Building a Smarter AI-Powered Spam Classifier

Designing a **Web Application** To Classify Spam Messages Using TF-IDF, Multinomial Naive Bayes,and other nltk libraries.Iterative Improvement is implemented to enhance accuracy, precision, recall, and F1-score.

Data Collection: The code assumes that you have a dataset located at "./src/data/spam.csv" with two columns: "label" (containing spam or ham labels) and "message" (containing the text messages).

Data Preprocessing: The code preprocesses the text data by removing special characters, converting text to lowercase, and tokenizing the text into individual words. It also uses the NLTK library to remove stopwords and performs stemming using Porter Stemmer.

Feature Extraction: TF-IDF (Term Frequency-Inverse Document Frequency) is used to convert the tokenized words into numerical features. The maximum number of features is set to 2500.

Model Selection: The selected machine learning algorithm is Multinomial Naive Bayes, and it's trained on the TF-IDF-transformed data.

Evaluation: The code evaluates the model's performance using metrics like accuracy, precision, recall, and F1-score on a test dataset.

Iterative Improvement: When a new message is predicted, the code updates the dataset with the prediction result ("spam" or "ham") and then refreshes the model to incorporate this new data. This allows the model to learn from new examples and potentially improve over time.