Solution Strategy / Logic

Dataset Structure

The dataset is organized into two folders: one containing training emails and the other containing test emails. Spam and non-spam emails are mixed within each folder. The class label for each email is inferred directly from the file name, where spam emails begin with the prefix "spmsg" and non-spam emails follow a numeric naming format.

Text Preprocessing and Vocabulary Setup

All training emails are read and split into individual words. To improve data quality, tokens that contain non-alphabetic characters and single-letter words are removed. A vocabulary is then created by counting how often each word appears across the training set. The 3000 most frequently occurring words are retained to define the feature space.

Feature Extraction

Emails are converted into numerical vectors using word frequency counts. Each position in the vector corresponds to a word from the dictionary and stores how many times that word appears in the email. While extracting features, a label is also assigned to each email using the file name. Files that start with "spmsg" are treated as spam, and all others are treated as non-spam. The first two lines of each file are skipped because the actual message content begins after the header.

Training the Classifier

The featurized training data is used to train a Multinomial Naive Bayes classifier. This model is well suited for the problem because it is designed to work with discrete count-based features such as word frequencies.

Testing and Performance Evaluation

After training, the model is applied to the test dataset to predict whether each email is spam or not spam. Model performance is evaluated using classification accuracy, which measures the proportion of correctly classified emails.

Weaknesses of the Current Design

- The approach relies on a simple Bag-of-Words representation and does not account for word order or contextual meaning.
- Vocabulary selection is based purely on word frequency, which may include common but uninformative words.
- No additional text preprocessing steps such as lowercasing, stop-word removal, or stemming are applied.
- Model evaluation focuses only on accuracy, which may not fully reflect performance in a spam classification context.
- All words contribute equally to the model, regardless of how strongly they differentiate spam from non-spam emails.

Improved Design

The improved implementation focuses on simplifying and speeding up feature extraction. In particular, a word-to-index dictionary is created so that feature updates can be done in constant time instead of scanning the vocabulary repeatedly. A word-to-index mapping is introduced to eliminate repeated dictionary lookups during feature extraction, which significantly reduces processing time. In addition, the model makes use of scikit-learn’s optimized Multinomial Naive Bayes implementation to ensure stable and efficient training. The overall solution is modularized into clearly defined functions for vocabulary creation, feature extraction, model training, and evaluation, making the code easier to maintain and extend.