**Email Spam Classification Using SVM:**

**STEPS:**

**Email Pre-processing:**

• **Lower-casing**: The entire email is converted into lower case, so

that capitalization is ignored (e.g., IndIcaTE is treated the same as

Indicate).

• **Stripping HTML**: All HTML tags are removed from the emails.

Many emails often come with HTML formatting; we remove all the

HTML tags, so that only the content remains.

• **Normalizing URLs**: All URLs are replaced with the text “httpaddr”.

• **Normalizing Email Addresses:** All email addresses are replaced

with the text “emailaddr”.

• **Normalizing Numbers**: All numbers are replaced with the text

“number”.

• **Normalizing Dollars:** All dollar signs ($) are replaced with the text

“dollar”.

• **Word Stemming:** Words are reduced to their stemmed form. For ex-

ample, “discount”, “discounts”, “discounted” and “discounting” are all

Replaced with “discount”. Sometimes, the Stemmer actually strips o\_

additional characters from the end, so “include”, “includes”, “included”,

and “including” are all replaced with “include”.

• **Removal of non-words:** Non-words and punctuation have been re-

moved. All white spaces (tabs, newlines, spaces) have all been trimmed

to a single space character.

The Above operation is performed using the function processEmail.m function. The input given to the function is the text extracted from the file mentioned in the code. Then the above mentioned function performs all the above mentioned operations and maps the words in the pre-processed email to different words mentioned in the vocab.txt file. This file contains most of the commonly occurring words indexed from 1 to 1899 and the list of indexes of words being mapped are returned to the main function in the vector word\_indices.

After this the features vectors calls the function emailfeatures.m by passing word\_indices to it and a 1899 sizes vector called features is returned which contains 1 entries at the indexes where the word mentioned in the file vocab.txt is present in email and rest entries are 0.

These set of methods are used to create spamTest.mat and spamTrain.mat dataset which contain 4000 training data and 1000 testing data. A model is trained with help of these dataset and is tested for the email mentioned in the sample.txt file.