Spam Message Classifier Project

In the digital age, the rapid proliferation of electronic communication platforms has brought countless advantages, allowing for quick and efficient information exchange. However, it has also paved the way for a significant problem: spam. These are unwanted, often irrelevant messages, sent over the internet to a large number of users, primarily for the purposes of advertising, phishing, spreading malware, and more. With the ongoing influx of spam affecting user experience and posing security threats, the need for a robust spam detection mechanism has never been more critical.

Our project aimed to address this pressing issue by developing a machine learning model to differentiate between spam and legitimate messages. By automating the spam detection process, we hoped to protect users from potentially harmful content and improve the overall user experience by filtering out undesired communications.

1. Data Collection and Pre-processing

The foundation of any machine learning project is the dataset. We utilized a labelled dataset comprising thousands of electronic messages, each categorized as 'spam' or 'ham' (legitimate). Given the nature of text data, it was imperative to pre-process the raw data to make it suitable for model training.

The pre-processing pipeline consisted of several steps. Firstly, we tokenized the messages, splitting them into individual words or tokens. We then employed techniques to reduce dimensionality and noise:

- Lemmatization: Converting words to their base or dictionary form. For instance, 'running' would be lemmatized to 'run'. This helps in reducing the number of unique words while retaining the core meaning.

- Removing stop words: Common words such as 'and', 'the', 'is', which don't necessarily add significant meaning in the context of spam detection, were filtered out.

- Removing punctuations: Non-alphanumeric characters were eliminated, streamlining the dataset further.

2. Feature Engineering and Extraction:

With the text data cleaned, the next challenge was to convert it into a format suitable for a machine-learning algorithm. We employed the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer. This technique not only considers the frequency of a word in a particular message (Term Frequency) but also how unique the word is across all messages (Inverse Document Frequency). The result is a matrix where each row corresponds to a message, and each column represents a unique word in the dataset. The value in each cell is the TF-IDF score, effectively capturing the importance of a word in a message relative to the entire dataset.

3. Model Selection and Training:

For the classification task, we chose the Logistic Regression algorithm. Given its efficiency in binary classification problems and its ability to provide probabilities for each class, it was a fitting choice for our spam detection endeavour.

The data was split into training and test sets, with the majority being used for training. Our pipeline integrated pre-processing, vectorization, and the classifier, ensuring a smooth transformation and training process. Once trained, the model could take in a raw message and output whether it was spam or ham.

4. Results and Future Work:

The model demonstrated 96.33% accuracy on the test set, showcasing its potential for real-world applications. However, no model is perfect. As spammers continuously evolve their tactics, it's crucial to have a system that adapts and learns from new types of spam.

In conclusion, spam remains a pervasive problem in the realm of electronic communications. Through this project, we've taken a significant step towards shielding users from unwanted content. As we look ahead, continuous learning and adaptation will be the key to staying ahead of malicious actors and ensuring that our digital communication platforms remain secure and user-friendly.