Email Spam Detection Using Machine Learning Approach

1st Fatima .
P2833125
MSc. Artificial Intelligence
De Montfort University
p2833125@my365.dmu.ac.uk

***Abstract*—Email spam is a irrelevant email that is sent to unwanted recipient and fills the inboxes worldwide. This results waste of time for users and business holders and computational resources. This paper handles the challenges in identifying the spam emails among the emails available. To do this, this paper uses machine learning approach. Three algorithms have been used to identify the spam emails in this report such as, Naive Bayes, Support Vector Machine (SVM), and Logistic Regression. The main purpose of this report is to implement the models using these three above mentioned algorithms, and then, distinguishes the performance of the algorithms and assess their efficiency to recognize spam emails correctly. To carry out study, real world data set has been used which holds spam and non-spam emails. This data has been extracted and trained to test the algorithms. By taking it through various steps of experiment and analysis, we monitor the performance of all algorithms by using evaluation metrics. Furthermore, this report provides the improvements and suggestions for upcoming research in this digital world. In Summary, this paper collaborates the continues efforts in conflicting email spams vs hams by detail evaluation of three famous machine learning approaches. Through the experiment and analysis of different models, this will help to build more efficient spam filtering system and will also improve the user experience through out the world in this digital world.**

***Index Terms*—Email spam, Machine learning, Naive Bayes, Support Vector Machine (SVM), Logistic Regression, Classification, Evaluation metrics, Performance analysis,**

I. Introduction

Communication via email has become very crucial part of the life in this era. But, it has its advantages and disadvantages too. One of the drawback found is spam emails. Unwanted messages such as marketing emails from various websites or malicious content appears to be flooded in our inboxes makes things hard for users to handle it. It can also breach security risks to business holders or organisations. To help organisation and individuals to solve this problem, developers and researchers have built algorithms in machine learning to help system identify emails as spam or ham. The machine learning algorithms associated with the Artificial Intelligence enabled several techniques which automate this identification and thereby securing the environment by handling these situations.

This paper uses three well-known algorithms from machine learning to identify spam detection. The three algorithms are as follows: Naive Bayes, Support Vector Machines, and Logistic Regression. To test how these algorithms can be used to identify spam detection individually, this research has built models based on each and predicted with the help of real world dataset which contain spam and ham emails. Also, comparison of the model has been performed with evaluation matrix especially, accuracy, precision, recall and F1-score.

The main purpose in this report is to highlight the improvements that machine learning can bring in spam detection by using one over another. An improved model can implement a system that will bring more safety and will make email platform safer for individuals and organisation. Challenges when deploying these solutions will be discussed in this report such as identifying new spam types and more test data to further test the algorithms. This report also provides the basics of building more efficient ways of implementing algorithm to identify spam emails machine learning techniques. Thereby, this paper aims to contribute the research community and academia space to extend the research as well as an improved model for the real world implementations done by the email providers which identifies and isolate the irrelevant emails from the inbox automatically.

II. Background Work

A. Introduction to Machine Learning (ML)

Machine Learning (ML) is defined as a set of techniques and approaches that enables the machines to learn from data and then make predictions or decisions without human intervention. This is one of the main area comes under Artificial Intelligence.

ML algorithms identify patterns within data, allowing systems to improve their performance on tasks over time. This capability makes ML particularly useful for tasks like email spam detection, where the nature of spam continually evolves. In the context of email spam detection, ML algorithms can analyze features such as the content of the email, metadata, and sender information to classify emails which are irrelevant(spam) or legitimate(ham).

The following provides brief information about the algorithms that has been used for model implementation in this research, a literature review of research which have been done in this area of email spam detection, and tools and concepts used in this research.

B. ML Algorithms Used in This Paper

This paper focused on three widely known machine learning algorithms which are as given below.

1) Logistic Regression: Logistic regression is a classification problem [1]. It relays on basic statistical functions in mathematics. It does the feature extraction by finding relationship between independent variable from the dataset. This accomplish the feature extraction and the connection with the outcome too. It does assign a probability for each observation it makes, and this probability would be using to predict the belonging of that observation into a specific class. Unlike linear regression, logical regression follows continuous prediction. This is achieved by expressing the relationship between features and the expected outcomes using the logistic function.

2) Support Vector Machines(SVM): Support Vector Machines, introduced by Vapnik et al. (1995) [2], are a powerful supervised machine learning approach known for their effectiveness in high-dimensional spaces. SVMs does the classification by finding a suitable hyperplane which separates different data classes. This margin is defined by the closest data points to the hyperplane; this data points is called as support vectors. SVMs offer advantages in terms of memory efficiency as the decision function relies on these support vectors, a subset of the training data. Additionally, SVMs can be made to work with non-linear data through the use of kernel functions. This makes them best among other classification algorithms in machine learning

3) Naive Bayes Classifier: Naive Bayes classifiers comes under supervised learning algorithms for classification tasks [3]. This algorithm uses Bayes' theorem to find the probability associated with a data point to predict which class it belongs to. These algorithms often achieve good performance in various domains due to their simplicity and efficiency [4]. However, this very assumption can also lead to limitations in more complex scenarios where feature dependencies are significant.

C. Related Research

Machine learning has created a large impact in automating modern spam filtering due to its ability to learn and adapt to the spam tactics. This section presents the overview of significant research works carried out across the world in the field of exploring various machine learning approaches for email spam classification. Due to their simplicity and efficiency, Naive Bayes algorithms are widely used for spam detection. Research by Tabish [5] highlights the effectiveness of Naive Bayes in classifying spam emails based on features like sender information, subject line content, and presence of URLs. This work demonstrates the importance of feature selection and data pre-processing for optimal performance. Several studies happened with the use of SVM too. Studies by Manevitz [6] explore using SVMs for spam classification. This study shows high accuracy even with limited training data. Logistic regression has been a tool for several researches. Early works by Drucker [7] demonstrates its ability to learn from email features and classify emails as spam or legitimate. Since then, research by Sakaki [8] further investigated its effectiveness, highlighting its suitability for real-time spam filtering due to its computational efficiency.

The literature review found several other research work that shows the machine learning approach to separate the email into legitimate or not. To conclude, after the deep literature review, it is evident that machine learning offers a dynamic and adaptable approach to email spam classification. As spammers develop new tactics, this field needs further exploration, and this time is the right time to check how different algorithms to find themselves in solving email spam classification problem. This research is considered as a preliminary where further extension can be made by exploring the transfer learning, deep learning and other new innovations to solve the same problem with improved efficiency.

D. Tools used for this Research

For conducting this research, several tools and libraries were utilized to facilitate data prepossessing, model implementation, and analysis.

This research uses TensorFlow [9]. It is a powerful machine learning framework developed by Google. It helps developers and students to build and train machine learning models. TensorFlow provides a high-level interface that enables easy construction of neural networks and other machine learning algorithms. Which makes it suitable for tasks such as email spam classification. Additionally, scikit-learn [10], another popular machine learning library in Python, was utilized for various tasks including data preprocessing, feature extraction, and evaluation of machine learning models. Scikit-learn offers a wide range of algorithms and tools for classification, regression, clustering, and dimensional reduction, making it invaluable for conducting experiments and analyzing results.

As a workbench Visual Studio Code (VSCode) [11] has been preferred to create a platform where the above mentioned libraries can be used for modeling the solution. It is code editor developed by Microsoft, served as the primary integrated development environment (IDE) for writing, debugging, and running code throughout the research process.

Overall, the combination of TensorFlow, scikit-learn, and VSCode provided a robust and versatile tool-set for making this research successful.

III. Implementation of ML Models

This section comprised of two subsections. First subsection details the step by step procedure used for the implementation of the model. Second subsection provides how the three selected machine learning algorithms has been used for the implementation of three separate models.

A. Steps to Model Implementation

The implementation has been divided into four steps, such as data splitting, feature extraction, building and training the model, and making predictions. Note that the data has been preprocessed.

1) Data Splitting: The implementation begins with Data splitting. In which the data is split into two, one for training the model, and other for testing the model. This step is to make ensure that the model is evaluated for unseen data, which is the test data. This test data has been used for testing the accuracy of the model. The sklearn library contains a function named 'train_test_split',which is used to achieve it. This research uses 80% of the data allocated for training, and rest of the data allocated for testing. This ensures the model has got enough data for training of the model.

In this step itself, from the two columns in the dataset, such as "Message", and "Category", the "Category" columns is converted from textual labels ('spam' and 'ham') to numerical labels (0 for spam and 1 for ham) to facilitate model training.

2) Feature Extraction: This is the second step, in which the textual data is transformed into numerical features. This has been done to reduce the data processing effort of the model. In this implementation, a vectorizer named TF-IDF (Term Frequency-Inverse Document Frequency) [12] is used to reconstruct the email messages into a matrix of TF-IDF features. This technique helps in emphasizing the important words while reducing the impact of less important ones. The TF-IDF vectorizer is fitted on the data to be trained to learn the vocabulary and then transformed to both training and testing data to create the feature matrices. This step is essential to convert the data which is row into the data which can be understandable by the model created and do prediction.

3) Building and Training the Model: In this steps, building and training of the model is performed. The data prepared and the features extracted in the previous steps attributes this phase. Depends upon the algorithm selected, this model would be varied. The model is initialized by the appropriate class from sklearn and trained using the training data features and labels. After initialization, the method named 'fit', which is also from the sklearn, is used to train the model. This is the step which does the learning the relationship between the features and the labels. At the end, the model is evaluated based on the training data. That is done to make sure that the model has learned enough from the data. This step establishes the foundation for the model to from which the model can do predictions on new, unseen data with reasonable accuracy.

4) Making Predictions: After training the model, it is ready to make predictions. This involves using the trained model to predict whether the newly created email is spam or ham. The method named 'predict' is used for this purpose. A custom function 'predict_spam' is defined to take an email message as input, transform it into TF-IDF features using the trained vectorizer, and then with the help of the model, predict the label(spam or ham). The function returns 'spam' or 'ham' based on the model's prediction. This step demonstrates the model's ability to do the necessory classification for any email, which would make it fit in well with real-world implementations.

B. Three Models Selected

In the implementation of the email spam detection system, three distinct machine learning models were explored: Naive Bayes, Logistic Regression, and Support Vector Machine (SVM). By the use of appropriate class from the scilearn library, such that MultinomialNB(), LogisticRegression(), and svm.SVC(), the three models has been implemented. This implementation part has been explained as a third step in the above section.

Each model was build, trained, and then evaluated using the same dataset and performance metrics. This will allow us to compare the models and find the best suitable approach for email spam detection. This evaluation, and results have been explained in the following section.

IV. Comparison, Analysis, and Discussion

This section presents the results of the performace analysis done, then compare the model based on the results obtained. Addition to that, the discussion of various analysis done has been provided.

A. Comparison

The result obtained based on the matrix parameters and others are provided in Table I. The comparison of the models based on the results obtained has put forth several insights into the models' effectiveness. The observations have been summarized below.

TABLE I: Results of Performance Analysis

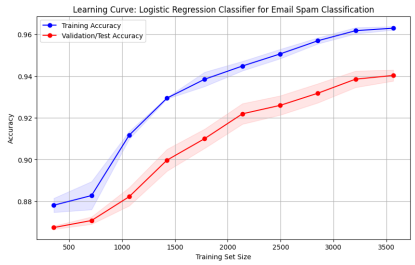
Metric	Logistic Regression	Naive Bayes	SVM
Evaluation Matrix			
Accuracy	0.97	0.97	0.98
Precision	0.96	0.97	0.98
Recall	1	0.96	1
F1 Score	0.98	0.98	0.99
Resource Utilization			
Training Time (s)	0.08	0.02	1.82
Inference Time (ms)	0.0021	0.0020	0.3470

1) Evaluation Matrix:

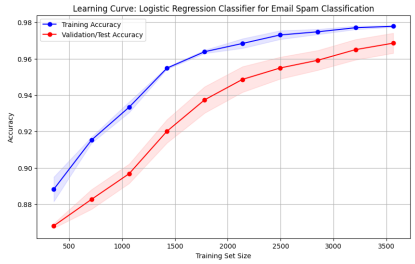
- Accuracy: All three models demonstrate high accuracy. SVM achieving the highest score of 0.98 among all, then followed by Logistic Regression and Naive Bayes at 0.97. This indicates that the models are adept at correctly classifying emails as spam or non-spam.
- Precision: SVM and Logistic Regression models exhibit similar precision scores of 0.98. This indicates that among the predictions done, majority of the predictions are positive predictions. Naive Bayes also performs well in precision with a score of 0.97, showing its effectiveness in correctly identifying spam emails.
- Recall: Both Logistic Regression and SVM achieve perfect recall scores of 1, indicating that they can identify all actual spam emails correctly. Naive Bayes achieves a slightly lower recall score of 0.96, indicating that it may miss a small fraction of spam emails.

- F1 Score: SVM outperforms the other models in F1 score with a score of 0.99. This means that SVM could maintain a significant balance between precision and recall. Both Logistic Regression and Naive Bayes perform well in F1 score, with scores of 0.98. This shows the relationship between precision and recall are good.

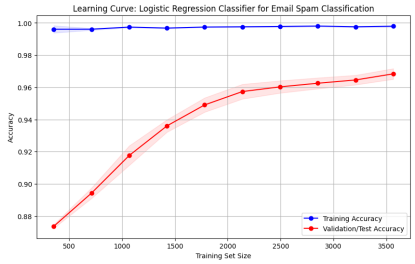
2) Learning Curve: The learning curve obtained for each model has been shown in Fig. 1.



(a) Logistic Regression



(b) Naive Bayes



(c) SVM

Fig. 1: Learning curves

The main observations and inferences are as listed below.

- *Logistic Regression:* The training and test accuracy both increase as the training set size grows. LR shows moderate performance, with both training and test accuracy converging around 94% to 96%. There is a slight gap between training and test accuracies, indicating some degree of overfitting but still acceptable generalization performance.
- *Naive Bayes:* The training and test accuracy both increase with the training set size, similar to LR. NB exhibits slightly better performance than LR, with both training and test accuracies converging around 96The gap between training and test accuracies is smaller compared to LR, indicating better generalization performance and less overfitting.
- *Support Vector Machine:* SVM shows the highest training accuracy among the three models, consistently above 99The training and test accuracies both increase with the training set size, but SVM shows a larger gap between them compared to LR and NB. Despite the gap, SVM achieves high test accuracy, indicating good generalization performance, although it might be prone to overfitting, especially with smaller training sets.

In summary, all three models show improvements in performance as the training set size increases, with Naive Bayes exhibiting slightly better generalization performance and Support Vector Machine demonstrating the highest training accuracy but a potential for overfitting. The choice of model would depend on various factors, including computational resources, interpretability requirements, and the trade-off between training and inference time.

B. Analysis and Discussion

1) Resource Utilization: Understanding the resource utilization of the model involves measuring the time taken for training and inference. Training time is the duration the model that takes to learn. This is measured using the 'time' module. Inference time is the time taken for the model to make predictions on the test data. Based on the Table I results, the following observations can be made.

- Training Time: Naive Bayes demonstrates the fastest training time of 0.02 seconds, followed by Logistic Regression at 0.08 seconds. SVM exhibits the longest training time at 1.82 seconds. This highlights Naive Bayes' efficiency in learning from the training data.
- Inference Time: Naive Bayes and Logistic Regression models show similar inference times, both below 0.002 seconds, indicating their efficiency in making predictions. SVM exhibits a higher inference time of 0.347 seconds, which is expected due to its more complex decision boundary.

2) Model Interpret-ability: Logistic Regression and Naive Bayes models offer greater interpretability compared to SVM, as they provide coefficients or probabilities that can be directly interpreted as the influence of features on the classification decision. This interpretability is valuable for understanding the factors contributing to spam classification and gaining insights into the underlying data patterns. In contrast, SVM's decision boundary in which where the hyperplane should be drawn is found to be a challenging task, or more optmization is required.

3) Model Deployment Considerations: The choice of model for deployment depends not only on its performance but also on factors like resources for computation, interpretability requirements, and the specific needs of the application. For instance, in latency-sensitive applications where real-time predictions are crucial, Naive Bayes or Logistic Regression may be preferred due to their faster inference times. Conversely, in applications where the highest accuracy is paramount, such as critical spam filtering systems, the additional computational cost of SVM may be justified.

Overall, while all three models demonstrate strong performance in email spam classification, each has its strengths and weaknesses. Naive Bayes excels in efficiency and simplicity. It is suitable for applications in the real world where computational resources are limited. Performance of Logistic Regression is impressive. It is a best choice for scenarios where understanding model decisions is important. SVM exhibits the highest accuracy. Ultimately, the model selection is based on the requirements of the application, and availability of the resources.

V. Future Directions with This Research

While this paper provide many observations in the effectiveness of traditional machine learning methods, there are several avenues for further research. The following are some comprehensive areas for future investigation:

1) Deep Learning Architectures: Explore the deep learning architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants (e.g., LSTM, GRU), in email spam classification is very important to be done. Since NLP, and Computer Vision uses these deep learning methods to solve complex problems, these algorithms may offer improved feature extraction and classification capabilities for spam detection.

2) Multi-Modal Learning: This research focused on text based dataset for classification. At present, email can be embedded with any digitally transferable information, like images, videos, zip files, and so on... Explore the integration of multiple modalities in email spam classification is an important aspect to think of. The innovative techniques in Deep learning can facilitate the fusion of heterogeneous data sources to capture rich contextual information and detect sophisticated spamming techniques, including image-based spam and phishing attacks.

3) Transfer Learning: Investigate the feasibility of transfer learning can be a further research. Pre-trained deep learning models can be fine-tuned or adapted to large set of email data. This will leverage the knowledge learned from related tasks and domains to improve spam detection performance.

4) Real-Time and Scalable Solutions: Design efficient and scalable model architectures is important to be discussed. With the increasing volume and velocity of email traffic, scalable solutions capable of processing large-scale email datasets in real-time are essential for effective spam filtering in practice.

In conclusion, further research in leveraging deep learning methods for email spam classification holds great promise in addressing the evolving challenges posed by sophisticated spamming techniques. The above mentioned comprehensive research directions can contribute to enhancing automating email spam detection which can be adaptable to newly or future of any spam techniques in email communication.

VI. Conclusion

In the world of email spam detection, we've explored three powerful models: Logistic Regression, Naive Bayes, and Support Vector Machine (SVM). Each has its own strengths and weaknesses, but they all aim to keep our inboxes clean from unwanted messages. After thorough analysis, we've found that SVM tends to be the most accurate, while Naive Bayes and Logistic Regression are faster and simpler to understand.

But choosing the right model isn't just about accuracy. It's also about considering factors like how fast it can learn and make predictions, and how easy it is to understand why it makes certain decisions. For example, if we need quick results and don't have much computing power, Naive Bayes might be the way to go. On the other hand, if we want the most accurate results and are willing to wait a bit longer, SVM could be a better choice.

In the end, there is no such thing like best solution for all the problems. It all depends on what we prioritize: speed, accuracy, or interpretability. Based on what one model can deliver and what it can not, we can make informed decisions to keep our email inboxes spam-free and our communication channels clear.

References

[1] Cox, D. The Regression Analysis of Binary Sequences. *Journal Of The Royal Statistical Society: Series B (Methodological)*. **20**, 215-232 (1958)

[2] Vapnik, V., Chervonenkis, A. & Drucker, H. Support vector method for function approximation, regression estimation, and signal processing. *Advances In Neural Information Processing Systems*. pp. 958-965 (1995)

[3] Kohavi, R. & John, G. Naive bayes for text classification. *Update: Applications Of Information Systems*. **11**, 85-98 (2001)

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. Scikit-learn: Machine learning in Python. *Journal Of Machine Learning Research*. **12** pp. 2825-2830 (2011)

[5] Tabish, A. Machine Learning Techniques for Spam Detection in Email. *Medium*.

[6] Manevitz, L., Eskin, E., Wachter, S. & Zimmermann, D. A learning framework for collaborative spam filtering. *Proceedings Of The Eleventh ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. pp. 107-114 (2003,8), https://ieeexplore.ieee.org/document/8674973

[7] Drucker, H., Vapnik, V. & Oh, J. Support vector machines for spam filtering. *Neural Networks*. **11**, 1613-1621 (1998)

[8] Sakaki, Y. & Sakurada, M. A hybrid approach to filtering spam emails. *Proceedings Of The 1st International Conference On Web Information Systems And Technologies (WEBIST 2002)*. **1** pp. 141-146 (2002)

[9] Authors, T. TensorFlow. (Google LLC,2024), https://www.tensorflow.org/, Accessed on June 6, 2024

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. & Others Scikit-learn: Machine Learning in Python. *Journal Of Machine Learning Research*. **12**, 2825-2830 (2011)

[11] Microsoft Visual Studio Code. (https://code.visualstudio.com/,2024), Accessed: 2024-06-06

[12] Wikipedia Tf-idf. , https://en.wikipedia.org/wiki/Tf