Topic : Spam Filter
Developer : Shweta Garg

Problem Statement: Design a spam filter to distinguish between spam and ham emails effectively.

Approach: The problem is to identify whether a given mail is spam or non-spam (ham). To manually classify e-mails, we have trained ourselves to pick certain words to mark it as useful or abuse. This is because in spam mails generally some words occur more than usual no of times. For ex: Free, 1billion$, price used to occur in lots of spam mails. Apart from this, sender data, information present in headers, etc. helps to classify the mails.

The problem can easily be translated to a binary classification problem of machine learning. From perspective of spam filtering, spam mails are considered as +ve instances and ham mails as -ve instances. There can be multiple approaches to spam filtering like supervised, unsupervised or some hybrid approach. Here, I have used supervised learning approach of spam filtering. To classify the mails, features are the tokens/words present in the mail. To find tokens I have used the following approach. First, I divide the corpus into two parts as “ham” and “spam”. I scan the full text, including headers, of each message of both parts. Then I perform some preprocessing steps, like lower casing, removing htms tags and attributes, replacing all url addresses with “httpaddr”, replacing all email addresses with “emailaddr”, replacing $ with “dollar”, replacing numbers like 10 with “number”, replacing multiple underscore with single underscore, removing = occuring because of word-wrap, etc. I remove all the stopwords occuring in the text with the help of a downloaded list of stopwords. I tokenize the text and get rid of any non-alphanumeric characters, punctuation tokens, numbers and single letters. I perform stemming of the words to get final tokens from the scanned text. Finally I prepare a pair of each token and count of the token in the mail. But all these preprocessing steps has their pros and cons in respect of false positives and false negatives which I will dicuss in the next section.

To build the model from extracted features, I have used most widely used approach of Naive Bayes for spam filtering.