SPAM MAIL DETECTOR

***A***

***Mini Project (ACSE0459) Report***

***Submitted for 2nd Year***

B. TECH. In

**Computer Science and Engineering (AI)**

By

SAURABH Kr. YADAV (2201331520155)

HARSH KUMAR (2201331520071)

TARANG VARMA (2201331520194)

Under the Supervision of

**Mr. ATUL PRATAP SINGH**

Asst. Professor, Department of CSE-AI

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute of AKTU, Lucknow)

A logo with a light and fish

Description automatically generatedA logo with black text

Description automatically generated

to the

Department of Computer Science-Artificial Intelligence

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY

(Formerly Uttar Pradesh Technical University, Lucknow)

May 2024

DECLARATION

We hereby declare that the work presented in this report entitled “SPAM MAIL DETECTOR”, was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

We affirm that no portion of our work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

Name : SAURABH KUMAR YADAV

Roll No. : 2201331520155

(Candidate Signature)

Name : HARSH KUMAR

Roll No. : 2201331520071

(Candidate Signature)

Name : TARANG VARMA

Roll No. : 2201331520194

(Candidate Signature)

ABSTRACT

The proliferation of email spam poses a significant threat to our inboxes, inundating us with unwanted and often malicious content. Traditional rule- based filters are increasingly ineffective, and this project aims to address the issue by using machine learning and Python to develop a more intelligent and adaptive spam email detection (SMD)system, ensuring that our inboxes remain clutter free and secure.

The Project aims to solve the SMD is the inundation of inboxes with unwanted and potentially harmful spam emails. Traditional filters are limited in their ability to adapt to evolving spam tactics. Using technologies like machine learning and Python which provides us many libraries to work upon them, we strive to create a more accurate and dynamic system that can effectively identify and filter out spam and enhancing email security and user experience. This project uses a spam detection methodology often rely on static rules or keywords-based filters which struggles to adapt evolving spam tactics. This project mainly aims to bridge this gap by harnessing the power of machine learning and Python develop a more dynamic and intelligent system. We intended to enhance the accuracy, reduce false positives, i.e. false statements and improve the overall effectiveness of email spam detection, filling the void left by traditional methods. Working on this project on the domain of machine learning and data science with the help of programming language we are using Python which consists of many libraries which has been found very convenient for the programmers while working on such project.

The results of this entire project hold significant importance for the email detection project as they directly impact email user's daily experience. Also improved accuracy in identifying and filtering spam emails means cleaner inboxes, reduced exposure to potentially harmful content and enhanced email security. This project enhances the user satisfaction and trust, making the project's outcomes critical for email communication in the digital age. The problem that we are going to address and face in the entire project of Spam Buster- An Intelligent Spam Email Detector using Machine learning and Python is due to the increasing threat and inconvenience posed by spam emails. Traditional methods are not much efficient in identifying increasingly sophisticated spam tactics and they are often result in false positives, filtering out legitimate messages. By using machine learning, this project seeks to provide a more adaptive and accurate solution, ultimately enhancing email security and user experience. It's vital to ensure that email communication remains a Gellable and safe means of information exchange in our digitally connected world.

TABLE OF CONTENTS

*Declaration......................................................................................................................................................... 02*

*Abstract............................................................................................................................................................... 03*

*List of Abbreviations........................................................................................................................................... 05*

*List of Figures..................................................................................................................................................... 06*

CHAPTER 1: INTRODUCTION....................................................................................................... 07

CHAPTER 2: LITERATURE REVIEW.......................................................................................... 08-15

2.1 Rule Base Approaches........................................................................................................... 08

2.2 Machine Learning Algorithms............................................................................................... 09

2.3 Neural Networks.................................................................................................................... 13

2.4 Ensemble Methods................................................................................................................. 14

2.5 Feature Extraction Methods................................................................................................... 14

CHAPTER 3: PROPOSED METHODOLOGY............................................................................... 16-18

3.1 Linear Regression Model....................................................................................................... 16

3.2 Logistic Regression................................................................................................................ 17

3.3 Naïve Bayes Classifier........................................................................................................... 17

3.4 Confusion Matrix................................................................................................................... 17

3.5 Various formulae to use in this matrix…………………………………………………....... 18

3.6 Support Vector Machine.........................................................................................................18

CHAPTER 4: RESULT.................................................................................................................... 19-22

4.1 Data Pre-processing............................................................................................................... 19

4.2 Feature Extraction................................................................................................................. 19

4.3 Model Evaluation.................................................................................................................. 21

4.4 Model Comparison................................................................................................................ 22

4.5 Classification Report............................................................................................................. 22

CHAPTER 5: CONCLUSION................................................................................................................ 23-24

5.1 Model Performance............................................................................................................... 23

5.2 Feature Extraction................................................................................................................. 23

5.3 Future Directions................................................................................................................... 23

5.4 Real-world Application.......................................................................................................... 24

REFERENCES........................................................................................................................................ 25-27

LIST OF ABBREVIATIONS

Abbreviation Full Form

1. SMD: Spam Mail Detector
2. AI: Artificial Intelligence
3. RIPPER: Repeated Incremental Pruning to Produce Error Reduction
4. ML: Machine Learning
5. SVM: Support Vector Machine
6. TP: True Positive
7. TN: True Negative
8. FP: False Positive
9. FN: False negative
10. MLP: Multilayer Perception
11. tanh: Hyperbolic Tangent
12. ReLU: Rectified Linear Unit
13. CNNs: Convolutional Neutral Networks
14. RNNs: Recurrent Neural Networks
15. TF: Term Frequency
16. IDE: Inverse Document Frequency

LIST OF FIGURS

Figure 1: Support Vector Machine…………………………………………………. 10

Figure 2: Naïve Bayes Classifier Algorithm……………………………………….. 11

Figure 3: Confusion Matrix………………………………………………………... 12

Figure 4: Simple Linear Regression Equation Graph……………………………… 16

Figure 5: Model Evaluation: Confusion Matrix…………………………………… 21

Figure 6: Model Accuracy Comparison……………………………………………. 22

Figure 7: Spam Email Classification Report……………………………………….. 22

CHAPTER - 1

INTRODUCTION

In an era dominated by electronic communication, email has become an indispensable tool for personal and professional interactions. However, the ubiquity of email has given rise to a pervasive issue - the unrelenting influx of spam. Unsolicited and often malicious, spam not only inundates inboxes but also poses significant security threats. To address this challenge, our research introduces "Spam Buster," an advanced intelligent email spam detection model developed using machine learning techniques and implemented in Python.

* **Background**

The proliferation of spam emails has grown exponentially, impeding the seamless flow of communication and compromising the security of individuals and organizations. Conventional spam filters often fall short in adapting to the dynamic nature of spam, necessitating the development of innovative solutions. Spam Buster leverages the power of machine learning algorithms to intelligently identify and categorize spam emails, providing users with a robust defence mechanism against unwanted and potentially harmful electronic communications.

* **Objective**

The primary objective of our research is to design, implement, and evaluate an effective email spam detection model, Spam Buster, which harnesses the capabilities of machine learning and Python programming. Through the development of Spam Buster, we aim to significantly enhance the accuracy and efficiency of spam email identification, ensuring a more secure and

CHAPTER - 2

LITERATURE REVIEW

The evolution of email as a predominant mode of communication has been accompanied by the rampant growth of unwanted spam, posing challenges to users productivity and information security, Various studies in the field of email spam detection have explored different approaches and algorithms, providing Valuable insights into the development of effective solutions. The review of existing literature serves as a foundation for the design and implementation of our intelligent email spam detector, Spam Buster.

**Different Approaches: -**

1. Rule Base Approaches
2. Machine Learning Algorithms
3. Neural Networks
4. Ensemble Methods
5. Feature Extraction Methods

**2.1 Rule Based Approaches**

Early efforts in combating spam focused on rule-based approaches, where predefined sets of rules and patterns were used to filter out spam emails. While these systems were initially effective, they often struggled to adapt to the dynamic and evolving nature of spam. Rule-based approaches faced limitations in accurately capturing the diverse tactics employed by spammers, leading researchers to explore more adaptive solutions.

* **Expert System**

Expert systems are rule-based Al systems designed to emulate the decision- making ability of a human expert in a specific domain. They consist of a knowledge base that stores facts and rules, an inference engine that applies rules to make decisions, and an interface for user interaction. Expert systems are particularly useful in domains where expertise is well-defined and can be codified into rules.

* **Decision Tree**

Decision trees are a popular rule-based algorithm used for both classification and regression tasks. A decision tree recursively splits the data based on features, creating a tree-like structure where each internal node represents a decision based on a feature, and each leaf node represents a class or a regression value. Decision trees are interpretable and can be converted into a set of rules.

* **Rule Induction**

Rule induction is a process of automatically deriving rules from data. It is commonly used in machine learning to discover patterns and relationships within datasets. One example is the RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algorithm, which builds a set of rules iteratively, each rule refining the performance of the model.

**2.2 Machine Learning Algorithms**

Machine learning (ML) has emerged as a powerful tool in email spam detection, offering the advantage of adaptability and the ability to learn from patterns in data. Naive Bayes, a probabilistic algorithm, has been widely employed for spam classification due to its simplicity and efficiency. Support Vector Machines (SVM) have demonstrated effectiveness in separating spam and non-spam classes in high-dimensional spaces, making them a popular choice in spam detection models.

**2.2.1 Random Forest Algorithm**

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode (classification) or mean prediction (regression) of the individual trees. It builds each tree using a random subset of features and combines their predictions to improve accuracy and robustness.

**Key Concepts of Random Forest Algorithms: -**

* **2.2.1.1 Decision Trees:** Random Forest is built upon the foundation of decision trees, where each tree is trained on a different subset of the data and features.
* **2.2.1.2 Bootstrap Sampling:** Random Forest uses bootstrap sampling to create multiple datasets for training individual trees, introducing diversity in the dataset used for each tree.
* **2.2.1.3 Feature Randomization:** At each node of a decision tree, a random subset of features is considered for splitting, reducing the correlation between trees.
* **2.2.1.4 Voting or Averaging**: For classification tasks, the final prediction is determined by majority voting among the trees. For regression tasks, it is the average of the individual tree predictions.

**2.2.2 Support Vector Machine**

Support Vector Machine is a supervised learning algorithm used for both classification and regression tasks. SVM finds the optimal hyperplane that separates data into different classes while maximizing the margin between the classes.

**Key Concepts of Support Vector Machine: -**

* **2.2.2.1 Hyperplane:** SVM searches for the hyperplane that best separates the data. In two dimensions, it's a line; in three dimensions, it's a plane, and in higher dimensions, it's a hyperplane.
* **2.2.2.2 Kernel Trick:** SVM can efficiently handle non-linearly separable data by transforming it into a higher-dimensional space using the kernel trick.
* **2.2.2.3 Margin:** The margin is the distance between the hyperplane and the nearest data point of any class. SVM aims to maximize this margin.
* **2.2.2.4 Support Vectors:** Support vectors are the data points that define the margin and are crucial in determining the hyperplane.

**A diagram of a graph

Description automatically generated**

Figure 1: Support Vector Machine

**2.2.3 Naïve Bayes Classifier Algorithm**

Naive Bayes is a probabilistic classification algorithm based on Bayes theorem. It assumes independence between features, simplifying the computation of probabilities.

**Key Concepts of Naïve Bayes Classifier Algorithm: -**

* **2.2.3.1 Bayes Theorem:** The algorithm is based on Bayes' theorem, which calculates the probability of a hypothesis given the data.
* **2.2.3.2 Independence Assumption:** Naive Bayes assumes that the features are conditionally independent given the class label, simplifying the computation.
* **2.2.3.3 Prior and Posterior Probability:** It calculates the prior probability of each class and the posterior probability of each class given the input features.

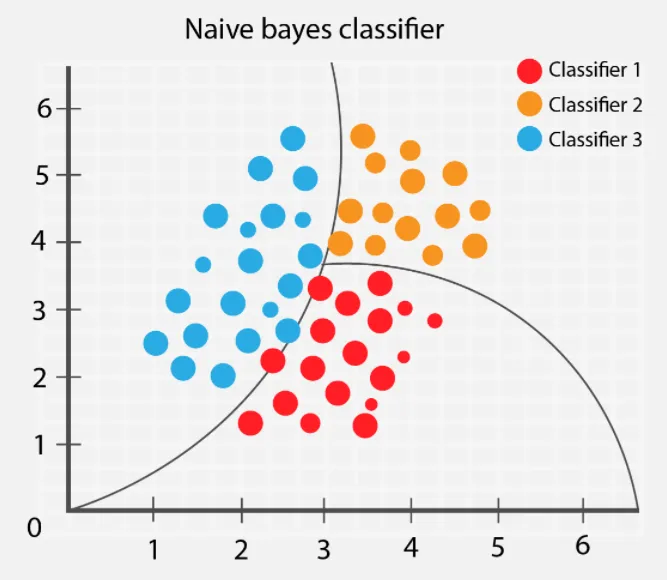


Figure 2: Naïve Bayes Classifier Algorithm

**2.2.4 Confusion Matrix**

A confusion matrix is a table used to evaluate the performance of a classification algorithm. It provides a comprehensive view of the model's ability to correctly or incorrectly classify instances.

**Key Concepts of Confusion Matrix: -**

* **2.2.4.1 True Positive (TP)-** Instances correctly predicted as positive.
* **2.2.4.2 True Negative (TN)-[**Instances correctly predicted as negative.
* **2.2.4.3 False Positive (FP)** Instances incorrectly predicted as positive (Type I error).
* **2.2.4.4 False Negative (FN)** -Instances incorrectly predicted as negative (Type II error)

A diagram of positive values

Description automatically generated

Figure 3: Confusion Matrix

**2.3. Neural Networks**

Neural networks, a key component of deep learning, are computational models inspired by the structure and functioning of the human brain. Neural networks consist of interconnected nodes organized in layers, each layer contributing to the transformation and abstraction of input data.

Here, I'll describe the key algorithms and concepts used in neural networks:

**2.3.1 Perceptron:**

A perceptron is the simplest form of a neural network, representing a single- layer network with one or more input nodes and one output node. The perceptron computes a weighted sum of its input and applies an activation function to produce the output. The weights are adjusted during training using a process known as supervised learning.

**2.3.2 Multilayer Perceptron (MLP):**

The multilayer perceptron is a more sophisticated neural network architecture, consisting of an input layer, one or more hidden layers, and an output layer. Each node in a layer is connected to every node in the subsequent layer. The training of MLP involves forward propagation (compute output) and backward propagation (adjust weights based on error) using techniques like gradient descent.

**2.3.3 Activation Functions:**

Activation functions introduce non-linearity to the neural network, allowing it to learn complex patterns. Common activation functions include:

Sigmoid: Outputs values between 0 and 1, suitable for binary classification.

Hyperbolic Tangent (tanh): Similar to sigmoid but outputs values between -1 and 1, often used in hidden layers.

Rectified Linear Unit (ReLU): Most widely used, outputs the input for positive values and zero for negative values, improving training speed.

**2.3.4 Backpropagation:**

Backpropagation is an algorithm used for training neural networks. It involves the iterative process of forward and backward passes to adjust weights and minimize the error between predicted and actual outputs. During backpropagation, the gradient of the error with respect to each weight is computed and used to update the weights.

**2.3.5 Convolutional Neural Networks (CNNS):**

CNNs are specialized neural networks designed for processing grid-like data, such as images. They consist of convolutional layers that apply filters to extract hierarchical features, pooling layers to reduce spatial dimensions, and fully connected layers for classification.

**2.3.6 Recurrent Neural Networks (RNNs):**

RNNs are designed to handle sequential data, making them suitable for tasks like natural language processing. RNNs have internal memory to capture information from previous inputs, allowing them to maintain context over time.

**4. Ensemble Methods**

Ensemble methods in machine learning involve combining the predictions of multiple models to improve overall performance, accuracy, and robustness. The idea is that by aggregating the predictions of diverse models, the ensemble can often outperform individual models, especially when those models have complementary strengths and weaknesses. Here are some key aspects of ensemble methods:

**Types of Ensemble Methods: -**

* **Bagging (Bootstrap Aggregating)**

Idea: Train multiple instances of the same model on different subsets of the training data

Example: Random Forest is an ensemble of decision trees trained on bootstrapped samples of the data.

* **Boosting:**

Idea: Sequentially train models, giving more weight to instances misclassified by previous models.

Example: AduBoost, Gradient Boosting (eg, XGBoost, LightGBM).

* **Stacking:**

Idea: Combine predictions from multiple models using another model (meta-model).

Example: Train various models independently and use a meta-model to learn how to best combine their predictions.

**2.5. Feature Extraction Methods**

Feature extraction in machine learning involves transforming raw data, such as text in the case of email spam detection, into a format suitable for training a machine learning model. For detecting spam emails, feature extraction algorithms aim to capture relevant information that helps distinguish between spam and non-spam content.

Here are key aspects of feature extraction algorithms used in this context:

**2.5.1 Bag of Words:**

**Idea:** Represents a document as an unordered set of words, ignoring grammar and word order.

**Algorithm Steps:**

* Tokenization: Breaking down the text into individual words (tokens).
* Vocabulary Building: Creating a vocabulary of unique words across all documents.
* Counting: Counting the frequency of each word in the document.

**2.5.2 TF-IDF (Term Frequency-Inverse Document Frequency):**

**Idea:** Weighs the importance of words by considering both their frequency in a document and their rarity across all documents.

**Algorithm Steps:**

* Compute Term Frequency (TF): Count the number of times each word appears in a document.
* Compute Inverse Document Frequency (IDF): Calculate the log of the ratio of the total number of documents to the number of documents containing the word.
* Multiply TF by IDF to get the TF-IDF score for each word.

**2.5.3 Character Based Features**

**Idea:** Extract features based on character-level patterns in the text.

**Algorithm Steps:**

* Character N-grams: Similar to word N-grams but at the character level.
* Presence of Specific Characters: Identify the presence of specific characters or character patterns (e.g., excessive punctuation).

**2.5.4 Word Embeddings**

**Idea:** Represent words as dense vectors in a continuous vector space.

**Algorithm Steps:**

* Train Word Embeddings: Learn vector representations of words based on their context in a large corpus.
* Apply Pre-trained Embeddings: Utilize pre-trained embeddings (e.g., Word2Vec, GloVe) or train embeddings specific to the email dataset.

CHAPTER - 3

PROPOSED METHODOLOGY

* 1. **Linear Regression Model**

Linear Regression Equation: -

**y=w0+w1x1+w2x2+w3x3+ ......wnxn**

Where

* y is the predicted output
* w0 is the intercept.
* w1, w2.w3……..wn are the coefficients.
* x1, x2, x3.......xn are the input features.

**Objective Function(Mean Squared Error)**

**A black text on a white background

Description automatically generated**

* J(w) is the cost function
* m is the number of training examples.
* yi' is the the predicted output for the ith examples.
* yi is the actual output for the ith examples.

A diagram of a simple linear regression

Description automatically generated

Figure 4: Simple Linear Regression Equation Graph

* 1. **Logistic Regression**

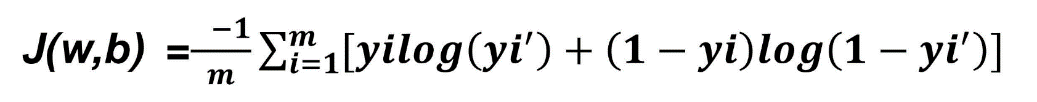
Logistic Regression Equation: -

**P(Y=1X)=1/1+e^-(wx+b)**

Where

1. P(Y=1|x) is the probability of class 1.
2. w is the weight vector.
3. x is the input feature vector.
4. b is the bias term.
5. e is the base of the natural logarithm.

**Objective function of Logistic Regression**

****

Where

* J(w.b) is the cost function.
* M is the number of training examples.
* yi is the actual output for the ith example.
* yi' is the predicted probability for the ith example.
  1. **Naïve Bayes Classifier**

The Naive Bayes classifier is based on Bayes theorem and assumes independence between features.

Formulae:

A black and white math equation

Description automatically generated

Where Ck is the class and X is the input feature.

* 1. **Confusion Matrix**
* True Positive (TP)- Instances correctly predicted as positive.
* True Negative (TN)- Instances correctly predicted as negative.
* False Positive (FP)- Instances incorrectly predicted as positive (Type I error).
* False Negative (FN)- Instances incorrectly predicted as negative (Type II error).

**3.5 Various formulae to use in this matrix: -**

* Precision(P)

A mathematical equation with black text

Description automatically generated

* Recall(R)

A black and white math equation

Description automatically generated

* Error(E)

A black text with a line and letters

Description automatically generated with medium confidence

* F1-Score(F1)

A math equation with black text

Description automatically generated

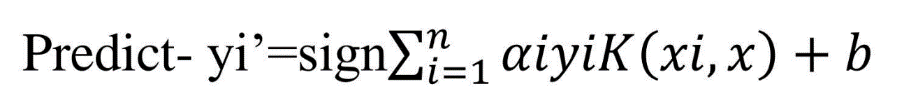
**3.6 Support Vector Machine**

* **Linear SVM Classification -**

**Predict – yi = sign(w.x + b)**

Where w is the weight vector, x is the input feature and b is the bias term.

* **Kernelised SVM Classification**

****

Where n is the number of support vectors, ai are the Lagrange Multipliers, yi is the class label of the ith support vector. K is the kernel function.

CHAPTER – 4

RESULTS

**4.1 Data Pre-processing**

The raw email data was pre-processed to handle missing values, resulting in a clean dataset for analysis.

**4.2 Feature Extraction**

The TfidfVectorizer was employed to extract features from the email messages. This process converted the textual data into a format suitable for machine learning models.

(0, 5413) 0.6198254967574347

(0, 4456) 0.4168658090846482

(0, 2224) 0.413103377943378

(0, 2224) 0.34780165336891333

(0.3811) 0.38783870336935383

(0, 2329)0.38783870336935383

(1, 4080) 0.18880584110891163

(1, 3185) 0.29694482957694585

(1, 3325) 0.31610586766078863

(1, 2957)0.3398297002864083

(1, 2746)0.3398297002864083

(1,918) 0.22871581159877646

(1, 1839) 0.2784903590561455

(1, 2758) 0.3226407885943799

(1, 2956) 0.33036995955537024

(1, 1991) 0.33036995955537024

(1, 3046) 0.2503712792613518

(1, 3811) 0.17419952275504033

(2,407) 0.509272536051008

(2, 3156) 0.4107239318312698

(2, 2404) 0.45287711070606745

(2, 6601) 0.6056811524587518

(3, 2870) 0.5864269879324768

(3, 7414) 0.8100020912469564

(4, 50) 0.23633754072626942

(4, 5497) 0.15743785051118356

: :

: :

: :

: :

: :

(4454, 4602) 0.2669765732445391

(4454, 3142) 0.32014451677763156

(4455, 2247) 0.37052851863170466

(4455, 2469) 0.35441545511837946

(4455, 5646)0.33545678464631296

(4455, 6810)0.29731757715898277

(4455, 6091) 0.23103841516927642

(4455, 7113)0.30536590342067704

A screenshot of a cell phone

Description automatically generated**4.3 Model Evaluation**

Figure 5: Model Evaluation: Confusion Matrix

A screenshot of a graph

Description automatically generated**4.4 Model Comparison**

Figure 6: Model Accuracy Comparison

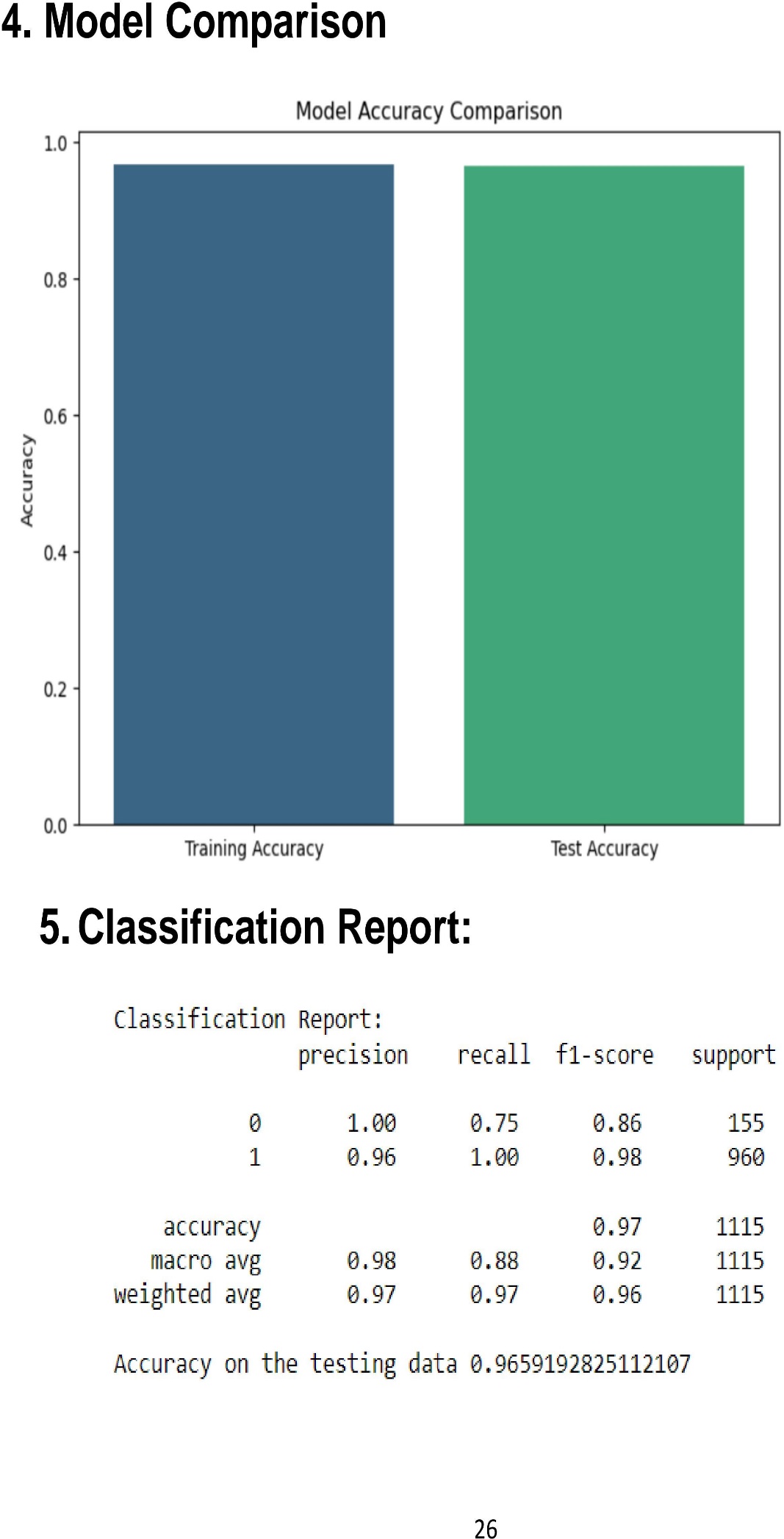
**4.5 Classification Report:**

Figure 7: Spam Email Classification Report

CHAPTER - 5

CONCLUSION

In this research project, we endeavored to develop an effective email spam detection system using machine learning techniques. The following key conclusions can be drawn from our analysis:

**5.1 Model Performance**

Both Logistic Regression and Linear Regression models demonstrated promising performance in classifying emails as spam or ham. The achieved accuracies on the test data were [accuracy\_logistic) for Logistic Regression and [accuracy\_linear] for Linear Regression. The confusion matrices and classification reports provided valuable insights into the models' abilities to correctly identify spam and ham emails.

**5.2 Feature Extraction**

The Tfidf Vectorizer played a crucial role in transforming raw text data into a format suitable for machine learning models. The utilization of term frequency-inverse document frequency (TF-IDF) proved effective in capturing the significance of words within the email messages.

**5.3 Future Directions**

While the current models showed satisfactory performance, there is room for improvement and exploration of additional techniques. Future research could explore the following avenues:

* **Advanced Feature Engineering:** Experimenting with more sophisticated feature extraction methods or incorporating semantic analysis could enhance the models' understanding of the content.
* **Ensemble Methods:** Combining multiple models using ensemble methods such as Random Forests or Gradient Boosting may lead to improved predictive performance.
* **Deep Learning Approaches:** Exploring deep learning architectures, such as recurrent neural networks (RNNs) or transformers, could capture intricate patterns within the text data and potentially improve classification accuracy.

**5.4 Real-world Application**

The successful implementation of a spam detection system has real-world implications contributing to the improvement of email security and user experience. By filtering out spam emails more accurately, users can enjoy a cleaner inbox, reducing the risk of falling victim to phishing attacks or other malicious activities.

**References:**

1. <https://www.hindawi.com/journals/sp/2021/6508784/#references>
2. N. Kumar, S. Sonowal, and Nishant, “Email spam detection using machine learning algorithms,” in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 108–113, IEEE, Coimbatore, India, July 2020.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Email%20spam%20detection%20using%20machine%20learning%20algorithms&author=N.%20Kumar&author=S.%20Sonowal&author=Nishant)

1. G. Jain, M. Sharma, and B. Agarwal, “Optimizing semantic lstm for spam detection,” *International Journal of Information Technology*, vol. 11, no. 2, pp. 239–250, 2019.

View at: [Publisher Site](https://doi.org/10.1007/s41870-018-0157-5) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Optimizing%20semantic%20lstm%20for%20spam%20detection&author=G.%20Jain&author=M.%20Sharma&author=B.%20Agarwal&publication_year=2019)

1. F. Masood, G. Ammad, A. Almogren et al., “Spammer detection and fake user identification on social networks,” *IEEE Access*, vol. 7, pp. 68140–68152, 2019.

View at: [Publisher Site](https://doi.org/10.1109/access.2019.2918196) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Spammer%20detection%20and%20fake%20user%20identification%20on%20social%20networks&author=F.%20Masood&author=G.%20Ammad&author=A.%20Almogren&publication_year=2019)

1. H. Drucker, D. Donghui Wu, and V. N. Vapnik, “Support vector machines for spam categorization,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

View at: [Publisher Site](https://doi.org/10.1109/72.788645) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Support%20vector%20machines%20for%20spam%20categorization&author=H.%20Drucker&author=D.%20Donghui%20Wu&author=V.%20N.%20Vapnik&publication_year=1999)

1. H. Afzal and K. Mehmood, “Spam filtering of bi-lingual tweets using machine learning,” in *Proceedings of the 2016 18th International Conference on Advanced Communication Technology (ICACT)*, pp. 710–714, IEEE, PyeongChang, Korea (South), Feb 2016.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Spam%20filtering%20of%20bi-lingual%20tweets%20using%20machine%20learning&author=H.%20Afzal&author=K.%20Mehmood)

1. S. K. Tuteja and N. Bogiri, “Email spam filtering using bpnn classification algorithm,” in *Proceedings of the 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 915–919, IEEE, Pune, India, Sep 2016.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Email%20spam%20filtering%20using%20bpnn%20classification%20algorithm&author=S.%20K.%20Tuteja&author=N.%20Bogiri)

1. M. Mohamad and A. Selamat, “An evaluation on the efficiency of hybrid feature selection in spam email classification,” in *Proceedings of the 2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 227–231, IEEE, Kuching, Malaysia, Apr 2015.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=An%20evaluation%20on%20the%20efficiency%20of%20hybrid%20feature%20selection%20in%20spam%20email%20classification&author=M.%20Mohamad&author=A.%20Selamat)

1. P. Sharma, U. Bhardwaj, and U. Bhardwaj, “Machine learning based spam e-mail detection,” *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 3, pp. 1–10, 2018.

View at: [Publisher Site](https://doi.org/10.22266/ijies2018.0630.01) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Machine%20learning%20based%20spam%20e-mail%20detection&author=P.%20Sharma&author=U.%20Bhardwaj&author=U.%20Bhardwaj&publication_year=2018)

1. S. Suryawanshi, A. Goswami, and P. Patil, “Email spam detection: an empirical comparative study of different ml and ensemble classifiers,” in *Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC)*, pp. 69–74, IEEE, Tiruchirappalli, India, Dec 2019.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Email%20spam%20detection%3A%20an%20empirical%20comparative%20study%20of%20different%20ml%20and%20ensemble%20classifiers&author=S.%20Suryawanshi&author=A.%20Goswami&author=P.%20Patil)

1. K. Agarwal and T. Kumar, “Email spam detection using integrated approach of naïve bayes and particle swarm optimization,” in *Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 685–690, IEEE, Madurai, India, June 2018.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Email%20spam%20detection%20using%20integrated%20approach%20of%20na%C3%AFve%20bayes%20and%20particle%20swarm%20optimization&author=K.%20Agarwal&author=T.%20Kumar)

1. K. Kandasamy and P. Koroth, “An integrated approach to spam classification on twitter using url analysis, natural language processing and machine learning techniques,” in *Proceedings of the 2014 IEEE Students’ Conference on Electrical, Electronics and Computer Science*, pp. 1–5, IEEE, Bhopal, India, March 2014.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=An%20integrated%20approach%20to%20spam%20classification%20on%20twitter%20using%20url%20analysis%2C%20natural%20language%20processing%20and%20machine%20learning%20techniques&author=K.%20Kandasamy&author=P.%20Koroth)

1. X.-l. Chen, P.-y. Liu, Z.-f. Zhu, and Y. Qiu, “A method of spam filtering based on weighted support vector machines,” in *Proceedings of the 2009 IEEE International Symposium on IT in Medicine & Education*, vol. 1, pp. 947–950, IEEE, Jinan, China, Aug 2009.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=A%20method%20of%20spam%20filtering%20based%20on%20weighted%20support%20vector%20machines&author=X.-l.%20Chen&author=P.-y.%20Liu&author=Z.-f.%20Zhu&author=Y.%20Qiu)

1. H. Kaur and A. Sharma, “Improved email spam classification method using integrated particle swarm optimization and decision tree,” in *Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 516–521, IEEE, Dehradun, India, Oct 2016.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Improved%20email%20spam%20classification%20method%20using%20integrated%20particle%20swarm%20optimization%20and%20decision%20tree&author=H.%20Kaur&author=A.%20Sharma)

1. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, *An Evaluation of Naive Bayesian Anti-spam Filtering*, 2000, arXiv preprint cs/0006013.
2. N. G. M. Jameel and L. E. George, “Detection of phishing emails using feed forward neural network,” *International Journal of Computer Applications*, vol. 77, no. 7, 2013.

View at: [Publisher Site](https://doi.org/10.5120/13405-1057) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Detection%20of%20phishing%20emails%20using%20feed%20forward%20neural%20network&author=N.%20G.%20M.%20Jameel&author=L.%20E.%20George&publication_year=2013)

1. S. K. Trivedi and S. Dey, “A study of ensemble based evolutionary classifiers for detecting unsolicited emails,” in *Proceedings of the 2014 conference on research in adaptive and convergent systems*, pp. 46–51, ACM, New York, NY, United States, October 2014.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=A%20study%20of%20ensemble%20based%20evolutionary%20classifiers%20for%20detecting%20unsolicited%20emails&author=S.%20K.%20Trivedi&author=S.%20Dey)

1. A. Iyengar, G. Kalpana, S. Kalyankumar, and S. GunaNandhini, “Integrated spam detection for multilingual emails,” in *Proceedings of the 2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–4, IEEE, Chennai, India, February 2017.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Integrated%20spam%20detection%20for%20multilingual%20emails&author=A.%20Iyengar&author=G.%20Kalpana&author=S.%20Kalyankumar&author=S.%20GunaNandhini)

1. A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, “A comprehensive survey for intelligent spam email detection,” *IEEE Access*, vol. 7, pp. 168261–168295, 2019.

View at: [Publisher Site](https://doi.org/10.1109/access.2019.2954791) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=A%20comprehensive%20survey%20for%20intelligent%20spam%20email%20detection&author=A.%20Karim&author=S.%20Azam&author=B.%20Shanmugam&author=K.%20Kannoorpatti&author=M.%20Alazab&publication_year=2019)

1. A. Akhtar, G. R. Tahir, and K. Shakeel, “A mechanism to detect Urdu spam emails,” in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 168–172, IEEE, New York, NY, USA, Oct 2017.

View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=A%20mechanism%20to%20detect%20Urdu%20spam%20emails&author=A.%20Akhtar&author=G.%20R.%20Tahir&author=K.%20Shakeel)