PROJECT REPORT

SPAM MESSAGE DETECTION USING MACHINE LEARNING

LINK: https://github.com/priyadarshini-777/Spam-Message-Detection-using-Machine-Learning

The project focuses on the classification of SMS messages into spam or ham using machine learning techniques. Below is a comprehensive report addressing the specified questions, based on the details provided and the typical approach for such a project.

Who is Your Stakeholder?

The stakeholders are service providers and users of digital messaging platforms who are directly affected by the prevalence of spam messages. These stakeholders are interested in a solution that can accurately filter out spam to enhance user experience and security.

What is the Problem They are Trying to Solve?

The problem at hand is the classification of SMS messages into spam (unsolicited messages) or ham (legitimate messages). The goal is to develop a machine learning model that can automatically and accurately categorize incoming messages, thereby reducing the volume of spam messages that users receive.

Where Your Dataset is From?

The dataset used for this project is the "SMS Spam Collection Dataset," available on Kaggle. This dataset consists of 5,572 SMS messages in English, tagged as either 'spam' or 'ham'. The dataset can be accessed [here](https://www.kaggle.com/uciml/sms-spam-collection-dataset).

What Models Did You Try, Why Did You Choose Those Models?

1. Multinomial Naive Bayes: Chosen for its simplicity and effectiveness in text classification tasks, especially spam detection due to its assumption of feature independence.

2. Random Forest Classifier: Selected for its ability to handle non-linear data and provide higher accuracy through ensemble learning, making it robust against overfitting.

3. Support Vector Machine (SVM): Known for its effectiveness in high-dimensional spaces, making it suitable for text classification problems.

Each model was subjected to at least three hyperparameter tunings to optimize performance. For example, variations in alpha for Naive Bayes, the number of estimators in Random Forest, and the regularization parameter in SVM were explored.

What Features Did You Select/Engineer? How Did You Choose Those?

Features were engineered from the text data through:

Preprocessing: Lowercasing, tokenization, removing stop words, and punctuation, to reduce noise.

TF-IDF Vectorization: To convert text into numerical format, capturing the importance of words within documents relative to the entire corpus. This method was chosen for its ability to highlight the most relevant words for classification.

How Did You Evaluate the Model? What Evaluation Metrics Did You Use? Why?

Models were evaluated using accuracy, precision, recall, and F1 score. Precision and recall were particularly emphasized to balance the trade-off between correctly identifying spam messages and not misclassifying ham messages as spam. F1 score provided a single metric to assess the balance between precision and recall.

What Would You Do Different Next Time or Given More Time What Would Your Future Work Be?

With more time or in future work, experimenting with deep learning models like LSTM or BERT for natural language processing could potentially improve classification accuracy. Additionally, exploring more advanced feature engineering and selection methods might enhance the model's ability to differentiate between spam and ham messages.

Do You Recommend Your Client Use This Model? Is the Precision/Recall Good Enough for the Intended Use Case?

The Random Forest Classifier, with its robust performance and balanced precision and recall, is recommended for deployment. Its precision/recall metrics indicate a strong ability to identify spam accurately while minimizing false positives, making it suitable for the intended use case.

Explanation and Reasoning

The chosen models are widely recognized for their suitability in text classification tasks. Multinomial Naive Bayes offers a strong baseline with its simplicity. Random Forest provides robustness against overfitting through ensemble learning, and SVM is effective in high-dimensional spaces, like those encountered in text data. The engineering of features from the text was aimed at creating a numerical representation that captures the essential characteristics of spam and ham messages, which is crucial for any machine learning model's performance. The selected evaluation metrics provide a comprehensive view of each model's performance, emphasizing not just overall accuracy but also the model's ability to minimize false positives and false negatives.

In conclusion, the report details the systematic approach taken to solve the problem of spam detection in SMS messages, explaining the choices of models, features, and evaluation strategies based on their suitability to the task and their potential to meet the stakeholders' needs.

PRIYADARSHINI MUNIGALA