LING131 Term Project

SMS Spam Filtering[*]

by

Bo Wen, Liu Cao, Ti Zhou[**]

December 2019

1. Introduction

With the fast advancing of email spam filters, this formerly favorite method of spam marketers has gradually give way to a new strategy of bombing innocent customers, i.e. SMS spam. Spam marketers can now utilize computers and/or third-party apps to send out thousands of spam messages at the same time, nearly free of charge. According to Wikipedia, in North America alone, mobile spam has steadily increased from 2008 ed 2012 and is projected to account for half of all mobile phone traffic in 2019. Such occurrences could not only disturb people's normal lives, but also incur unnecessary costs as people pay to receive text messages a lot of the time. We cannot, however, arbitrarily block out random or unusually-formatted numbers because that could lead to false positives, for example government emergency/alert messages, appointment reminders or authorized subscriptions. Moreover, spam marketers nowadays are clever enough to generate pseudo numbers that look just as legitimate. Thereby, proper SMS spam filtering based on contents becomes a big concern that needs to be addressed.

2. Corpus

The lack of large public dataset designated for SMS spam studies has always been an issue. The standard character limit for a single text message is 160 characters. This nature also restrained the deployment of current well-performed email spam filters which are developed upon much larger entities.

For this project, a whole dataset composed of all together four parts was used.

I. an extraction of 425 SMS spam messages from the Grumbletext website, a UK forum in which cell phone users make public claims about SMS spam messages[1]

[*] Our original plan was to focus on urban planning for cities. Ideally, by analyzing comments from online resources about commercial places, we would be able to help the government or urban planners with better decision making and resource allocation. However, when using Google API to collect raw comment data, we realized that manual labeling without a proper reference is both inefficient and ineffective. Therefore, we decided to switch to this project where proper labeling is present, hence more concrete results.

[**] Strategies and methods were discussed and implemented together by all team members.

[1] http://www.grumbletext.co.uk/

II. a subset of 3,375 randomly-chosen SMS ham messages of the NUS SMS Corpus, a dataset of about 10,000 legitimate messages collected by researchers at the Department of Computer Science of the National University of Singapore[2]

III. a collection of 450 SMS messages gathered from Caroline Tag's PhD Thesis[3]

IV. a mix of 1,002 SMS ham messages and 322 spam messages from SMS Spam Corpus v.0.1 Big[4]

This way, we were able to make use of a total of 5,574 short mobile messages, with 4,827 ham messages and 747 spam messages respectively.

3. Methods and results

A. Pre-processing

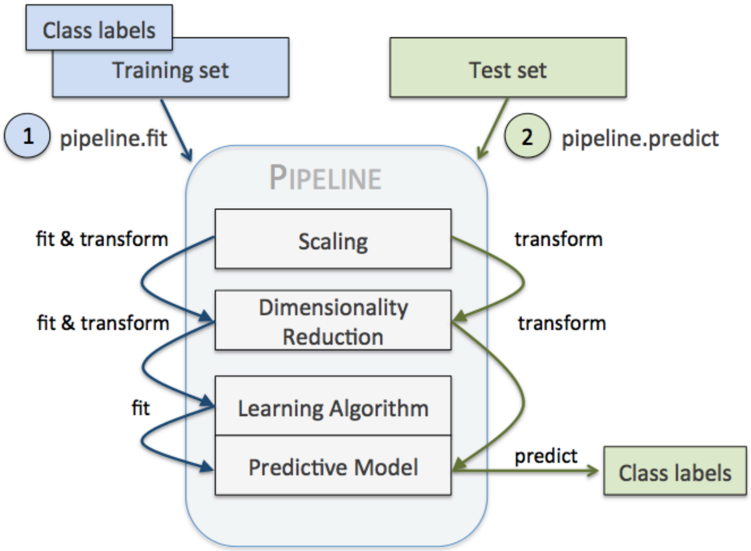
A proper amount of data cleaning and processing were performed during this step utilizing NLTK, including *stopword removal* and *word stemming*.

B. Text preparation

The special preparation needed of text data before being used for predictive modeling was conducted. With the help of features like *TfidfVectorizer* from Scikit-Learn, we were able to tokenize documents, learn the vocabulary and inverse document frequency weightings, and encode new documents.

C. Modeling

Several different approaches were tested out, out of which three most efficient models were presented, i.e. *Naïve Bayes, Decision Tree* and *SVM*. Scikit-Learn was again involved to implement the training/testing process, where a *training versus testing* set ratio of *7:3* with a *random_state = 21* was observed. *Pipeline* was introduced to bypass the issue of oversimplification.

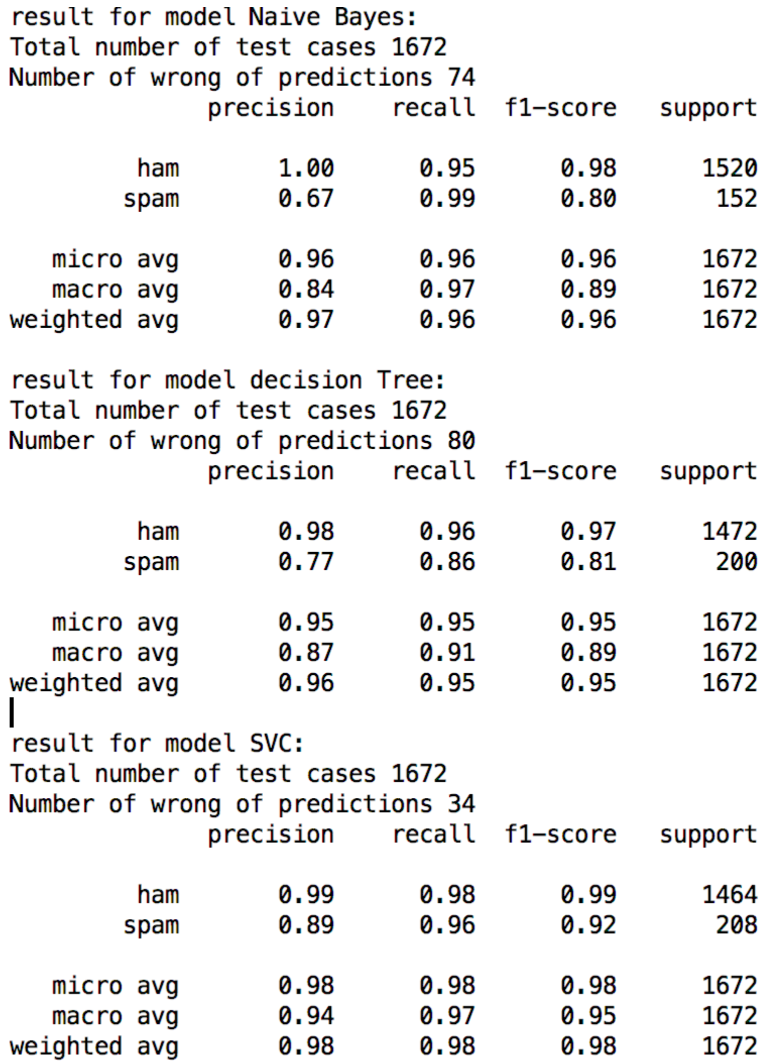


[2] http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/

[3] http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf

[4] http://www.esp.uem.es/jmgomez/smsspamcorpus/

Prediction results of all three models are shown as below. Among these, SVM is the one with the highest accuracy.



```
result for model Naive Bayes:
Total number of test cases 1672
Number of wrong of predictions 74
              precision    recall  f1-score   support

         ham       1.00      0.95      0.98      1520
        spam       0.67      0.99      0.80       152

   micro avg       0.96      0.96      0.96      1672
   macro avg       0.84      0.97      0.89      1672
weighted avg       0.97      0.96      0.96      1672

result for model decision Tree:
Total number of test cases 1672
Number of wrong of predictions 80
              precision    recall  f1-score   support

         ham       0.98      0.96      0.97      1472
        spam       0.77      0.86      0.81       200

   micro avg       0.95      0.95      0.95      1672
   macro avg       0.87      0.91      0.89      1672
weighted avg       0.96      0.95      0.95      1672

result for model SVC:
Total number of test cases 1672
Number of wrong of predictions 34
              precision    recall  f1-score   support

         ham       0.99      0.98      0.99      1464
        spam       0.89      0.96      0.92       208

   micro avg       0.98      0.98      0.98      1672
   macro avg       0.94      0.97      0.95      1672
weighted avg       0.98      0.98      0.98      1672
```

4. Future work

The simplicity as well as easy accessibility of SMS have made it attractive to malicious users like spam marketers, therefore wasting money and resources of the receivers and the network providers. With the prevailing of such trends in Asian countries, or even stateside in languages other than English, we hope to extend our study towards additional dominant languages like Spanish and Chinese. Some potential challenges might include tokenization, as words are not separated straightforwardly by blanks in languages as Chinese. We would also like to introduce BERT-Embeddings for our dataset, so as to allow for state-of-the-art results.