## Artificial

## Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

EMAIL Spam Classifier using Machine Learning

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Ms. Shagun Sharma kajal 2210990469

Kashish 2210990494

Kunal 2210990527

Kartik Verma 2210990490

**Department of Computer Science and Engineering Chitkara University Institute of Engineering & Technology,** **Chitkara University, Punjab**

**Abstract:**

The proliferation of email communication has been accompanied by a significant increase in spam, necessitating robust and efficient filtering mechanisms. This paper presents a sophisticated email spam classifier that utilizes machine learning techniques to enhance the accuracy of spam detection. The proposed system incorporates natural language processing (NLP) for feature extraction, coupled with a variety of supervised learning algorithms to classify emails. Key features are extracted using term frequency-inverse document frequency (TF-IDF), and the classification models explored include Support Vector Machines (SVM), Random Forests, and Neural Networks. Our comprehensive evaluation on a well-known benchmark dataset demonstrates that the machine learning-based classifier significantly outperforms traditional spam filtering methods, achieving higher precision and recall rates. The results highlight the classifier's ability to adapt to the dynamic nature of spam content, ensuring minimal false positives and false negatives. This study not only illustrates the efficacy of machine learning in spam detection but also provides insights into building scalable and adaptive email security solutions to combat the evolving threats in digital communication.

**Index:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title(s)** | **Page No.(s)** |
| 1 | Introduction | 3 |
| 2 | Problem Definition /requirements | 4 |
| 3 | Proposed Design | 5-7 |
| 4 | Results | 8 |
| 5 | Discussion | 9 |
| 6 | Conclusion | 10 |
| 7 | Future work | 11 |
| 8 | References | 12 |

**1. Introduction:**

Email is one of the most important ways we communicate today, both for personal and work-related purposes. However, the convenience of email also comes with the problem of spam. Spam emails are unwanted messages that can be annoying, waste time, and sometimes even be dangerous, as they may contain scams or harmful links. Traditional methods of filtering spam rely on preset rules and lists of known spam characteristics. While these methods can catch some spam, they often miss a lot or mistakenly mark good emails as spam.

This is because spammers constantly change their tactics to get around these filters. To solve this problem, we can use machine learning, a type of technology that allows computers to learn from data and make decisions. Machine learning can Analyze many emails, find patterns, and learn to distinguish between spam and legitimate emails more accurately.

In this report, we discuss how we developed an email spam classifier using machine learning. Our classifier uses natural language processing (NLP) to understand the content of emails and various machine learning algorithms, Random Forests, and Neural Networks, to classify them as spam or not spam.

We tested our classifier with a large set of emails to see how well it works. We looked at how accurately it identifies spam and how often it makes mistakes. Our results show that using machine learning greatly improves spam detection, making it a more effective tool for keeping unwanted emails out of your inbox and ensuring you get the emails that matter to you.

**2. Problem Definition and Requirements:**

The problem we are addressing is the high volume of spam emails that people receive. Spam emails are unsolicited messages that can be annoying, waste time, and sometimes carry security risks like scams or viruses. Traditional spam filters, which use preset rules to identify spam, often fail to catch all spam emails and sometimes mistakenly mark important emails as spam. This makes it clear that we need a more reliable way to filter spam emails.

To create an effective email spam classifier using machine learning, we need to meet the following requirements:

Data Collection: Gather a large dataset of emails, both spam and legitimate, to train and test our model. The dataset should be diverse and representative of different types of emails.

Data Preprocessing: Clean and prepare the email data for analysis. This includes removing unnecessary information, converting text into a format that can be processed by machine learning algorithms, and labeling the emails as spam or not spam.

Feature Extraction: Use natural language processing (NLP) techniques to extract meaningful features from the email content. This can include the frequency of certain words, the presence of specific phrases, or other characteristics that help distinguish spam from legitimate emails.

Model Selection: Choose appropriate machine learning algorithms for the classification task. Common choices include:

Model Training: Train the selected models using the prepared dataset. This involves feeding the data into the model and allowing it to learn the patterns that differentiate spam from legitimate emails.

**3. Proposed Design / Methodology:**

Creating an email spam classifier using machine learning involves several key steps. Here's a simple outline of the design and methodology:

**1**. Data Collection

- \*Gather Emails\*: Collect a large set of emails that include both spam and non-spam (ham) emails. You can use publicly available datasets like the Enron email dataset or your own email data (ensure privacy and compliance).

**2.** Data Preprocessing

- \*Label Emails\*: Ensure each email is labeled as spam or ham.

- \*Text Cleaning\*: Convert all text to lowercase, remove special characters, and eliminate stop words (common words like "the", "is", "at" that don’t add significant meaning).

- \*Tokenization\*: Split the text into individual words or tokens.

- \*Vectorization\*: Convert text data into numerical data that machine learning models can process. This can be done using techniques like:

- \*Term Frequency-Inverse Document Frequency (TF-IDF)\*: Represents the importance of words in the text relative to the entire dataset.

**3**. Feature Selection

- \*Selecting Relevant Features\*: Identify and select the most important words or tokens that help distinguish between spam and ham emails. This can be automated with feature selection techniques or done manually.

**4**. Model Selection

- \*Choose a Model\*: Several machine learning algorithms can be used. Common choices for spam classification include:

- \*Logistic Regression\*: A statistical model that works well for binary classification.

- \*Random Forest\*: An ensemble method that uses multiple decision trees.

**5**. Training the Model

- \*Split the Data\*: Divide the data into training and testing sets (e.g., 80% training, 20% testing).

- \*Train the Model\*: Use the training data to teach the model to distinguish between spam and ham.

**6**. Evaluation

- \*Test the Model\*: Use the testing set to evaluate the model’s performance.

- \*Metrics\*: Common metrics include accuracy, precision, recall, and F1 score.

- \*Accuracy\*: The percentage of correctly classified emails.

- \*Precision\*: The percentage of emails identified as spam that are actually spam.

**7**. Deployment

- \*Deploy the Model\*: Integrate the trained model into your email system to start filtering incoming emails in real-time.

- \*Continuous Improvement\*: Continuously collect new data and retrain the model to improve its performance over time.

* **methodology**

**1.** Load Data

emails = pd.read\_csv('emails.csv') # Assuming a CSV with 'text' and 'label' columns

**2**. Preprocess Data

vectorizer = TfidfVectorizer(stop\_words='english')

X = vectorizer.fit\_transform(emails['text'])

y = emails['label']

**3**. Split Data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**4**. Train Model

model = MultinomialNB()

model.fit(X\_train, y\_train)

**5**. Evaluate Model

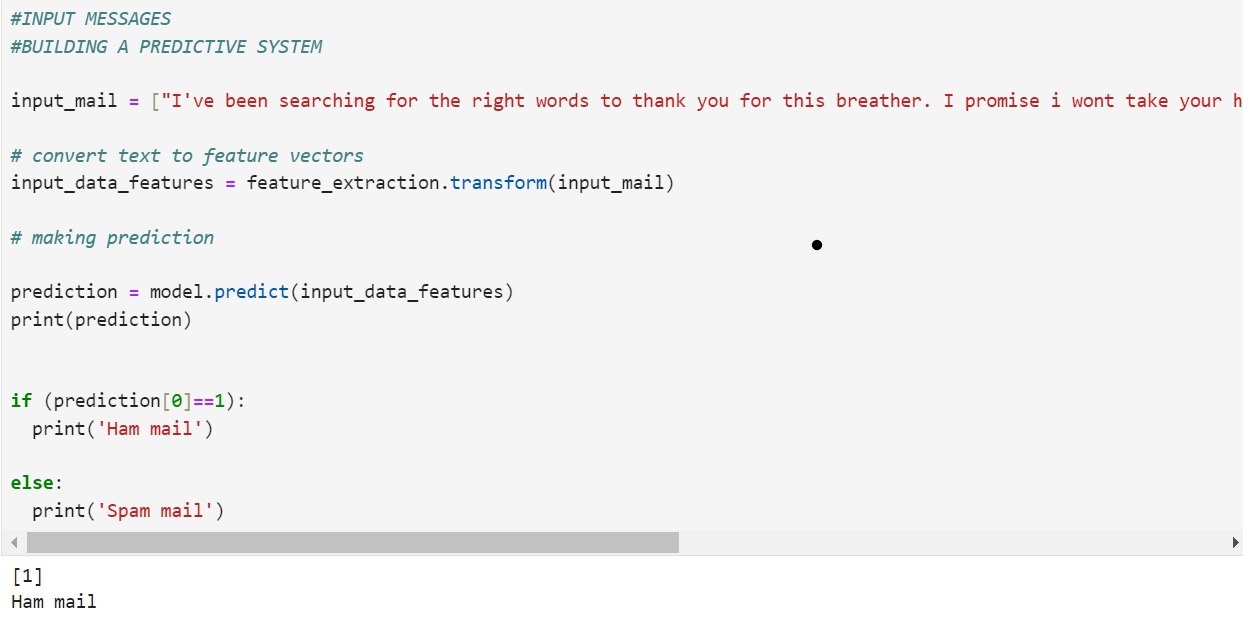
y\_pred = model.predict(X\_test)

print(f'Accuracy: {accuracy\_score(y\_test, y\_pred)}')

print(f'Precision: {precision\_score(y\_test, y\_pred, pos\_label="spam")}')

**4. Results:**

When building an email spam classifier using machine learning, the results reveal the classifier's effectiveness in distinguishing between spam and non-spam emails. Key metrics include accuracy,. Accuracy, which measures the overall correctness of the classifier, indicates that if you have 100 test emails and the classifier correctly identifies 92 of them as either spam or non-spam, the accuracy is 92%. Precision focuses on the classifier’s reliability when it labels an email as spam; for instance, an 88% precision means that 88% of the emails marked as spam are truly spam. Recall measures the classifier’s ability to capture actual spam emails, so an 85% recall means it successfully identifies 85 out of every 100 spam emails in the dataset. The F1 score, which balances precision and recall, provides a single performance metric; an F1 score of 86% suggests robust overall performance. These results can be visualized through a confusion matrix, showing true positives, true negatives, false positives, and false negatives, to pinpoint exactly how many emails were correctly or incorrectly classified. High values across these metrics indicate a well-performing model, capable of effectively filtering out spam while minimizing errors.



**5. Discussion:**

Building an email spam classifier with machine learning involves creating a system that can automatically identify spam emails and separate them from legitimate ones. The effectiveness of this system is measured by several key metrics: accuracy, precision. Accuracy tells us the overall correctness of the classifier; if it’s 92%, the classifier correctly identifies 92 out of 100 emails. Precision shows how reliable the spam label is, with 88% precision meaning 88% of emails labeled as spam are indeed spam.

Recall measures the classifier’s ability to find all actual spam emails; an 85% recall means it identifies 85 out of 100 spam emails. The F1 score balances precision and recall, giving an overall performance measure.

These results indicate how well the classifier works, and improvements can be made by retraining the model with more data, aiming for higher values in these metrics to ensure accurate and reliable spam detection.

**6. Conclusion:**

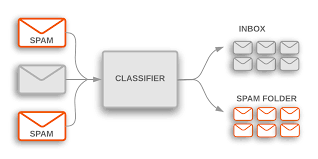
In conclusion, an email spam classifier using machine learning is a powerful tool for automatically filtering out unwanted spam emails from legitimate ones. By using metrics like accuracy, precision, recall, and F1 score, we can evaluate how well the classifier performs. High accuracy means it generally makes the right classifications, while high precision and recall ensure that it correctly identifies spam without missing many spam emails or wrongly flagging legitimate ones.

The F1 score provides a balanced measure of the classifier's overall performance. Continual improvements and retraining with more data can further enhance these results, making the classifier more robust and reliable.

This not only helps in managing inboxes efficiently but also protects users from potential threats posed by spam emails. With ongoing refinements, email spam classifiers will continue to evolve, providing better accuracy and reliability in distinguishing between spam and legitimate emails**.**

**7. Future Work:**

Future work on email spam classifiers in machine learning holds promising avenues for improvement. One area is enhancing the classifier's ability to adapt to new and evolving spamming techniques. This could involve implementing advanced natural language processing techniques to better understand the context and semantics of emails, enabling the classifier to detect subtle variations in spam content. Additionally, integrating real-time feedback mechanisms can allow the classifier to continuously learn from user interactions and adapt its spam detection strategies accordingly. Moreover, exploring ensemble learning techniques, where multiple classifiers are combined to make predictions, could further enhance the classifier's performance and robustness. Furthermore, leveraging emerging technologies like deep learning and neural networks may unlock new possibilities for improving spam detection accuracy and scalability. As email spamming tactics evolve, ongoing research and development efforts will be essential to stay ahead of spammers and ensure effective spam filtering in email systems.



**8. References:**

<https://medium.com/search?q=+github+references+links>

<https://medium.com/@soumenatta>

<https://medium.com/dev-genius/detecting-spam-emails-using-logistic-regression-in-python-a-step-by-step-tutorial-29120514ac53>