Performance Analysis of Spam Detection on Five Classification Algorithms

Ragul T, Seemantula Namratha,
Selvi Parasakthi K, Shanmukha Naveen K,
Sharon Roshini S

Department of Information Technology
SSN College of Engineering
Kalavakkam, Chennai, India

{ragul2010444, seemantula2010636,
parasakthi2010445, shanmukhanaveen2010809 &
sharonroshini2010942}@ssn.edu.in

***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.**

Index Terms – Spam Detection, Exploratory Data Analysis, k-Nearest Neighbors, Support Vector Machine, Recurrent Neural Networks, Random Forest Classifier

I. 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.

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.

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 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.

B. Preparing Your PDF Paper for IEEE Xplore©

TensorFlow Datasets are a collection of datasets that are easy to use, and Spam Detection is one of them. In some special cases, text data is required to be converted to Comma Separated Value (CSV) data.

II. 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.

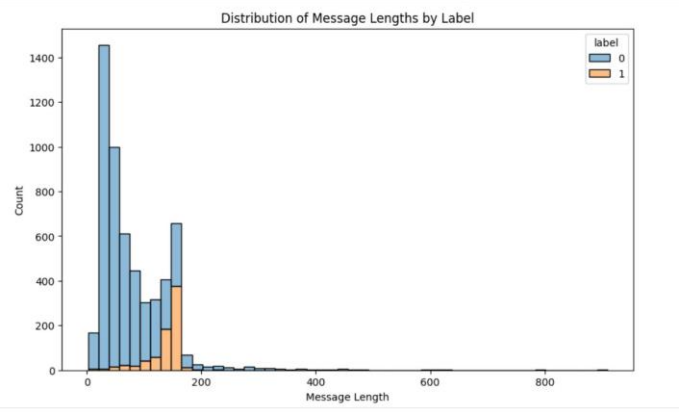
3) Data Visualization:

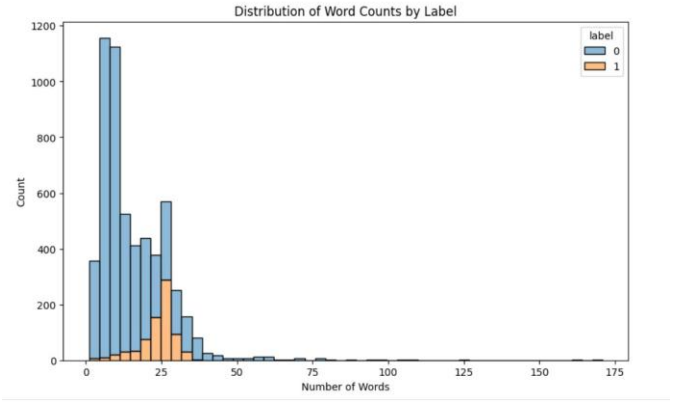
EDA can be visualized in various ways, such as histograms, bar charts, and word clouds. For example, a histogram can be used to show the distribution of email lengths, while a bar chart can be used to show the distribution of word frequencies.

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 the distribution of labels, message lengths and word counts in the dataset which is useful in building machine learning models for spam detection. The distribution plot is given by





C. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even if they have been defined in the abstract. Abbreviations such as IEEE, SVM, RNN, KNN and RBF do not have to be defined. Do not use abbreviations in the title unless they are unavoidable.

D. Methodology

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

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 adjustments.

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.

E. Other Recommendations

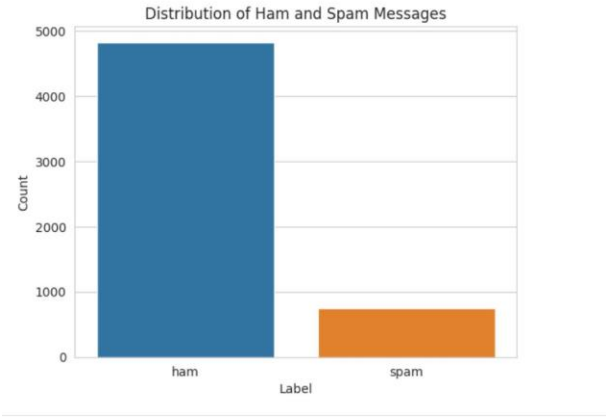
The Roman numerals used to number the section headings are optional. Do not number ACKNOWLEDGEMENT and REFERENCES and begin Subheadings with letters. Use one space after full stops. Hyphenate complex modifiers. Avoid dangling participles. Use a zero before decimal points: “0.25,” not “.25.” Do not add page numbers.

III. IMPLEMENTATION

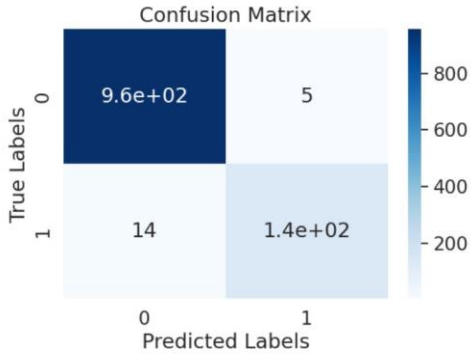
The implementation of algorithms for the spam dataset is inferred as:

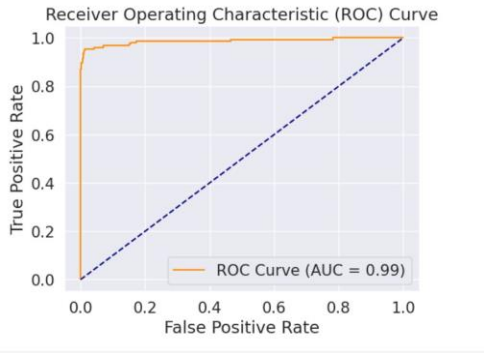
1) *KNN:* The performance of the KNN algorithm on the dataset depends on the choice of hyperparameters, including the value of k (the number of nearest neighbors to consider) and the distance metric used to measure the similarity between instances. The optimal values of these hyperparameters can be selected 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.

2) 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%.



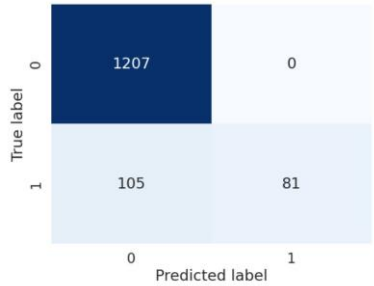
3)Naive Bayes: Naïve Bayes algorithm estimates the parameters of the model (i.e., the probabilities of the features given the class) using a small number of training instances and can handle high-dimensional feature spaces. It is trained on the given dataset using various types of probability distributions,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





4) 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%.

5) 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%.



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. https://doi.org/10.1007/s42979-019-0002-2

[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. https://doi.org/10.35940/ijitee.K9392.099620

[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. https://doi.org/10.1016/j.jnca.2018.12.002

[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. https://doi.org/10.1016/j.jnca.2019.03.010

[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