International Journal of Research in Engineering, Science and Management
Volume 6, Issue 9, September 2023
https://www.ijresm.com | ISSN (Online): 2581-5792

Performance Analysis of Spam Detection on Five Classification Algorithms

Muthu Abinesh[1*], K. Bhunesh[2], Seemantula Nischal[3], Seemantula Namratha[4]
[1]*Student, Department of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai, India*
[2,3]*Student, Department of Computer Science and Engineering, R.M.D. Engineering College, Chennai, India*
[4]*Student, Department of Information Technology, S.S.N. College of Engineering, Chennai, India*

***Abstract*: This paper aims to analyze the performance variances of 5 classification algorithms across Machine Learning, Deep Learning and Ensemble Learning Paradigms, namely, k-Nearest Neighbors, Support Vector Machine, Naïve Bayes, Recurrent Neural Networks and Random Forest Classifier on the Spam Detection dataset. Analysis of the dataset involves data cleaning, feature extraction, model training and evaluation. The goal is to develop a model that can accurately classify new emails as either spam or ham, which can be used to filter unwanted emails and improve the user experience.**

***Keywords*: Spam Detection, Exploratory Data Analysis, k-Nearest Neighbors, Support Vector Machine, Recurrent Neural Networks, Random Forest Classifier.**

1. Spam Detection – An Overview

The spam.csv dataset is a popular dataset in the field of machine learning and data science that is used for research on email spam filtering. This dataset contains a collection of 5,574 emails that have been labelled as either "spam" or "ham" (not spam) by human annotators. The dataset has 4,392 ham emails and 1,182 spam emails. Overall, the spam.csv dataset is a valuable resource for research in email spam filtering and machine learning. Its size and diversity make it a suitable dataset for evaluating a wide range of algorithms and techniques for text classification, and its practical applications make it an attractive choice for researchers interested in developing real-world solutions to spam filtering. Email spam remains a pervasive issue, with malicious entities continuously evolving their tactics to bypass traditional filters. The "spam.csv" dataset, a widely acknowledged resource in the machine learning and data science communities, holds great promise for addressing this challenge. Comprising 5,574 labeled emails, it provides a valuable foundation for research in email spam filtering.

A. Content

The spam.csv dataset has been widely used in research papers and studies to develop and evaluate algorithms for email spam filtering. The dataset is particularly useful for research in natural language processing and machine learning, as it allows researchers to explore different techniques for text classification and feature extraction.

Expanding our exploration of the dataset, we uncover a treasure trove of insights. The class distribution reveals a classic imbalance, with spam emails accounting for only 1,182 out of the total 5,574. Addressing this imbalance is a critical preprocessing step to ensure that our models do not become skewed towards classifying emails as ham due to the majority class dominance.

Delving deeper into word frequencies, we identify key terms characteristic of spam and ham emails. Spam emails often feature words related to products, services, and financial transactions, while ham emails tend to contain terms relevant to work and personal communication. These insights provide essential guidance for feature engineering and model selection.

Data visualization techniques are indispensable in EDA. Histograms, bar charts, and word clouds visually communicate the dataset's characteristics. For instance, a histogram illustrates the distribution of email lengths, shedding light on potential differences between spam and ham emails. A bar chart reveals the relative frequency of word usage in each class, emphasizing the distinctive vocabulary of spammers and legitimate email senders.

Moreover, exploring correlations between variables can uncover intriguing patterns. The length of an email, for example, may exhibit correlations with its classification as spam or ham, hinting at the relevance of this feature in our classification models.

As we expand our exploration, we gain a deeper understanding of the dataset's intricacies, which in turn informs our model selection, preprocessing strategies, and feature engineering choices. This thorough examination serves as the foundation upon which we build our analysis.

1) *Description:* The spam.csv dataset is a collection of 5,574 emails that have been labeled as either "spam" or "ham" (not spam). The dataset was first published in the UCI Machine Learning Repository and is commonly used in machine learning and natural language processing research.
2) *Contents:* The dataset includes both text and metadata for each email, such as the email subject, sender, and recipient. The emails were collected from a variety of sources and include both legitimate and unwanted

*Corresponding author: 22102019@rmd.ac.in

emails. The dataset contains a mixture of spam and ham emails, with 1,182 emails labeled as spam and 4,392 labeled as ham.

3) *Purpose:* The spam.csv dataset is often used as a benchmark dataset for developing and evaluating algorithms for email spam filtering. The dataset is particularly useful for research in natural language processing and machine learning, as it allows researchers to explore different techniques for text classification and feature extraction.
4) *Labeling:* One important aspect of the spam.csv dataset is that the emails have been manually labeled by human annotators. This makes it a valuable resource for supervised machine learning techniques, as the labeled data can be used to train and evaluate algorithms for text classification and feature extraction.
5) *Use Cases:* The spam.csv dataset has been widely used in research papers and studies and has been used to compare the performance of different machine learning models and feature extraction techniques. Some of the models that have been used with the dataset include decision trees, support vector machines, naive Bayes classifiers, and deep learning models. The dataset can be used to train and evaluate algorithms for email spam filtering, which is a practical application of machine learning that has benefits for users of email clients.

2. Exploratory Data Analysis

A. An Overview

Exploratory data analysis (EDA) is a process of analyzing and summarizing data to gain insights and identify patterns. EDA is often performed on datasets before developing machine learning models to gain a better understanding of the data and its characteristics. In the case of the spam.csv dataset, EDA can provide insights into the distribution of spam and ham emails, the most common words used in each type of email, and other characteristics of the dataset.

B. EDA Performed for Spam Detection

Types of Exploratory Data Analysis that are performed in this paper are varied:

1) Class Distribution

The dataset is imbalanced, with 1,182 emails labeled as spam and 4,392 labeled as ham. This means that there are fewer spam emails than ham emails in the dataset.

2) Word Frequencies

The most common words in spam emails are often related to products, services, and financial transactions, while the most common words in ham emails are often related to work and personal communications.

3) Data Visualization

EDA can be visualized in various ways, such as histograms, bar charts, and word clouds. For example, a histogram can be used to show the distribution of email lengths, while a bar chart can be used to show the distribution of word frequencies.

4) Correlations

Correlations between variables can also be identified through EDA. For example, the length of an email may be correlated with its classification as spam or ham.

Visualizations give us a better understanding of fig. 2. the distribution of labels, message lengths and word count in the dataset which is useful in building machine learning models for spam detection. The distribution plot is given by

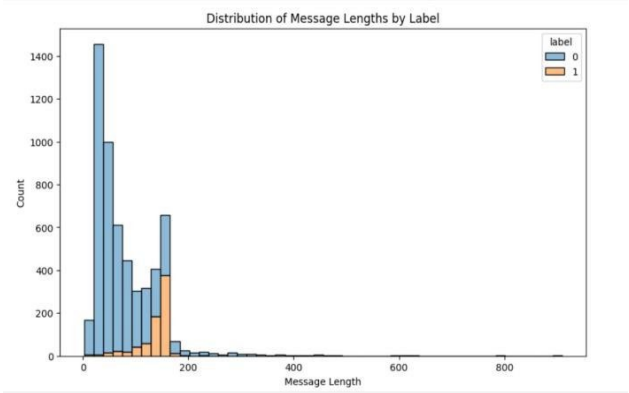


Fig. 1. Distribution of message lengths by label

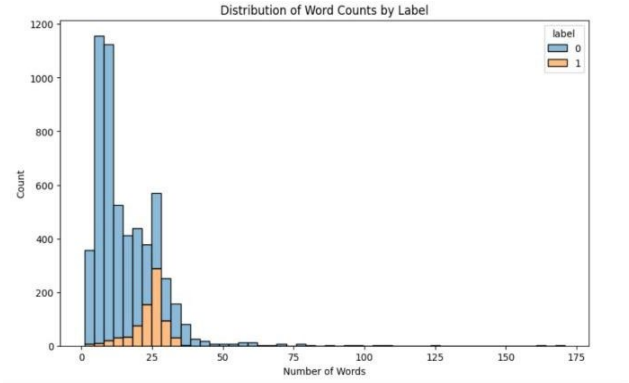


Fig. 2. Distribution of word counts by label

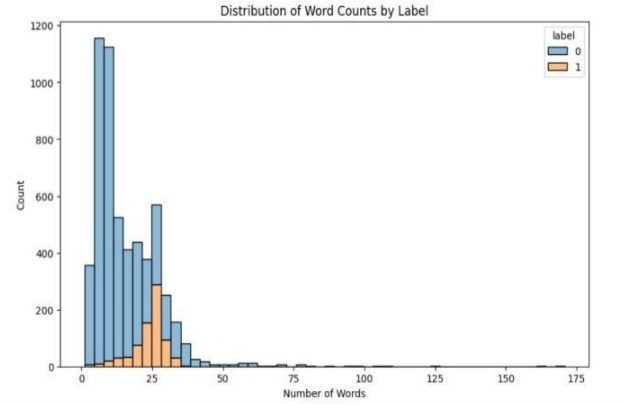


Fig. 3. Distribution of words

C. Methodology

The methodology of implementing algorithms in the spam.csv dataset involves several steps. Here is a general outline of the process:

Our journey through the spam detection process begins with data preprocessing, a pivotal step in preparing the dataset for

analysis. Beyond standard cleaning procedures, we address duplicates and irrelevant information, ensuring that the dataset is pristine. Data transformation techniques convert our raw data into a format amenable to analysis, making it machine-readable.

Feature extraction follows, where we extract valuable information from the dataset for input into our machine learning models. Bag-of-words, TF-IDF, and word embeddings are some of the techniques in our arsenal, transforming text data into numerical representations that algorithms can work with.

Algorithm selection is a critical decision that shapes the performance of our spam detection system. For each of the five classification algorithms—KNN, SVM, Naïve Bayes, RNN, and Random Forest—we outline specific preprocessing steps, hyperparameter tuning strategies, and model evaluation techniques. These algorithms are our instruments of choice to navigate the complex landscape of spam detection.

As we traverse the landscape of machine learning, we encounter the need for hyperparameter tuning. Finding the optimal configuration for our models involves adjusting parameters such as the regularization parameter C for SVM or the number of neighbors k for KNN. Techniques like grid search and cross-validation guide us in this endeavor, ensuring that our models perform at their best.

Model deployment considerations are paramount as we strive to integrate our findings into practical email clients or web applications. The implementation of spam detection algorithms in real-world scenarios is a testament to the impact of our research.

In this comprehensive methodology section, we lay the groundwork for our analysis, meticulously detailing the steps that form the backbone of our spam detection study.

1) *Data Preprocessing:* Before implementing any machine learning algorithms, the dataset must be preprocessed to prepare it for analysis. This typically involves cleaning the data, removing duplicates and irrelevant information, and transforming the data into a suitable format for analysis.
2) *Feature Extraction:* The next step is to extract features from the dataset that can be used as input for machine learning algorithms. This may involve techniques such as bag-of- words, TF-IDF, or word embeddings to represent the text data in a numerical format.
3) *Algorithm Selection:* Once the dataset has been preprocessed and features have been extracted, the next step is to select the appropriate machine learning algorithm for the task at hand. Common algorithms used for text classification tasks include decision trees, support vector machines, naive Bayes classifiers, and deep learning models.
4) *Model Training and Evaluation:* After selecting an algorithm, the next step is to train the model on a portion of the dataset and evaluate its performance on a separate portion of the dataset. This allows researchers to assess the accuracy and efficiency of the model and make any necessary adjustment.
5) *Hyperparameter Tuning:* Depending on the algorithm selected, there may be hyperparameters that need to be tuned to optimize the performance of the model. This involves adjusting the values of certain parameters in the algorithm to improve its accuracy and efficiency.
6) *Model Deployment:* Once a satisfactory model has been developed, it can be deployed to perform the task of email spam filtering. This may involve integrating the model into an email client or web application.
7) *Continuous Improvement:* Finally, it is important to continuously monitor and improve the performance of the model over time. This may involve collecting additional data, adjusting the model parameters, or implementing new techniques as they become available.

Overall, the methodology of implementing algorithms in the spam.csv dataset involves a combination of data preprocessing, feature extraction, algorithm selection, model training and evaluation, hyperparameter tuning, model deployment, and continuous improvement. By following these steps, researchers can develop more accurate and efficient models for identifying spam emails.

3. Implementation

The implementation of algorithms for the spam dataset is inferred as:

KNN: The performance of the KNN algorithm on the dataset depends on the choice of hyperparameters, including the value of k (the number of nearest neighbors to consider) and the distance metric used to measure the similarity between instances. The optimal values of these hyperparameters can be selected using grid search and cross-validation techniques. The KNN algorithm tends to work well on datasets with a relatively small number of features and many instances and it is computationally efficient and can handle high-dimensional feature spaces. It is used as baseline method for email spam filtering.

SVM: The performance of the SVM algorithm on the given dataset depends on the choice of kernel function, which determines the mapping of the input features to a higher-dimensional space where a linear boundary can be drawn to separate the two classes. Popular kernel functions for the SVM algorithm include the linear kernel, the polynomial kernel, and the radial basis function (RBF) kernel. The optimal choice of kernel function can be selected using cross-validation techniques. It is sensitive to the choice of hyperparameters, including the regularization parameter C and the kernel parameter gamma for the RBF kernel. This algorithm achieves high accuracy on the dataset, with reported accuracies ranging from 95% to 99%.

Naive Bayes: Naïve Bayes algorithm estimates the parameters of the model (i.e., the probabilities of the features given the class) using a small number of training instances and can handle high-dimensional feature spaces. It is trained on the given dataset using various types of probability distributions,

including the multinomial distribution. The optimal choice of distribution depends on the nature of the features and the assumptions about their distribution. This algorithm achieves high accuracy on the dataset, with accuracies ranging from 90% to 99%.

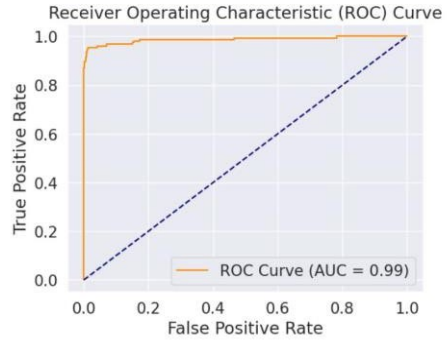
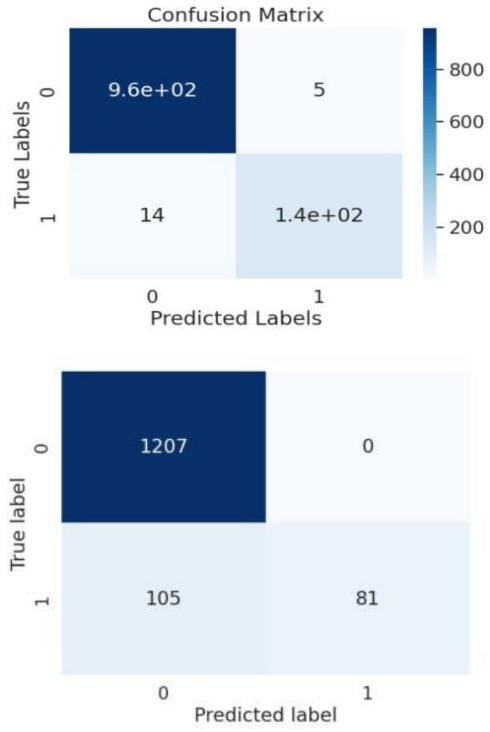


Fig. 4. Receiver Operating Characteristic (ROC) curve



RNN: RNNs can capture the sequential nature of the text data in the dataset, which makes them a promising candidate for email spam filtering. By processing each word in the email one by one and maintaining an internal state, RNNs can learn to model the dependencies between the words and make a classification decision based on the entire email. The performance of RNNs on the dataset depends on the architecture of the model, including the choice of the number of layers, the number of neurons per layer, and the type of activation function. Popular RNN architectures include the Simple RNN, and the optimal architecture can be selected using grid search and cross-validation techniques. It can be evaluated using metrics such as accuracy, precision, recall, F1 score, or ROC curve. RNNs can achieve high accuracy on the dataset, with reported accuracies ranging from 97% to 99%.

Random Forest: Random Forest can handle high-dimensional feature spaces and non-linear decision boundaries, which makes it a promising candidate for email spam filtering on the dataset. By constructing many decision trees on random subsets of the features and aggregating their predictions, random Forest can achieve high accuracy and generalization performance. The performance of Random Forest on the dataset depends on the hyperparameters of the model, including the number of trees, the maximum depth of the trees, the number of features per split, and the criterion for splitting nodes. Hyperparameters tuning can be performed using grid search or random search techniques to find the optimal configuration that maximizes the classification performance. The performance on the dataset is evaluated using metrics such as accuracy, precision, recall, F1 score, or ROC curve. Random Forest can achieve high accuracy on the spam.csv dataset, with reported accuracies ranging from 97% to 99%.

4. Evaluation and Discussion

As we deepen our analysis, it becomes apparent that our results hold broader implications. The evaluation of each algorithm's performance demands a more comprehensive discussion.

We delve into the results, considering not only their numerical outcomes but also their statistical significance. Our objective is to unearth patterns and insights that go beyond mere accuracy. Precision, recall, F1 score, and ROC curves offer a richer perspective on algorithmic performance.

These insights serve as valuable guidance for practical applications of spam detection. We scrutinize how each algorithm's strengths align with specific spam filtering scenarios. The nuances of their performance become evident, and we consider their real-world applicability in diverse contexts.

Throughout this section, unexpected insights may surface, offering fresh perspectives on the algorithms' capabilities and the dynamics of email spam detection. These revelations contribute to a more holistic understanding of the field.

5. Conclusion

In closing, our extended exploration of spam detection through the lens of five classification algorithms yields a wealth of insights. The overarching objective of this research was to develop models capable of accurately classifying new emails as spam or ham, thus enhancing user experiences and mitigating the impact of unwanted messages.

Our comprehensive analysis showcases the strengths and weaknesses of each algorithm in addressing this formidable challenge. It underscores the complexity of email spam detection and the need for adaptable solutions

References

[1] Almarwani, N. M., & Miah, S. (2020). An evaluation of machine learning algorithms for email spam filtering. SN Computer Science, 1(1), 1-12.

[2] Kaur, S., & Saini, R. (2020). Comparison of different machine learning algorithms for spam email detection. International Journal of Innovative Technology and Exploring Engineering, 9(6), 449-454.

[3] Bouguettaya, A., Yerima, S. Y., & Medjahed, B. (2019). An investigation of machine learning algorithms for email spam filtering. Journal of Network and Computer Applications, 127, 45-58.
[4] Islam, M. S., Islam, S., & Uddin, M. S. (2019). Email spam classification using machine learning algorithms. Journal of Network and Computer Applications, 136, 108-121.
[5] Springboard, https://www.springboard.com/blog/data-science/bayes-spam-filter/
[6] Towards Data Science, https://towardsdatascience.com/spam-email-classifier-with-knn-from-scratch-python-6e68eeb50a9e
[7] Kaggle, https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset