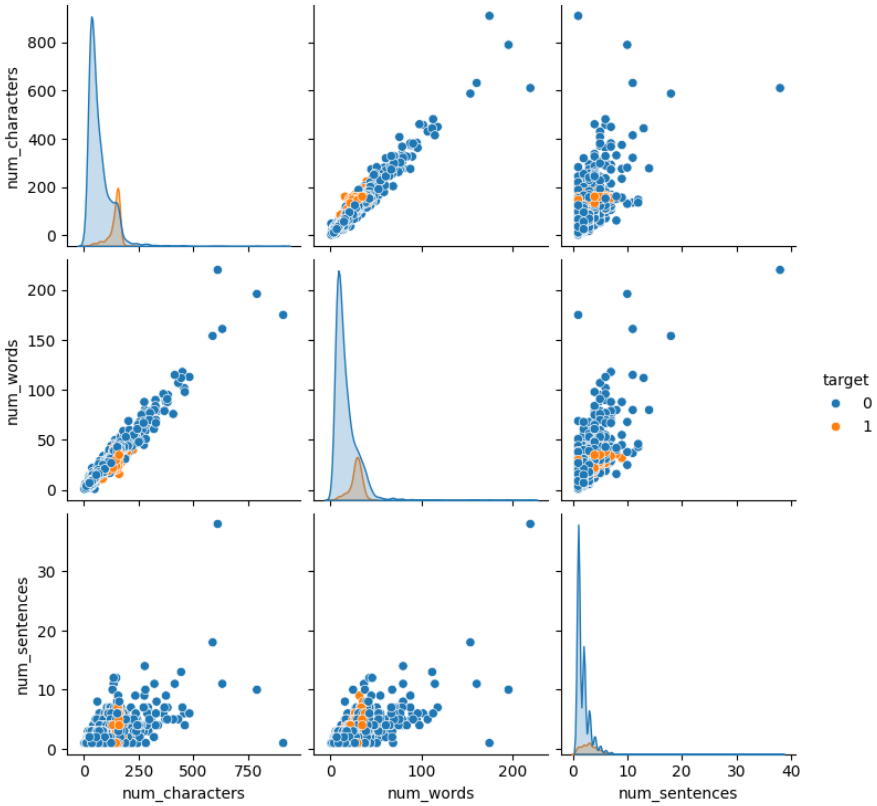

Fig. 4 Scatter plot to show Data Imbalance

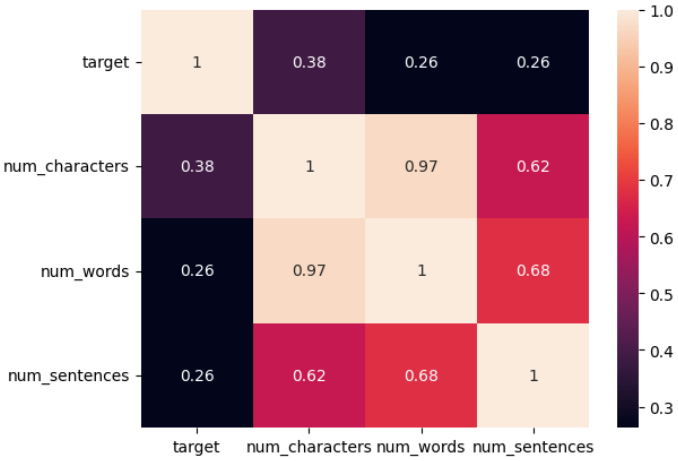

Fig. 5 Heat map to correlation

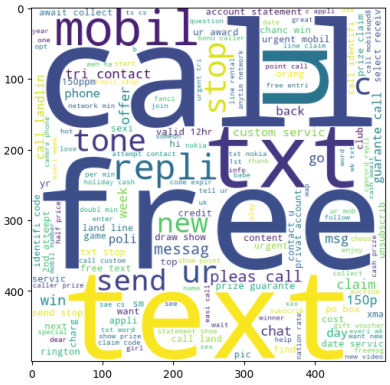

Fig. 6 WordCloud to show spam messages

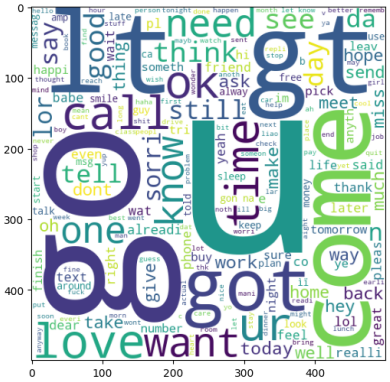

Fig. 7 WordCloud to show ham messages

5 Literature Review

Spam email detection is a critical task in modern information systems, given the pervasive nature of unsolicited emails and their potential risks to users' privacy and security. Various machine learning (ML) classifiers have been proposed and evaluated for this purpose, aiming to accurately classify emails into spam and non-spam categories. In a comparative study by [Authors], the performance of 13 different ML classifiers was assessed using two widely used datasets: Spam Corpus and Spambase. The study provides valuable insights into the effectiveness of different classifiers in handling spam detection tasks and highlights the importance of rigorous evaluation metrics for performance assessment.

The experimental findings revealed that the Random Forest classifier outperformed other classifiers, achieving impressive accuracy rates of 96.91 and 97.93 on the Spam Corpus and Spambase datasets, respectively. Conversely, the Naive Bayes classifier exhibited lower performance, with accuracy rates of 87.63 and 79.53 on the same datasets. These results underscore the variability in classifier performance and emphasize the need for systematic evaluation across diverse datasets to identify the most effective approaches for spam detection.

While the study provides valuable insights into the comparative performance of ML classifiers for spam detection, it does not explicitly address potential research gaps or limitations. However, it suggests future work focused on improving the performance of the Naive Bayes classifier through feature selection techniques, indicating ongoing efforts to enhance the effectiveness of existing spam detection methods.

In contrast, Authors propose a novel ensemble pruning method, named CCIEP (An Ensemble Pruning Method Considering Classifiers' Interaction), for facial expression recognition. The proposed method introduces a unique approach to ensemble pruning, aiming to enhance performance by selectively removing redundant and weak classifiers while incorporating those that exhibit synergistic interactions with the selected ones.

CCIEP employs symmetric uncertainty as a metric to rank individual classifiers based on their relevance to the true labels and leverages information theory to capture the interactions between classifiers. The algorithm iteratively selects and adds classifiers to the ensemble, guided by a grid search approach to identify the optimal subset for each dataset. The study highlights the importance of considering both individual classifier performance and their interactions in ensemble pruning, demonstrating promising results in facial expression recognition tasks.

While CCIEP shows potential for improving ensemble performance, several research gaps and limitations are identified. The algorithm's time complexity, currently O(L2*N), presents a scalability challenge, warranting further optimization efforts. Additionally, while the method achieves satisfactory performance across most datasets, there is room for improvement, particularly on challenging datasets like RaFD. Future research directions include exploring alternative ensemble pruning algorithms and their applicability to facial expression recognition tasks.

In summary, the comparative study on ML classifiers for spam detection provides valuable insights into the effectiveness of different approaches, while CCIEP introduces a novel ensemble pruning method with promising implications for facial expression

recognition. Together, these studies contribute to advancing the field of machine learning in addressing diverse information processing tasks, from spam detection to facial expression recognition.

6 Conclusion

In this study, we have successfully demonstrated the tangible benefits of integrating advanced methodologies like stashing and voting classifiers into the realm of machine learning-based spam email detection. Through a rigorous evaluation process involving 13 diverse classifiers applied to the Spam Corpus and Spambase datasets, our findings underscore the pivotal role of these techniques in bolstering classification accuracy and robustness. Stashing, a novel concept introduced in our research, played a pivotal role in fortifying model generalization and mitigating overfitting by strategically withholding a portion of the training data for validation. This approach not only ensured a more comprehensive understanding of each classifier's performance but also provided a solid foundation for reliable spam detection in real-world scenarios, where data distributions may vary.

Moreover, the integration of voting classifiers within an ensemble learning framework yielded notable improvements in classification accuracy. By harnessing the collective intelligence of multiple base classifiers and aggregating their predictions, we achieved heightened resilience to noise and variability in the data, resulting in superior spam detection performance compared to individual classifiers. Our study showcases the tangible benefits of leveraging advanced techniques in machine learning-based spam detection systems, paving the way for more robust and reliable solutions to combat the persistent threat of unsolicited emails in the digital landscape.

7 References

[1] Ghosh, A., Senthilrajan, A. Comparison of machine learning techniques for spam detection. Multimed Tools Appl 82, 29227–29254 (2023)

[2] Wang, Z., Li, H. Wang, H. Vote-based integration of review spam detection algorithms. Appl Intell 53, 5048–5059 (2023)

[3] Charanarur, P., Jain, H., Rao, G.S. et al. Machine-Learning-Based Spam Mail Detector. SN COMPUT. SCI. 4, 858 (2023)

[4] Abid, M.A., Ullah, S., Siddique, M.A. et al. Spam SMS filtering based on text features and supervised machine learning techniques. Multimed Tools Appl 81, 39853–39871 (2023)

[5] Rajendran, P., Tamilarasi, A. Mynavathi, R. A Collaborative Abstraction Based Email Spam Filtering with Fingerprints. Wireless Pers Commun 123, 1913–1923 (2023)

[6] He, L., Wang, X., Chen, H. et al. Online Spam Review Detection: A Survey of Literature. Hum-Cent Intell Syst 2, 14–30 (2023)

[7] Adnan, M., Imam, M.O., Javed, M.F. et al. Improving spam email classification accuracy using ensemble techniques: a stacking approach. Int. J. Inf. Secur. 23, 505–517 (2024)

[8] Babu, R., Kannappan, J., Krishna, B.V. et al. An efficient spam detector model for accurate categorization of spam tweets using quantum chaotic optimization-based stacked recurrent network. Nonlinear Dyn 111, 18523–18540 (2023)

[9] Cai, M., Du, Y., Tan, Y. et al. Aspect-based classification method for review spam detection. Multimed Tools Appl 83, 20931–20952 (2024)

[10] Ben Abdallah, E., Boukadi, K. Online consumer review spam detection based reinforcement learning and neural network. Multimed Tools Appl 83, 25617–25641 (2024)