SMS Spam Filtering using Supervised Machine Learning Algorithms

Pavas Navaney
Student, Amity University
Noida , Uttar Pradesh
Pavasnavaney@gmail.com

Gaurav Dubey
Assistant Professor, Amity University,
Noida , Uttar Pradesh
gdubey@amity.edu

Ajay Rana
Professor, Amity University,
Noida , Uttar Pradesh
ajay_rana@amity.edu

***Abstract*— This paper presents detection of Spam and ham messages using various supervised machine learning algorithms like naïve Bayes Algorithm, support vector machines algorithm, and the maximum entropy algorithm and compares their performance in filtering the Ham and Spam messages. As people indulge more in Web-based activities, and with rising sharing of private – data by companies, SMS spam is very common. SMS spam filter inherits much functionality from E-mail Spam Filtering. Comparing the performance of various supervised learning algorithms we find the support vector machine algorithm gives us the most accurate result.**

I.INTRODUCTION

In the developing period of the Internet, individuals are involving increasingly in free online services. Individuals tend to share their data on different sites, though that data is imparted to different organizations that spam individuals to offer their services.

SMS Spamming [2] [10] in extremely disappointing for the clients: numerous critical and valuable messages can get lost because of spam messages, Spam messages are additionally used to trap individuals, or bait them into purchasing services.

As overall utilization of cell phones has grown, another road for e-junk mail has been opened for notorious advertisers. These publicists use instant messages (SMS) to target probable purchasers with undesirable publicizing known as SMS spam. This sort of spam is especially bothersome since, not at all like email spam, numerous PDA clients pay an expense for each SMS got.

Building up a classification algorithm [1] [11] that channels SMS spam would give a helpful apparatus for mobile phone suppliers.

Since naïve Bayes has been utilized effectively for email spam detection [9], it appears to be expected that it could likewise be used to build SMS spam classifier [7]. With respect to email spam [6][8], SMS spam represents extra difficulties for automated channels. SMS texts are regularly restricted to 160 characters, lessening the measure of content that can be utilized to distinguish whether a message is a ham or spam. People have also regularly started using shorthand notations and slang which further makes it difficult to distinguish between ham and spam. We will test how well a simple naïve Bayes classifier [4] manages these difficulties.

We additionally fabricate models to group messages utilizing the SVM algorithm and the maximum entropy algorithm [3], and it is discovered that SVM gives us the most precise outcomes, with exactness up to 98 %, took after by Naïve bayes algorithm, followed by maximum entropy algorithm.

Spam messages can be classified as redundant messages sent to large number of people at once. The rise of spam messages are based on the following factors:

1) The accessibility to cheap bulk SMS-plans; 2) dependability (since the message comes to the cell phone client); 3) low possibility of accepting reactions from some unaware recipients; and 4) the message can be customized.5) Free services.

II.BACKGROUND STUDY

To construct the naïve Bayes classifier [4], we will use information and data collected from the SMS Spam collection which is available openly and consists of about 5574records [5].

This dataset incorporates the content of SMS messages alongside a label signifying if the message is a ham or a spam. Junk messages are marked as spam, while true blue messages are marked as ham. A few cases of spam (Table 2) and ham (Table 1) are illustrated in the following illustration:

1. HAM MESSAGES

Draft a reasonable one. And I will see if something can happen.
Okay I can try, but cannot commit.
I am good too. Yes weekdays are busy, all thanks to office.

Table 1: Ham messages

As watched these messages are the everyday messages that individuals trade with each other, these are not junk messages and the client ought to get these messages with the spam filter not separating them through.

978-1-5386-1719-9/18/$31.00 ©2018 IEEE

2. SPAM MESSAGES

Post Diwali offer! Get 30% off + Free Cloudbar with select LED. Buy with your pre-approved loan.
Hi, good credit score makes you eligible for top loans & credit cards. Get your score in 3 minutes.
Want chocolate? Get a whole-some Chocolate Shake free on orders above Rs. 2000.

Table 2: Spam Messages

Taking a gander at the former specimen messages, we see some recognizing qualities or some repeated patterns of spam messages. One remarkable identification is that two of the three spam messages use the word "free", yet the same word (free) does not show up in any of the ham messages. Then again, two of the ham messages refer to particular days of week, at the point when contrasted with zero junk messages.
Our classifiers will exploit such examples in the word recurrence to decide if the SMS messages appear to better fit the profile of spam or ham. While it's not incomprehensible that "free" would show up outside of a spam SMS, a ham message is probably going to give extra words giving setting.

For example, a ham message may state "are you free on Saturday?", while a spam message may utilize the expression "free melodies and ringtones." The classifier will figure the likelihood of spam and ham given the confirmation gave by every one of the words in the message.

We have a total of 5574 records, out of which 4827 messages are ham and 747 messages are spam (Chart 1).

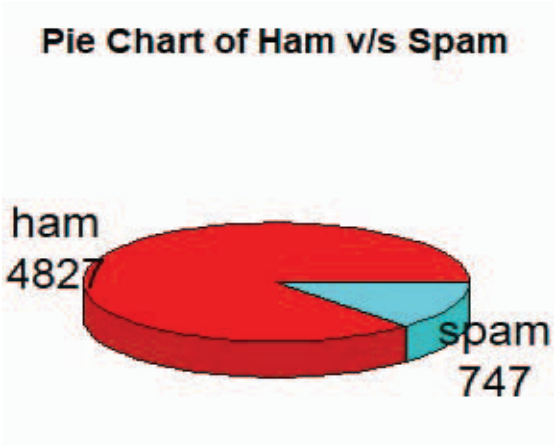
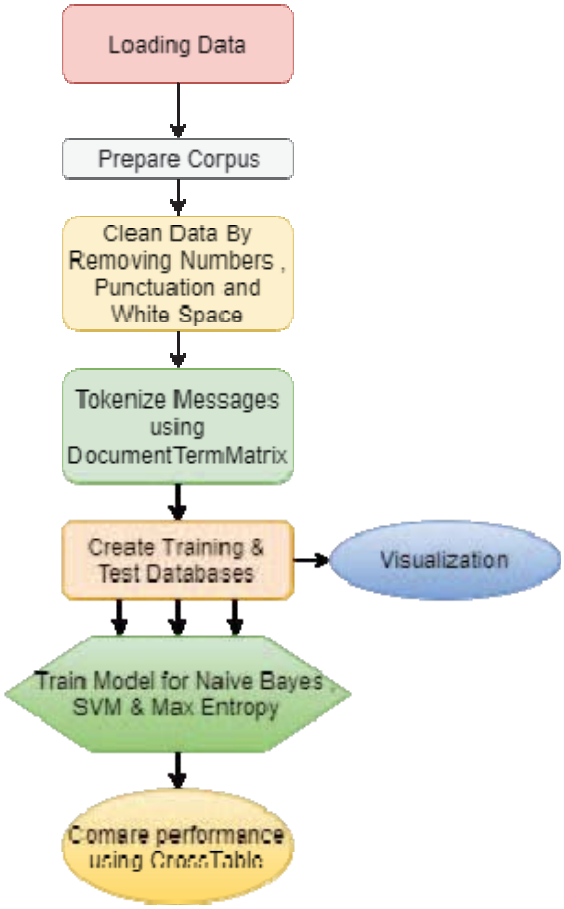


Chart 1: Ham v/s Spam

III. ARCHITECTURE OF THE CLASSIFIER



Flow-Diagram 1: Architecture of Spam Filter

As we have information in the crude shape in an excel record file, we initially import the information. We have two columns named "type" and "message". The message is the instant message while the type is the classifier of the message which is either ham or spam.
SMS messages are characters of content made out of words, punctuations, numbers, and breaks. Taking care of this kind of complex information takes a lot of attention and effort. We need to think how to evacuate punctuation, numbers, handle uninteresting words such as (and, or, but) which are called **stop words**, and how to break separated sentences into singular words. Gratefully, this utility has been given by individuals from the R group in a text mining bundle titled "tm".
The initial phase in preparing content information includes making a **corpus**, which alludes to an accumulation of text documents. For our understanding, a text document alludes to a solitary SMS message.
After removing the stop words, punctuations, numbers and blank spaces (Figure 1) we are ready to split the text messages

into single terms in the form of a data structure which is called **sparse matrix.**

Message Before Cleaning	Message After Cleaning
Hope you are having a good week. Just checking in	hope good week just checking
K...give back my thanks.	kgive back thanks
I see the letter B on my car.	See letter b car

Figure 1: Cleaning Before v/s after

Since the information is handled to our preferring, the last advance is to divide the messages into singular parts through a procedure called **tokenization**. A token is a single component of a content string; for this situation, the tokens are words.
The tokens are then represented in the form of the sparse matrix, in which each cell in the matrix contains a number indicating the count of a word that appears in a particular sentence. The sparse matrix indicates the words in the columns which the text messages are stored in the rows. The following snapshot displays a small part of the **DocumentTermMatrix**; the actual table contains 5574 rows and 7958 columns (Fig. 2).

	available	bugis	cine	crazy	got	great
1	Yes	Yes	Yes	Yes	Yes	Yes
2	No	No	No	No	No	No
3	No	No	No	No	No	No
4	No	No	No	No	No	No
5	No	No	No	No	No	No
6	No	No	No	No	No	No
7	No	No	No	No	No	No
8	No	No	No	No	No	No
9	No	No	No	No	No	No
10	No	No	No	No	No	No

Figure 2: Document Term Matrix

As we can see that many of the cells above in the table are filled with “No” which suggests that none of the above words exist in the initial ten messages of the corpus. Hence this observation is the main reason behind why this data structure is called a sparse matrix; the majority of the cells of the network are filled with “No”. Albeit each message contains a few words, the likelihood of a particular word showing up in guaranteed message is little.
The entry “yes” in the sparse matrix shows that the words available, bugis, cine, crazy, got and great are present in the first text message.

The data was then prepared by diving the dataset into **training and testing datasets, with 75% of the messages used as the training dataset and 25% was used as the testing dataset.** The training dataset consists of 4171 records and the testing dataset consists of 1403 records.

IV. Visualization using Wordclouds

WordCloud is an approach to outwardly delineate the recurrence at which words show up in information. The cloud is comprised of words scattered fairly haphazardly around the figure.
Words seeming all the more regularly in the content are appeared in a bigger text style, while less normal terms are appeared in littler textual styles. This sort of figure has developed in fame as of late since it gives an approach to watch trending activities on social networking sites.
We compare the wordclouds of ham and spam messages and see the difference between the frequently occurring terms in both the datasets.

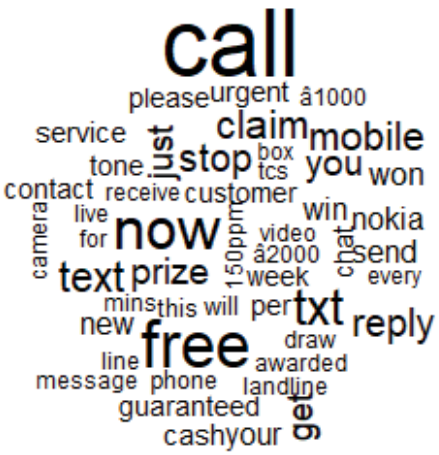


Figure 3: Wordcloud for Spam

As we observe the most frequent occurring terms in the spam messages are **call, free, text, reply, claim etc.** These are the words that we generally encounter in spam messages.

Contrasting the spam wordcloud (Fig.3) and the ham wordcloud (Fig.4) will give us a thought regarding the catchphrases that will be utilized by our classifiers in separating ham and spam. On the off chance that words present in the spam cloud likewise show up as often as possible in the ham cloud, our classifier would not have solid watchwords for correlation, while if the outcomes are distinctive, the models will have the capacity to separate amongst ham and spam well.

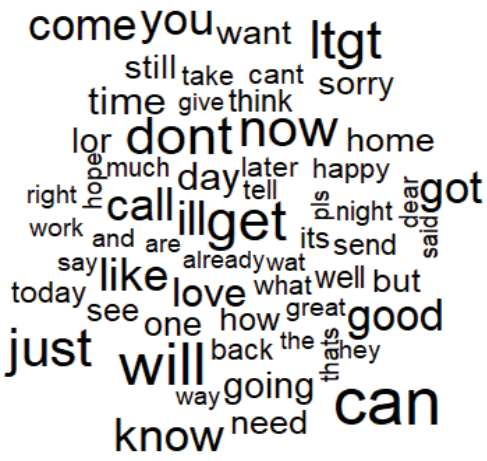


Figure 4: Wordcloud for Ham

As we observe the most frequently occurring terms are completely different from the spam wordcloud, with the words occurring in the ham wordcloud being completely different from the spam wordcloud. This difference suggests that our classifiers will have strong keywords to differentiate between ham and spam.

V. Naïve Bayes Classifier

We can characterize the issue as appeared in the accompanying formula, which catches the likelihood that a message is spam.

$$P(spam|W1\cap\sim W2\cap W3)= \frac{P(W1\cap\sim W2\cap W3|spam)P(spam)}{P(W1\cap\sim W2\cap W3)} \quad (1)$$

Suppose that there are total three words in the corpus , now if in a sentence word W1 and W3 appears but W2 does not appear , for finding the probability of spam , the naïve bayes algorithm takes the probability of word W1occuring in spam sentences. That is by dividing the total occurrences of word W1 in spam sentences divided by total occurrences of word W1 (Spam + Ham).
Similarly we can calculate for probability of ham, which will be given by the formula:

$$P(ham|W1\cap\sim W2\cap W3)= \frac{P(W1\cap\sim W2\cap W3|ham)P(ham)}{P(W1\cap\sim W2\cap W3)} \quad (2)$$

For numerous reasons this equation (Eq. II) is computationally very hard to solve. As more features are added, large amount of memory is required to store the probabilities for the large part of the possible intersections.
A large number of training data would also be needed to make sure that sufficient information exists to cover all possible associations.

Our task becomes less tedious and memory efficient if we take advantage of the fact that the naïve bayes algorithm assumes independence between the events. Naïve bayes algorithm assumes **class-conditional independence,** which means that the events are not dependent upon each other as long as they are conditioned on similar class values. That this fact into consideration allows us to simplify the above formula using the probability rule for independent events, which is given by (Eq.3):

$$P(A\cap B) = P(A)*P(B) \quad (3)$$

This result in a much simpler-to-compute equation, demonstrated below:

$$P(spam|W1\cap\sim W2)= \frac{P(W1|spam)P(\sim W2|spam)P(spam)}{P(W1)\,P(\sim W2)} \quad (4)$$

Similarly the equation for a ham message will be given by:

$$P(ham|W1\cap\sim W2)= \frac{P(W1|ham)P(\sim W2|ham)P(ham)}{P(W1)\,P(\sim W2)} \quad (5)$$

Essentially, the probability of level *L* for class *C*, given the evidence provided by features F_1 through F_n, is equal to the product of the probabilities of each piece of evidence conditioned on the class level, the prior probability of the class level, and a scaling factor *1/Z*, which converts the result to a probability:

$$P(C_L \mid F_1,\ldots,F_n) = \frac{1}{Z}\,p(C_L)\prod_{i=1}^{n} p(F_i \mid C_L) \quad (6)$$

Training the dataset with Naïve bayes model and comparing the performance on test dataset, we make the following CrossTable (Table 3).

Predicted / Actual	Ham (Messages) (Percentage)	Spam (Messages) (Percentage)	Total (Messages)
Ham	1205 98.2	22 1.7	**1227**
Spam	16 9.0	160 90.9	**176**
Total	1221 87.0	182 13.0	**1403**

Table 3: Cross Table for Naïve Bayes classifier

Therefore we can see that the naïve bayes is **98.2%** accurate in classifying a ham message and **90.9%** accurate in classifying a spam message. Therefore the naïve bayes algorithm gives an overall accuracy of **94.55%.**

V. SVM CLASSIFIER

SVMs use a linear boundary called a hyper plane to partition data into groups of similar elements, typically as indicated by the class values.
We train the model using the SVM algorithm and draw the crosstable to compare its performance.

Predicted / Actual	Ham (Messages) (Percentage)	Spam (Messages) (Percentage)	Total (Messages)
Ham	1215 98.4	20 1.6	**1235**
Spam	6 3.6	162 96.4	**168**
Total	1221 87.0	182 13.0	**1403**

Table 4: Cross Table for SVM classifier

As we observe in the crosstable, our SVM model performs better than the naïve bayes model and classifies ham with an accuracy of **98.4%** and classifies spam with **96.4%,** giving an overall accuracy **of 97.4 %(**Table 4**).**

VI. MAXIMUM ENTROPY CLASSIFIER

The principle behind Maximum Entropy is that the correct distribution is one that maximizes the Entropy or the uncertainty and still meets the constraints which are set by the 'evidence'. The mathematical formula for entropy is given by

$$H(p) = -\sum p(a,b)\log p(a,b) \qquad (7)$$

So the most likely probably distribution P is one that maximizes the entropy:

$$p = \arg\max H(p) \qquad (8)$$

We train the model using the Maximum Entropy classifier and draw the crosstable to compare its performance.

Predicted / Actual	Ham (Messages) (Percentage)	Spam (Messages) (Percentage)	Total (Messages)
Ham	1195 98.0	24 2.0	**1219**
Spam	26 14.1	158 85.9	**184**
Total	1221 87.0	182 13.0	**1403**

Table 5: Cross Table for Max. Entropy Classifier

As we observe that the maximum entropy algorithm gives us the least accuracy in classifying the messages. The maximum entropy method gives an accuracy of **98%** in classifying ham messages and **85.9%** in classifying the spam messages. The overall accuracy given by the maximum entropy method is **91.95 %(** Table 5**).**

VII. CONCLUSION

As observed using the crosstable, the SVM algorithm gives the highest accuracy in terms of classifying ham and spam messages, followed by naïve bayes method, and then Maximum Entropy method. Accuracy chart is illustrated in the below bar graph.

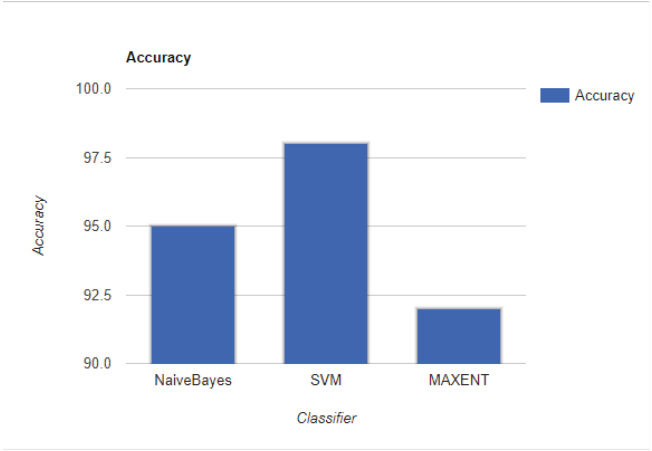


Figure 5: Comparison of Accuracy

Therefore we can safely conclude that building an SMS spam classifier using SVM algorithm gives us the best results possible with an accuracy of **97.4%.(**Fig. 5).

VIII. REFERENCES

[1]Michael Crawford,Taghi M. Khoshgoftaar,Joseph D. Prusa, Aaron N. Richter andHamzah Al Najada, "Survey of Review spam detection using machine learning techniques", Journal of Big Data 2015

[2] R Deepa Lakshmi , N. Radha , "Spam Classification using supervised learning techniques", A2CWiC'10 Proceedings of the 1st Amrita ACM-W Celebration of Women in Computing in India, Article No. 66

[3] Anju Radhakrishnan et al, "Email Classification using Machine learning algorithms", International Journal of Engineering and technology(IJET).

[4] Dea Delvia Arifin ,Shaufiah , Moch. ArifBijaksana , "Enhancing Spam Detection on mobile phone short message service(SMS) performance using FP-Growth and naïve bayes classifier" , Wireless and Mobile (APWiMob), 2016 IEEE Asia Pacific Conference(2016).

[5] J.M. Gómez Hidalgo, T.A. Almeida, andA. Yamakamim " On the Validity of a New SMS Spam Collection" , Proceedings of the 11th IEEE InternationalConference on Machine Learning and Applications, (2012.)

[6] H. Kaur , "Survey on E-mail spam detection using supervised approach with feature selection" , International Journal of Engineering Sciences and Research Technology.

[7] Rekha and S. Negi, "A Review on Different Spam Detection Approaches", International Journal of Engineering Trends and Technology (IJETT), Vol.11, No.6, 2014.

[8] A. S. Aski and N. K. Sourati, "Proposed efficient algorithm to filter spam using machine learning techniques", Pacific Science Review- A Natural Science Engineering- Elsevier, Vol. 18, No. 2, Pp.145– 149, 2016.

[9] S. P. Teli and S. K. Biradar, "Effective Email Classification for Spam and Non- spam", International Journal of Advanced Research in Computer and software Engineering, Vol. 4, 2014

[10] Shafi'l Muhammad Abdulhamid , "A Review on Mobile SMS Spam Filtering Techniques", IEEE Access, 2017.

[11] Naresh Kumar Nagwani , Aakanksha Sharaff , "SMS Spam Filtering and thread identification using bi-level text classification and clustering techniques", Journal of Information Science , 2017.