Ham and Spam Classification Using Different Features and various Machine Learning Algorithms

Aashal S Kamdar
AI Tech Systems
Link to AITS website – www.ai-techsystems.com
E-mail: aashalkamdar@gmail.com

***Abstract*— SMS spm has become quite a serious problem these days, with recipients receiving a large number of these messages everyday. This paper aims at classifying sms as ham or spam. Five classification algorithms are applied on one particular ham and spam data-set, while using different features like length of the message, number of punctuations used and the message itself to see which gives the best accuracy. Results of this research tell us that Linear Support Vector Machine algorithm, when used with the messages features, gives the best accuracy of 98.96%. This research can be useful for segregating ham and spam sms.**

Keywords—Spam; Ham; Text Classification; Machine Learning Algorithms; Feature Selection

I. INTRODUCTION

In recent years, there has been substantial growth in the mobile and smartphone industry. 1.56 billion smartphones were sold worldwide [2] in 2018 alone. Decrease in the cost of messaging service has also taken place. Short Messaging Service (which will be henceforth referred to as SMS) has become an integral part of every person's life in this modern society after the boom of mobile phones. It has developed into a multi billion dollar business Spammers have taken advantage of this fact to drive their business interests. We register our mobile number on various websites which then sell our data to these companies which spam us. Spam is often in the form of email spam, but now SMS spam also exists. Around 96% of Indians receive spam everyday [3]. The spam in SMS causes annoyance, resource consumption of the mobile phone and in some extreme cases, the receiver is even charged.

Hence, it is important that these spam messages be identified and be removed as soon as possible. The issue of SMS spam has not received as much attention from the research community as the more famous email spam.

In this paper, we present a machine learning and natural language processing based comaparative study of different features and how they affect the accuracy of detecting ham and spam. It relies on the fact that detecting spam is essentially a text classification problem [4].

II. LITERATURE REVIEW

Gordon V. Cormacket. altried to expand the execution of SMS spam discovery by applying the similar filter utilized as an email spam filters which accomplished the most elevated execution [5]. He used two data-sets, one English and the other , Spanish.

In [6] and [7] there was no stemming done on the data-set, but the data-set was split into tokens. T.Almeida introduced new data-set which comprised of 747 spam and 4827 ham messages [8]. He had the same approach as discussed in [2].

Feature extraction and selection are imperative for SMS spam detection as it will influence the exactness and execution of the classifiers. The effects of different features was examined in [9] after applying it on two data-sets, one English, and the other, Turkish. A combination between features initiated from bag-of-words (BoW) and structural features (SF) were utilized. Term Frequency – Inverse Document Frequency (TF-IDF) was used to calculate the frequency of terms and vector space model was used to represent the document as collection of words and their frequencies.

Mccallum and Nigam discussed about calculating the probability of a document by multiplying the probability of the words that occur, with the understanding that the individual word occurrences called "events" and the document to be the collection of word events (called the multinomial event model), and its application in the Naive Bayes generative model [10].

The BM algorithm, developed by Boyer and Moore in 1977, is a pattern matching algorithm that can be applied to filter SMS text on a RT-filtering system was proposed by Jun Liu et. all [11]. One important feature of BM algorithm is that algorithm achieves a high level of execution efficiency using a leap match which does not need to match each word.

In [12], the authors propose a new system where feature selection is done using stacked Restricted Boltzmann Machine (RBM), which is a greedy multilayer unsupervised deep learning technique. Deep Neural Network is used as a binary classifier which identifies whether a message is ham or spam.

III. PROPOSED METHODOLOGY

SMS spam is different from the more familiar email spam in several aspects like it does not contain a mailing list of emails, the message is less than 160 bytes and the mobile spam filtering system is to be implemented on limited resource mobile phones.

In this paper, we first obtain the data-set, perform some analysis and data normalization and normalization. Then we perform feature extraction, which is followed by application of classification algorithms.

3.1 Data Collection

The data-set was obtained from Kaggle [13] and consists of 4 columns.

	label	message	length	punct
0	ham	Go until jurong point, crazy.. Available only ...	111	9
1	ham	Ok lar... Joking wif u oni...	29	6
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155	6
3	ham	U dun say so early hor... U c already then say...	49	6
4	ham	Nah I don't think he goes to usf, he lives aro...	61	2

Figure 1. First 5 values of the data-set

1] Label – Tells if the message is ham or spam.

2] Message – Contains the text of the message sent.

3] Length – Total number of characters of the message. Spam messages will generally have higher length.

4] Punct (Punctuation) – The number of punctuations in a message.

This research paper is a comparative study. So we first choose the length and punctuation columns to predict the label which is ham or spam. Then we choose the messages column to predict whether the label is ham or spam. We apply five classification algorithms in each case and then compare the accuracy of all five algorithms and the two proposed methods.

3.2 Data Analysis and Prepartion

Figure 2 . - Length description

```
df['length'].describe()
```

```
count    5572.000000
mean       80.489950
std        59.942907
min         2.000000
25%        36.000000
50%        62.000000
75%       122.000000
max       910.000000
Name: length, dtype: float64
```

This dataset is extremely skewed. The mean value is 80.5 and yet the max length is 910.

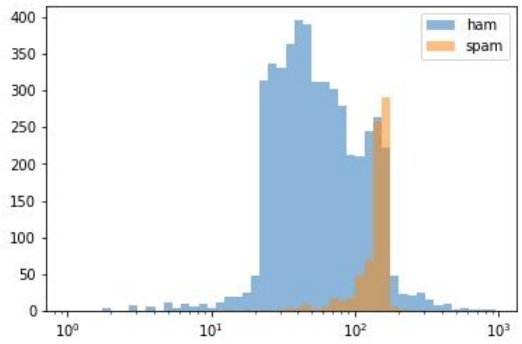

Figure 3. - Plotting length on logarithmic scale

There is a small range of values of length where a message is more likely to be spam than ham.

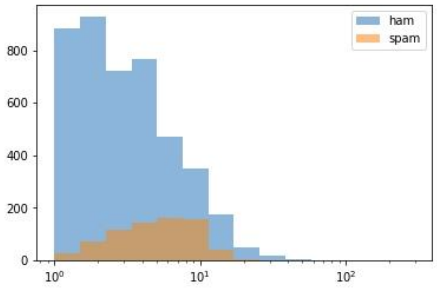

Figure 4. - Plotting Punctuation on logarithmic scale

This looks even worse - there seem to be no values where one would pick spam over ham.

By analyzing these two graphs we can get a rough idea that the results using length and punctuation will not be promising.

To prepare the text data, we perform Count Vectorization and Term Frequency Inverse Document Frequency (which shall be henceforth referred to as TF-IDF).

Count Vectorization [15] – It involved counting the number of occurences of each word that appears in a document. It tokenizes the document and builds a vocabulary.

TF-IDF [15] – An issue with simple counts is that words like 'the' will appear many times but they won't be meaningful in the encoded vectors. Therefore, we use TF-IDF.

- Term Frequency: This summarizes how often a given word appears within a document.

- Inverse Document Frequency: This downscales words that appear a lot across documents.

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Figure 5. - TF-IDF Formula

3.3 Classification Algorithms

In this paper we will make use of 5 different classification algorithms to classify a message as ham or spam.

1] Support Vector Machine (SVM):

This algorithm implements the following idea – input vectors are linearly mapped to a very high-dimension feature space [13], where a decision surface is constructed. Special properties of this decision surface ensures high generalization ability of the learning machine.

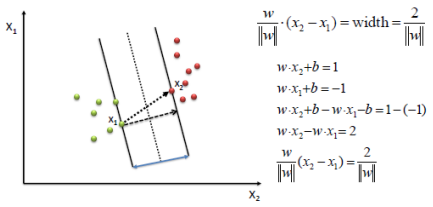

Figure 6.- SVM Graph Representation

2] Logistic Regression:

Introduced in 1958 [16], it is one of the earliest methods invented to perform classification. Logistic regression measures the relationship between the categorical dependant variable and one or more independent variables by estimating probabilities using a sigmoid curve.

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

Figure 7. - Sigmoid Curve Equation

3] Decision Tree:

This algorithm uses a tree-like graph or model of decisions and their possible consequences. It has a flow-like structure in which each internal node represents a “test” on an attribute [17].

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Figure 8.- - Decision Tree Examples

4] Random Forest:

This algorithm is an extension of the Decision Tree algorithm [18]. It works by creating more than one decision tree which is equivalent to a ‘forest’. Just like more trees in the forest, more robust the forest looks,higher the number of decision trees in a forest, higher the accuracy.

$$K_k^{cc}(\mathbf{x},\mathbf{z}) = \sum_{k_1,\ldots,k_d,\sum_{j=1}^{d} k_j=k} \frac{k!}{k_1!\cdots k_d!}\left(\frac{1}{d}\right)^k \prod_{j=1}^{d} \mathbf{1}_{\lceil 2^{k_j} x_j\rceil=\lceil 2^{k_j} z_j\rceil},$$
$$\text{for all } \mathbf{x},\mathbf{z} \in [0,1]^d.$$

Figure 9. - Random Forest Formula

5] Naive Bayes:

It is a group of linear classifiers that are simple and efficient. The probabilistic model of this algorithm is based on the Bayes Theorem, and the adjective model comes from the fact that the features in a data-set are mutually independent. [19]. There are 3 types of Naive Bayes [20]-

- Gaussian: It assumes features follow normal distribution
- Multinomial: It is used for discrete counts.
- Binomial: It is useful if the feature vectors are binary.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Figure 10. - Bayes Theorem

The data-set is split into training set (77%) and testing set (33%) for all cases. It has a random state value of 42.

IV. Result analysis

The proposed work is developed on Jupyter Notebooks running on the browser on Ubuntu 18.04. The algorithms have been implemented using Scikit Learn. The configurations of the machine on which this experiment was conducted is, 4 GB Ram, Intel Core i5 7th Generation, 2.50 GHz and 64-bit OS type.

We use two metrics for evaluation, namely the accuracy and F1 score.

1] Accuracy : Ratio of correctly predicted observation to the total observations.

2] F1 Score : F1 Score is the weighted average of Precision and Recall.

Here precision is the ratio of correctly predicted positive observations to the total predicted positive observations, and recall is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\begin{aligned} precision &= \frac{TP}{TP + FP} \\ recall &= \frac{TP}{TP + FN} \\ F1 &= \frac{2 \times precision \times recall}{precision + recall} \\ accuracy &= \frac{TP + TN}{TP + FN + TN + FP} \end{aligned}$$

Figure 11. - Evaluation Metrics

In the Figure 11 above -

TP – True Positive

TN – True Negative

FP – False Positive

FN – False Negative

The Table 1 below depicts the accuracy and Spam F1 Score when punctuation and length are chosen as features. As predicted in the earlier section of Data analysis and preparation, we find that the results obtained are not quite good. There are 4825 ham messages out of a total 5527, which is approximately 87.3% . This means that any machine learning model we create has to perform better than 86.6% in order to beat random chance. In this method, SVM, Decision Tree and Random Forest manage to beat this random chance, but by a very small margin.

Algorithm	Accuracy	Spam F1 Score
Support Vector Machine	88.63%	52%
Logistic Regression	84.39%	3%
Decision Tree	87.81%	49%
Random Forest	88.52%	53%
Multinomial Naive Bayes	86.07%	0%

Table 1

The Table 2 depicted below shows the accuracy and Spam F1 Score when messages is chosen as the feature. We can see that there is a significant improvement in the accuracy as well as the Spam F1 score.

Algorithm	Accuracy	Spam F1 Score
Support Vector Machine	98.96%	96%
Logistic Regression	97.17%	88%
Decision Tree	96.35%	86%
Random Forest	97.06%	88%
Multinomial Naive Bayes	96.19%	83%

Table 2

V. CONCLUSION

From the results of this research, it is found that SVM, particularly Linear Support Vector Classifier (Linear SVC), is the best performing algorithm with an accuracy of 98.96% and Spam F1 Score of 96%. This simple scheme can be implemented into an application on a mobile or smart phone. Hence a spam filter approach which is lightweight and simple and has an accuracy of 98.96% is introduced in this paper. In extension of this work, various other classification algorithms like K-Nearest Neighbours, Boosted Trees and other types of Naive Bayes. Artificial or Deep Neural Networks can also be used for classification purposes. Special types of neural networks like

Recurrent Neural Networks and Long Short Term Memory have also shown good results [21].

REFERENCES

[1] Press Release, Growth Accelerates in the Worldwide Mobile Phone and Smartphone Markets in the Second Quarter, According to IDC

[2] Number of smartphones sold to end users worldwide from 2007 to 2020

[3] https://qz.com/india/1573148/telecom-realty-firms-banks-send-most-sms-spam-in-india/

[4] Rick L. Allison, & Peter J.Morrison, US Patent Document – 6S19932 Methods and systems for preventing delivery of unwanted short message service (SMS) messages, (Nov 2004)

[5] G. V. Cormack, J. M. G. Hidalgo, and E. P. Sánz, "Feature engineering for mobile (SMS) spam filtering," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ˝07, 2007

[6] Shirani-Mehr, H., "SMS spam detection using machine learning approach," Stanford University, USA, pp. 1–4, 2013

[7] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering," in Proceedings of the 11th ACM symposium on Document engineering - DocEng ˝11, 2011.

[8] T. Almeida, J. M. G. Hidalgo, and T. P. Silva, "Towards sms spam filtering: Results under a new dataset.," International Journal of Information Security Science, 2013.

[9] A. K. Uysal, S. Gunal, S. Ergin, and E. SoraGunal, "The Impact of Feature Extraction and Selection on SMS Spam Filtering," Electronics and Electrical Engineering, vol. 19, no. 5, May 2013.

[10] Mccallum, A., & Nigam, K. (1998). "A comparison of event models for naive Bayes text classification". AAAI-98 Workshop on 'Learning for Text Categorization'

[11] Liu, J., Ke, H., & Zhang, G. (2010). "Real-time sms filtering system based on bm algorithm". System, 6-8.

[12] M. Nivaashini, R.S.Soundarya, A.Kodieswari, P.Thangaraj. "SMS Spam Detection using Deep Neural Network" in International Journal of Pure and Applied Mathematics, Volume 119, No 18.

[13] https://www.kaggle.com/uciml/sms-spam-collection-dataset

[14] Corinna Cortes, Vladimir Vapnik. "Support Vector-Networks" at AT&T Labs, USA

[15] https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

[16] D.R.Cox, "The Regression Analysis of Binary Sequences", Journal of the Royal Statistical Society. Series B (Methodological), Vol. 20, No. 2 (1958), pp. 215-242

[17] JR Quinlan, "Induction of Decision Trees"

[18] Tin Kam Ho, "Random Decision Forests" , ICDAR '95 Proceedings of the Third International Conference on Document Analysis and Recognition (Volume1),Page278

[19] https://sebastianraschka.com/Articles/2014_naive_bayes_1.html

[20] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[21] https://mc.ai/spam-detection-using-rnn-simplernn-lstm-with-step-by-step-explanation/