ML Algorithm based Email Processing Model
Karteek Pradyumna Bulusu | kb1304@rit.edu | Fall 2016
Advised by: Prof. Carol Romanowski
Rochester Institute of Technology

Motivation
Switch from using a knowledge based engine to machine learning in various domains such as google robotics arm intrigued me to implement an email processing model using the same which is currently built using rule-based engine.

Objective
Collect the dataset pertaining to the emails (Inbox and Spam mails) and analyse the subject and title of the mails and predict the label of the mails and classify accordingly with maximum accuracy and minimal delay in receiving. [3]

Methods
• Data Collection
• Text Preprocessing
• Multinomial Naive Bayes algorithm.

Implementation
Collection of Data
• Partially scrapped inbox and spam mails from my email.
• Post scrapping, considered only Subject and body of the emails.
• Stored each mail as text file.
• Some data obtained from Enron dataset. [1]

Mail Preprocessing
• Stop word removal, removing hyperlinks, numbers, emoticons. [Python NLTK]
• Checked and corrected the spell errors. [Python Enchant]
• Tokenization. Used for calculating term frequency in later stages.

Term Frequency, Tri-gram
• Term frequency of occurrence of each word with the label is calculated. [2]
• Tri-gram is applied over each email. This considers features as individual element, as a pair and as a group of three words.
• Tri-gram helps the model understand the context of its usage. [Most Important Task]

MultinomialNB Algorithm
• This uses results of TF and Tri-gram to train the data.
• Every feature is tagged with Label after training the model.
• Tri-gram and TF of test mail is obtained and compared with trained model.
• New mail is classified based on number of inbox/spam words in mail and its weight.

Results: Testing on my emails
The Test mail is: "I(M Karteek Pradyumma) Was pleased to announce the Predictive Analytics Innovation Summit will be coming to London this March fifth (the start)."
Target label: [Inbox]
Predicted label: [Inbox]
Time to predict the label: 1.80888734204177 seconds.
Accuracy of the model is: 95.9962094294 %
Running time of Algorithm: 0.230743000000000000000000000000000000000000 seconds.

The Test mail is: "I(M Karteek Pradyumna. Still waiting in IIT Guwahati Get a CreditCard & Go Cheeky 4 Free Life Time Credit CardHolder"
Target label: [Spam]
Predicted label: [Spam]
Time to predict the label: 9.2094631114444 seconds.
Accuracy of the model is: 95.9962094294 %
Running time of Algorithm: 0.230743000000000000000000000000000000000000 seconds.

Advantages over Existing System
• Email processing requires more than thousands of rules in their rule engine, this reduces speed and requires high maintenance.
• ML is good alternative for knowledge base since it could perform same task with minimal maintenance and more optimal results.
• It is self learning and need not feed new rule for every misclassification of the mail.

Conclusion
• Gives highly accurate and quick results, which are two of the most important requirements in classifying an email.
• Machine learning algorithms in certain domains such as Email processing can replace rule-based engines.
• Though there are exceptions such as dealing with unconventional mails, with abundant trained mails and appropriate ML algorithms such as MultinomialNB and ANN, appropriate classifications can be achieved.

Future Work
• Develop a self-learning model out of this where user can mark “useful”/“not useful” for correct/incorrect predictions respectively. Correctly predicted mails will be added to the trained model automatically.
• Train with my Gmail mails and classify "promotions”, "updates” as well. [Multi-split]

References