INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 11, November 2024, pp: 1696-1698

SPAM CLASSIFICATION: USING NLP AND ML FOR EFFECTIVE MANAGEMENT OF UNWANTED MESSAGES

Varshith. B[1], Varshith. C[2], Varun Reddy. G[3], Veera Venkata Laxman. A[4], Veerabhadra Sai Amarnath. A[5], Prof. G. Balaiah[6]

[1,2,3,4,5]B. Tech School of Engineering Computer Science-(AI&ML) Malla Reddy University, India.

[6]Guide, Assistant Professor School of Engineering Computer Science-(AI&ML) Malla Reddy University, India.

ABSTRACT

Spam classification is a crucial task in managing communication effectively. The goal of this project is to develop a Natural Language Processing (NLP) system that can accurately classify messages as spam or not spam. Spam messages, often called junk messages, include malicious advertisements or unwanted content. These can fill up communication channels and pose security risks. In this project, we utilize various NLP techniques and machine learning algorithms to build a spam classification model. Specifically, we apply NLP methods such as TF-IDF for feature extraction, and we preprocess the text using techniques such as tokenization, lemmatization, and stopword removal. Using Machine Learning, we employ Support Vector Machines (SVM) for classification to analyze patterns in the message content. The outcomes of this project include a trained NLP-based spam classification model and a detailed performance analysis. The system can automatically filter out spam messages, making communication management more efficient and secure. This reduces the time spent on manual message sorting and enhances the overall user experience. By implementing this project, we aim to offer a practical solution to a common problem in communication using advanced NLP techniques, ensuring users receive only relevant and important messages.

INTRODUCTION

1.1 Problem Statement

In today's digital communication landscape, the sheer volume of messages sent through email and mobile messaging apps includes a significant amount of spam—unsolicited content like promotional ads, phishing attempts, and potentially. This presents several challenges:

Cluttered Inboxes: Users struggle to find genuine messages amid spam, leading to wasted time and reduced productivity.

Security Risks: Spam messages often contain malicious links that can lead to data breaches and identity theft, making effective filtering essential.

User Frustration: Current spam detection systems often misclassify messages, resulting in false positives (legitimate messages flagged as spam) and false negatives (spam messages slipping through). This can decrease user trust and expose them to threats.

To address these issues, a more effective spam classification system is needed.

1.2 Objective of Project:

The primary objective of this project is to develop a robust Natural Language Processing (NLP) system that accurately classifies messages as spam or not spam. Key components include:

NLP Techniques: Utilizing methods like Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction to prepare text data for analysis.

Machine Learning Algorithms: Implementing Support Vector Machines (SVM) for classification, leveraging its effectiveness in handling text data.

Enhancing User Experience: The goal is to improve the accuracy of spam detection, allowing users to focus on important communications without distractions.

This project aims to provide a reliable solution for enhancing security and efficiency in digital communication.

1.3 Scope & Limitations of The Project:

Scope:

Target Platforms: The focus is on spam classification for email and mobile messaging apps, addressing a widespread digital communication issue.

Integration of NLP and Machine Learning: The project will integrate advanced techniques to enhance spam detection accuracy.

Real-World Applications: The system can be adapted for use in existing messaging platforms to improve their spam

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 11, November 2024, pp: 1696-1698

detection capabilities.

Limitations:

Training Data Quality: The model's effectiveness is reliant on the diversity and quality of the training data, which may affect classification accuracy.

Dynamic Nature of Spam: As spam tactics evolve, the model may face challenges in classifying new spam types not present in the training dataset.

Real-Time Processing Constraints: The system may experience latency during peak usage times, impacting user experience.

Feature Selection Complexity: While TF-IDF is effective, additional feature selection techniques may enhance classification results further.

2. METHODOLOGY

2.1 MODEL ARCHITECTURE

The architecture of the spam classification model comprises several key components, which work together to process the input data and produce predictions.

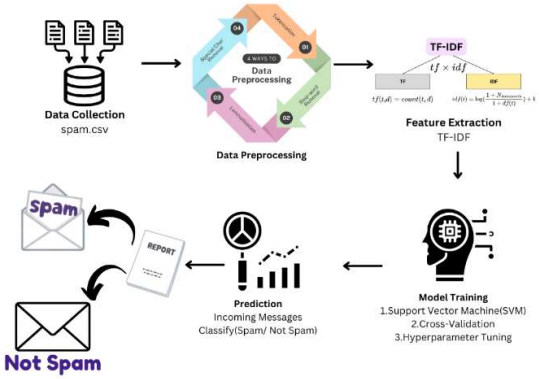

Fig 2.1: Model Architecture

Overview of Model Architecture:

Input Layer:

Text Input: Receives raw text messages for classification.

Data Preprocessing Layer:

Tokenization: Splits the input text into individual tokens (words).

Lemmatization: Reduces tokens to their base form to ensure consistency.

Stop-word Removal: Eliminates common words that do not contribute significant meaning (e.g., "and," "the").

Feature Extraction Layer:

TF-IDF Vectorization: Converts the preprocessed text data into a numerical format by calculating Term Frequency-Inverse Document Frequency (TF-IDF) scores for each term. This matrix representation captures the importance of words in the context of the entire dataset.

Classification Layer:

Support Vector Machine: The core classification algorithm that operates on the feature matrix generated from the previous layer. It constructs hyperplanes in a high-dimensional space to classify the messages as spam or not spam.

Output Layer:

Prediction Output: The model outputs the classification result, indicating whether the input message is "Spam" or "Not Spam."

Model Training Process:

The model is trained using labeled data (spam and ham messages) to learn the optimal hyperplane that separates the two classes.

Cross-validation techniques are employed to ensure robustness and prevent overfitting, while hyperparameter tuning is performed using methods like GridSearchCV to optimize the model's performance.

2.2 DESIGN

INTRODUCTION

The design phase establishes the structure and workflows for the spam classification system. This includes detailing system components, data flow, and the techniques and algorithms used. A well-structured design ensures that each component integrates seamlessly, contributing to the system's overall performance in accurately classifying spam and non-spam messages. The design includes:

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 11, November 2024, pp: 1696-1698

An overview of the chosen project diagram (DFD/UML/ER).

Details of the dataset used, including feature descriptions.

Data preprocessing techniques applied to prepare the text data.

A description of the algorithms and methods used in the classification model.

SEQUENCE DIAGRAM

The Sequence Diagram illustrates the interaction flow within the spam classification system, detailing how components communicate during the classification process. The diagram includes the following key interactions:

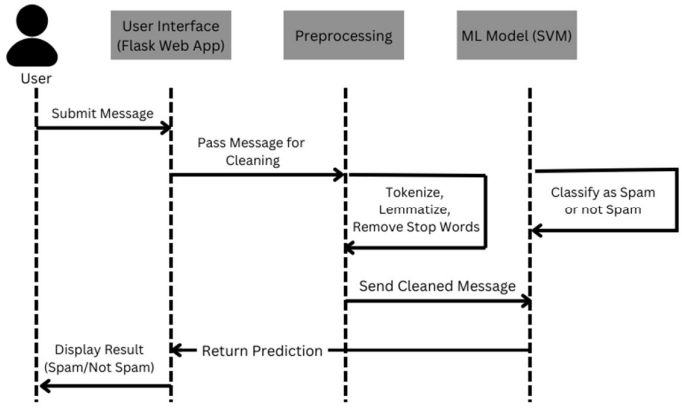

Fig 2.2: Sequence Diagram

User Input: The user initiates the process by submitting a message through the User Interface (Flask Web App).

Message Forwarding: The User Interface forwards the submitted message to the Preprocessing component for cleaning and preparation.

Preprocessing Steps: The Preprocessing component performs several tasks:

Tokenization: The message is broken down into individual words (tokens).

Lemmatization: Words are reduced to their base forms to standardize variations.

Stop-word Removal: Common words that do not contribute to meaning are removed.

Feature Extraction: After preprocessing, the cleaned message is sent to the ML Model (SVM) for classification.

Prediction Step: The ML Model processes the cleaned message, classifying it as either spam or not spam.

Return Prediction: The result of the classification is sent back to the User Interface.

Display Result: Finally, the User Interface displays the classification result to the user, indicating whether the message is spam or not spam.

2.3 DATA SET DESCRIPTIONS

The dataset consists of a collection of messages labeled as spam or ham (not spam). Each message is a labeled instance, enabling supervised learning for the classification model.

Attributes:

Label: Indicates whether the message is spam (1) or ham (0).

Message: The content of the message to be classified.

Dataset Details:

Total Records: 5,572

Classes: 0 (ham), 1 (spam)

Distribution: Approximately 13.4% spam messages and 86.6% ham messages.

2.4 DATA PREPROCESSING TECHNIQUES

Preprocessing is critical in preparing the text data for feature extraction and modeling. The techniques applied are:

Lowercasing: Converts all text to lowercase to avoid case sensitivity issues.

Tokenization: Breaks down the text into individual words, or tokens, which are analyzed separately.

Stop-word Removal: Removes common words (like "and," "is") that don't contribute to the context of the message.

Lemmatization: Reduces words to their base form, simplifying variations in word forms (e.g., "running" becomes "run").

Special Character Removal: Removes nonalphabetic characters, such as punctuation, to avoid unnecessary noise in the data.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 11, November 2024, pp: 1696-1698

2.5 METHODS & ALGORITHMS

The primary techniques and algorithms employed in the spam classification model include:

Feature Extraction (TF-IDF):

Term Frequency-Inverse Document Frequency (TF-IDF) is used to convert the processed text into numerical features. It quantifies word relevance by considering how often a term appears in a message compared to its occurrence across all messages. This provides a structured, meaningful representation of text data for model input.

Classification Algorithm (Support Vector Machine - SVM):

SVM is a powerful supervised learning algorithm, effective for binary classification tasks. The classifier finds an optimal hyperplane to separate spam and non-spam messages in the feature space. It performs well with high-dimensional data and can handle the sparse TF-IDF matrix effectively.

Hyperparameter Tuning (GridSearchCV):

GridSearchCV is used to fine-tune model parameters (like kernel type, regularization parameter C, and gamma) by testing combinations and selecting the ones that yield the best performance based on crossvalidation results.

3 RESULTS AND DISCUSSIONS

3.1 MODEL EVALUATION METRICS

Evaluation metrics assess the model's effectiveness in identifying spam messages accurately. Metrics include:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	960
1	0.98	0.91	0.94	180
accuracy			0.98	1140
macro avg	0.98	0.95	0.97	1140
weighted avg	0.98	0.98	0.98	1140

Fig 3.1.1: Evaluation Metrics

Accuracy: The percentage of correctly classified messages.

Precision: The proportion of actual spam messages among those classified as spam.

Recall: The model's ability to identify spam messages.

F1 Score: A balance of precision and recall, representing the model's overall performance.

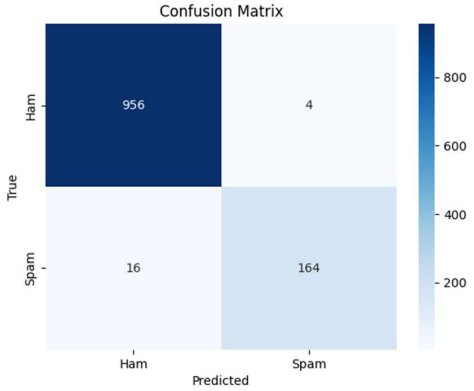

Fig 3.1.2: Confusion Matrix

Cross-Validation Scores: Highlighting consistency across multiple data subsets, reducing overfitting risks.

```
Cross-validation scores:  [0.98508772 0.98157895 0.98157895 0.98419666 0.91747147]
Mean cross-validation score:  0.969982748794726
```

3.2 RESULTS

The results section provides an analysis of the model's performance on the test set, showcasing its ability to accurately classify spam messages. Key highlights include:

Sample Predictions: Screenshots or examples of input messages with their respective classification outputs.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 11, November 2024, pp: 1696-1698

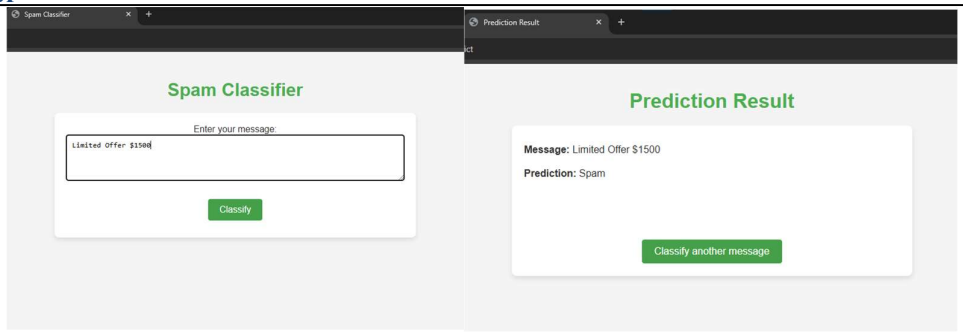

Fig 3.2.1 : Spam Message Identification

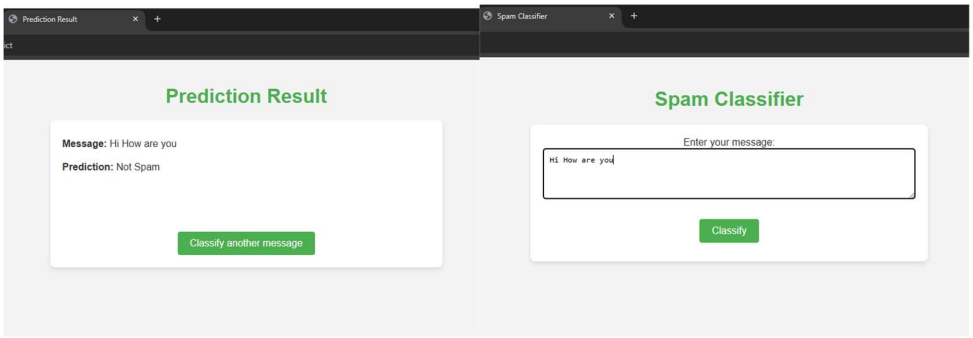

Fig 3.2.2 :Not Spam Message Identification

4. CONCLUSION

The Spam Detection System developed in this project demonstrates the potential of machine learning and Natural Language Processing (NLP) in effectively filtering spam messages. By leveraging Support Vector Machine (SVM) classification and TF-IDF vectorization, this model can reliably classify text messages as spam or non-spam with high accuracy. This system addresses the problem of unwanted spam communications, providing a scalable solution that can be easily integrated into applications and services, reducing user exposure to irrelevant and potentially harmful content. The deployed web application allows end-users to conveniently test message inputs, offering real-time spam classification.

5. FUTURE WORK

This project provides a foundation for further enhancements, such as:

- Expanding the dataset to improve accuracy across diverse spam formats and languages.
- Incorporating deep learning models, such as recurrent neural networks (RNN) or transformers, for improved performance with larger datasets.
- Developing an adaptive model that evolves with changing spam patterns, using online learning methods.
- Enhancing the application interface and deploying it as a service for broader use, potentially integrating with email or messaging platforms for real-time spam filtering.

ACKNOWLEDGEMENT

We express our sincere gratitude to Prof. G. Balaiah for his invaluable guidance, mentorship, and unwavering support throughout the development of the Spam Classification: Using NLP and ML For Effective Management of Unwanted Messages. We are also thankful to the School of Engineering at Malla Reddy University for providing the necessary resources and infrastructure for conducting this research. Additionally, we extend our appreciation to all individuals who contributed directly or indirectly to the project, for their assistance and encouragement. Their collective efforts have been instrumental in the successful completion of this endeavor.

6. REFERENCES

[1] G. James, D. Witten, T. Hastie, R. Tibshirani, "An Introduction to Statistical Learning," Springer, 2013, pp. 337-384. https://www.statlearning.com

[2] S. Raschka, "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2," Packt Publishing, 2019, pp. 219-243. https://www.packtpub.com

[3] NLTK Documentation, Natural Language Toolkit, https://www.nltk.org/

[4] Scikit-learn Documentation, scikit-learn: Machine Learning in Python, https://scikit-learn.org/

[5] Flask Documentation, Flask Web Development, https://flask.palletsprojects.com/