ISSN (e): 2250 – 3005 || Volume, 08 || Issue, 6|| Jun – 2018 ||
International Journal of Computational Engineering Research (IJCER)

Spam Filter using Naïve Bayesian Technique

Aditya Gupta[1], Khatri Mrunal Mohan[2], Sushila Shidnal[3]
[1]Sir MVIT, Bangalore [2]Sir MVIT, Bangalore
[3] Assistant Professor, Sir MVIT, Bangalore
Corresponding Auther: Aditya Gupta

ABSTRACT
The investigation of performance of Naïve Bayesian machine learning algorithm in the context of antispamfiltering is done here. The increasing volume of unsolicited bulk e-mail (spam) has generated a needfor reliable anti-spam filters. Filters of this type have so far been based mostly on keywordpatterns that are constructed by hand and perform poorly. The Naive Bayesian classifier hasrecently been suggested as an effective method to construct automatically anti-spam filterswith superior performance. The investigation of the performance of the Naïve Bayesian filter is done on a publicly available corpus, a dataset from Kaggle, contributing towards standard benchmarks. At the same time, performance analysis of the Naive Bayesian filter has been carried out. This Method achieves95.56%accuracy and 93.91% precision spam filtering for the considered dataset, outperforming the keyword-based filterof a widely used e-mail reader.

Date of Submission: 15-06-2018 Date of acceptance: 30-06-2018

I. INTRODUCTION

In recent years, with the internet becoming an integral part of our life, leading to substantial increased use of internet, numbers of emailusers are increasing day by day. Estimations lead to close to 294 billion emails beingexchanged every day. This excessively increased use ofemail has created problems caused by unsolicited bulk email messages commonly referred to as Spam. Spam messages are typically sent using bulk-mailers and address lists harvested from web pages and newsgroup archives. The difference is quite significant from vacation advertisements to get-rich schemes. These messages feature content that is usually of little interest to the majority of the recipients. In some cases, they may even be harmful, e.g. spam messages may contain clickbaits and virus Trojans. Apartfrom wasting time and bandwidth, spam e-mail also costs penny to users with dial-up connections.It isassumed that spam or viruses pile up around 90% of emails sent every day. Thesituation seems to beworsening, as without appropriate counter-measures, spam messagescould eventually undermine the usability of e-mail exchange.Attempts to introduce legal measures against spam mailing have had minimal effect. An effectivesolution is to develop tools to help recipients identify or remove automaticallyspam messages. Such tools, referred to asanti-spam filters, differ in functionality from blacklists of frequent spammers to content-based filters. The former are generally more powerful, as spammers opt to use fake addresses. Existing content-based filters tend to search for particularkeyword patterns in the messages. These patterns need to be hand crafted, and to achievebetter results they need to be tuned to each user and to be constantly maintained, a tedious task, which may require expertise that a user may not have.To address this issue of anti-spam filtering, machine learning comes in handy. The supervised learning methods have been examined, which learn to identify spam e-mail after receiving training on messages that have been manually classified as spam or non-spam.

A spam filter is a program that is mainlyemployed to detect unsolicited and unwanted email and prevent those messages from reaching a user's inbox. Just like other types of filtering programs, a spam filter looks for certain criteria on which it bases its judgments. Consider one of the simplest and earliest versions (such as the one available with Microsoft's Hotmail) can be set to watch for particular words in the subject line of messages and to exclude these from the user's inbox. This method was and is not especially effective; it may omit legitimate messages (called false positives) and passing actual spam messages. More advancedsophisticated programs such as Bayesian filters or other heuristic filters, aim at identifying spam through suspicious word patterns or word frequency.

The familiar Bayesian approach is being used in this paper. A dataset from Kaggle which contains 5572test casesof spam and ham messages sent via email is used here along with various python libraries, namely Numpy, NLTK, WordCloud, Panda and Matplotlib to help in filtering out the emails and visualisation of the frequently used keywords. Naïve Bayesian Machine Learning algorithm is based on the simple yet powerful probability theorem called Bayes Theoremas stated in formula 1.1

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A).P(A)}}{\mathrm{P(B)}}, \qquad 1.1$$

Where A and B are events and P(B) ≠ 0.
P(A) and P(B) are probabilities of observing A and B without regard to each other.
P(A|B), a conditional probability, is the probability of observing event A given that B is true.
P(B|A) is the probability of observing event B given that A is true.

Message m = (w1, w2, ... , wn), where (w1, w2, ... , wn) is a set of unique words contained in the message is used. We need to find P(spam|w1…) as stated in formula 1.2

$$\mathrm{P(spam|w1 \cap w2 \cap \ldots \cap wn)} = \frac{\mathrm{P(w1 \cap w2 \cap \ldots \cap wn\,|spam\,).P(spam\,)}}{\mathrm{P(w1 \cap w2 \cap \ldots \cap wn)}}, \qquad 1.2$$

Assuming that occurrence of a word is independent of all other words, it can be simplified to the expression 1.3,

$$\frac{\mathrm{P(w1|spam\,).P(w2|spam\,)\ldots P(wn\,|spam\,).P(spam\,)}}{\mathrm{P(w1).P(w2)\ldots P(wn)}}, \qquad 1.3$$

In order to classify, determine which is greater

$$\mathrm{P(spam|w1 \cap w2 \cap \ldots \cap wn)\ versus\ P(\sim spam|w1 \cap w2 \cap \ldots \cap wn)} \qquad 1.4$$

Whichever probability among P(spam|message) and P(ham|message) is greater in 1.4, the corresponding tag (spam or ham) is assigned to the input message.
The paper is organized infour chapters. First chapter presents related works on Spam Filter Application, second chapter presents the principles of Naïve Bayesian approach and detail explanation of the implementation, third chapter gives experimental results and evaluates precision, recall, F-score, accuracy values. Fourth chapter concludes this work and discuss further scope of this work.

II. LITERATURE SURVEY

Chae et al. identified that 100% accuracy in spam classification of email system is still an unmet need. Project has drawn upon the work of the existing email classification systems known as ‘context-based email classification system’ and ‘Linger’ to address the unmet need. Main steps of the context-based email classification system begins with pre-processing email using POS Tagger then it extracts several email features to transform emails into graphs and then graphs are matched to representative graph so that emails are classified to the folder which the representative graph with highest match represent . Linger implements information gain classifier for filtering spam and use neural network to classify emails into homogenous clusters. The proposed system adopts spam filter from Linger to reinforce the accuracy needed to separate spam emails without any mistake. Alurkar et al categorised emails in two categories, namely spam and non-spam. This has a myriad of implications for both organisations and individual users. At an organisational level, an effective and flexible classifier improves the soundness of its employees’ email systems. For an individual user, a secure email client who automatically blocks spam emails is absolutely essential. A self-learning system which is customizable to each user and based on their dataset will only ensure greater accuracy as the dataset grows in size. Thus the system approaches an optimal solution as time passes. Neelavathi et al. analysed the six selected classification algorithms based on Weak and various spam filtering techniques. The result showed the best classifier algorithm is Random Tree classifier for UCI Spambase dataset and performance of each of these six(JRip, Filtered classifier, K-Star, SGD, Multinomial, Random Tree) algorithms can be improved if the dataset is pre-processed using Partition Membership Filter. Among the spam filtering techniques random Tree generates the best spam mail filtering results in terms of more accuracy and less false positive rate.Feng et al. proposed an SVM-NB system to achieve effective and efficient spam email filtering. SVM-NB aims at removing the assumption of independence among features extracted from training set, when the NB algorithm is applied. The solution leverages SVM technique to divide training samples into different categories and identifies dependent training samples. Removing those samples results in a more independent training set with few overlapping features. Kumar et al. proposed a Dual-Margin Multi-Class Hypersphere Support Vector Machine (DMMH-SVM) classifier model and introduced cloaking-based novel features for web spam classification. The proposed classifier classifies Web pages into four categories, i.e., content spam, link spam, cloaking spam, and combined spam. The experiments on WEBSPAM-UK’07, CLUE WEB’09, and ECML/PKDD’10 datasets show that our

novel classifier model is very effective categorizing Web spam, and achieves higher accuracy, precision and recall than the state-of-art approaches. Roy et al. incorporated LCS logic to identify any word whether it is spam or ham even if it is misspelled. The proposed model enhanced the existing models in significant amount. Almeida et al. presented a spam classifier based on the Minimum Description Length principle and compare its performance with seven different models of the well-known NB classifiers and the linear SVM.Vipin et al .proposed a distributed on-line SPAM filtering scheme for encrypted messages. They used keyword based filtering. It uses Merkle-Hellman encoding to securely and compactly send the keyword status from client to filter. They further enhance this scheme by providing a parameter based clustering scheme.
Rekha et al.have categorized different approaches to spam detection as Whitelist/Blacklist, Bayesiananalysis, Mail header analysis, Keyword checking etc. and compared them on the basis of their advantages and disadvantages [19].Mohammed et al. introduced an approach for spam filtering that starts by generating a spam-hamlexicon from a given training data and uses this lexicon to filter the training and testing tables that can be used by variety of data mining algorithms. Using Python they demonstrated that it is a powerful language that can be used for emails text mining as it have very rich natural language and data mining packages.They worked on Nielson Email-1431 dataset and found that the most effective spam classifier approach is the Naïve Bayes approach.Mali et alpresented an effective technique to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. Using Bayesian filtering algorithm and effective pattern Discovery technique we can detect the spam mails from the email dataset with good correctness of term.Ann Nosseir et al. used a multi-neural networks classifier to identify bad and good words in the textual content of an email. Words in the message are reprocessed before using the multi-neural networks classifier. The word goes through stop words and noise removal steps then stemming process step to extract the word root or stem. The experiment shows positive results.Lin et al. proposed the Bro intrusion detectionsystem to monitor the SMTP sessions in a university campus, andtrack the number and the uniqueness of the recipients' emailaddresses in the outgoing mail messages from each individualinternal host as the features for detecting spamming bots. Dueto the huge number of email addresses observed in the SMTPsessions, we store and manage them efficiently in the Bloomfilters.Po-Ching et al. presented a method to detect spamming bots on the sender side. The detection features based on the number and uniqueness of REAs are simple yet effective. We monitored the SMTP sessions initiated from a large campus network for six months, and analysed the SMTP logs by tracking the features with Bloom filters to detect the internal spamming bots.Chiou et al. presented a method to build an enhanced grey list and a local RBL based on the analysis of the client behaviour to effectively block spam sessions in time, instead of relying on collecting spam messages on the spam trap or reports from users. This method can efficiently block around 70% of spam sessions with the false-positive rate lower than 0.01%. The performance was verified with the real-world mail traffic.

III. METHODOLOGY

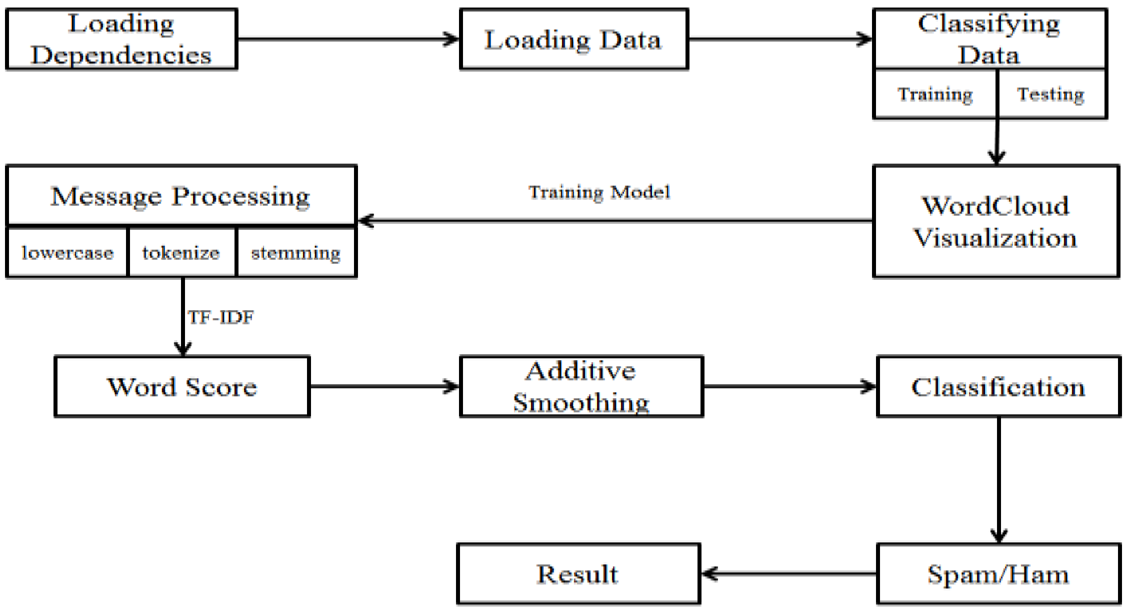


Fig.1. - Block Diagram of the Functioning of the Proposed Spam Filter

From the block diagram shown in figure 1, it can be seen that the first step in this Spam classifier is to load the required dependencies. Various python libraries have been used here.NLTK for processing the messages, WordCloud and matplotlib for visualization and pandas for loading data, NumPy for generating random probabilities for train-test split.Next data needs to be loaded from the excel file which consists of dataset having 5572test cases of Spam and Ham texts.In order to test the model, data should be split into train dataset and test dataset. The train dataset is used to train the model and then it will be tested on the test dataset, 75% of the dataset as train dataset and the rest as test dataset. Selection of this 75% of the data is uniformly random.Next, the most repeated words in the spam messages have been observed and WordCloud library is used for the purpose of visualization along with matplotlib. The output of will be plotted as an image using Matplotlib. Two separate images will be formed for both Spam and Ham messages respectively. Before starting with training, the messages need to be pre-processed. First of all, characters are convertedto lowercase,then each message in the dataset is tokenized. Tokenization is the task of splitting up a message into pieces and throwing away the punctuation characters.Next is Stemming, Porter Stemmer algorithm is used for stemming.Lastly, stop wordsare removed. These words do not give any information about the content of the text,thus it should not matter if these words are removed from the text.Next, the model is trained. To do so, two techniques have been implemented: Bag of words and TF-IDF. In Bag of words model, the 'term frequency'is found using formula 3.1 and 3.2 respectively, i.e. number of occurrences of each word in the dataset. Thus for word w,

$$\mathrm{P(w)} = \frac{\text{Total number of occurences of w in dataset}}{\text{Total number of words in the dataset}}, \qquad 3.1$$

and

$$\mathrm{P(w|spam)} = \frac{\text{Total number of occurences of w in spam messages}}{\text{Total number of words in the spam messages}}, \qquad 3.2$$

TF-IDF stands for Term Frequency-Inverse Document Frequency. In addition to Term Frequency, Inverse document frequency is computed,as stated in formula 3.3

$$\mathrm{IDF(w)} = \log\frac{\text{Total number of messages}}{\text{Total number of messages containing w}}, \qquad 3.3$$

In this model each word has a score, which is TF(w)*IDF(w). Probability of each word is counted, as shown in formula 3.4&3.5

$$\mathrm{P(w)} = \frac{\mathrm{TF(w)*IDF(w)}}{\sum_{\forall \text{words } x \in \text{train dataset}} \mathrm{TF(w)*IDF(w)}}, \qquad 3.4$$

and

$$\mathrm{P}(w|spam) = \frac{TF(w|spam)*IDF(w)}{\sum_{\forall words\ x \in train\ dataset} TF(x|spam)*IDF(x)}, \qquad 3.5$$

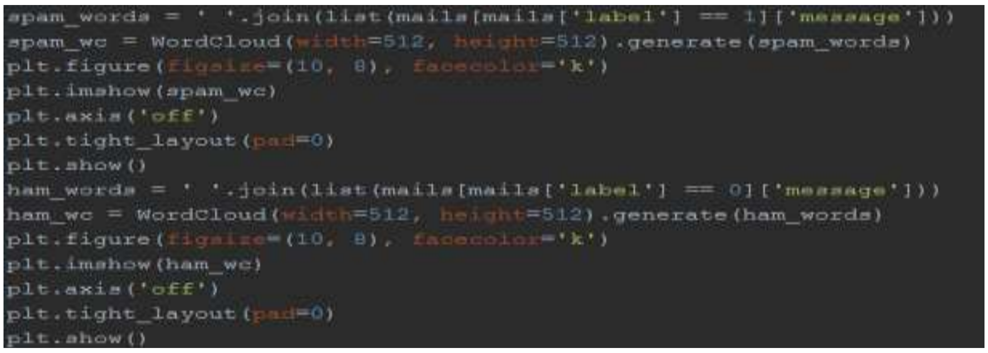
In additive smoothing,a number alpha is added to the numerator and alpha times the number of classes over which the probability is found in the denominator is added,as shown in formula 3.6. This is done so that the least probability of any word now should be a finite number. Addition in the denominator is to make the resultant sum of all the probabilities of words in the spam emails as 1.

$$P(w|spam) = \frac{TF(w|spam)*IDF(w)+\alpha}{\sum_{\forall words\ x \in train\ dataset} TF(x)*IDF(x)+\alpha\sum_{\forall words\ x \in spam\ in\ train\ dataset}}, \qquad 3.6$$

For classifying a given message, pre-processing it is done. Then, for each word w in the processed messaged, product of P(w|spam) is computed. If w does not exist in the train dataset,consider TF(w) as 0 and compute P(w|spam) using above formula, then multiply this product with P(spam). The resultant product is the P(spam|message). Similarly, P(ham|message) is computed. Whichever probability among P(spam|message) and (ham|message) is greater, the corresponding tag (spam or ham) is assigned to the input message.Finally, the output stating whether the given message is a Spam message or Ham message is obtained. In case of Spam message, "True" is printed while for Ham message, "False" is printed. Also the precision value, recall rate, F-score and accuracy of the dataset will be printed.

IV. RESULT AND DISCUSSION

Since WordCloud and matplotlib are used for plotting visualizations, as shown in figure 2, following visualizationsfigure 3 and 4 for Spam and Ham respectively are plotted, which are the most frequent words in spam messages and ham messages.



```
spam_words = ' '.join(list(mails[mails['label'] == 1]['message']))
spam_wc = WordCloud(width=512, height=512).generate(spam_words)
plt.figure(figsize=(10, 8), facecolor='k')
plt.imshow(spam_wc)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
ham_words = ' '.join(list(mails[mails['label'] == 0]['message']))
ham_wc = WordCloud(width=512, height=512).generate(ham_words)
plt.figure(figsize=(10, 8), facecolor='k')
plt.imshow(ham_wc)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

Fig. 2. – Plotting Spam and Ham Visualizations

This results in the following

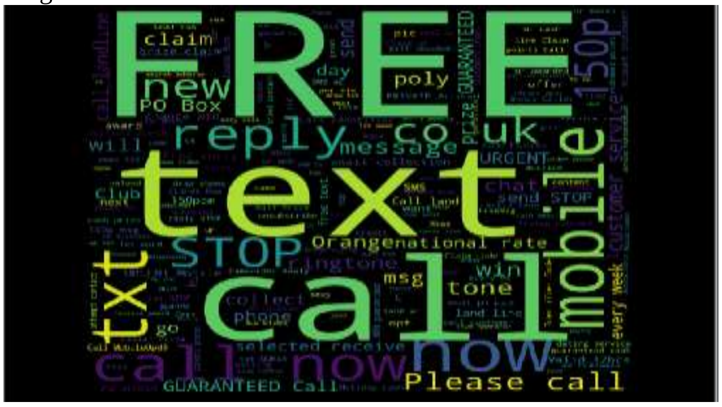


Fig.3. – Spam Visualization

As expected, these messages mostly contain the words like ‘FREE’, ‘call’, ‘text’, ‘ringtone’, ‘prize claim’ etc.Similarly the visualization of ham messages is as follows:

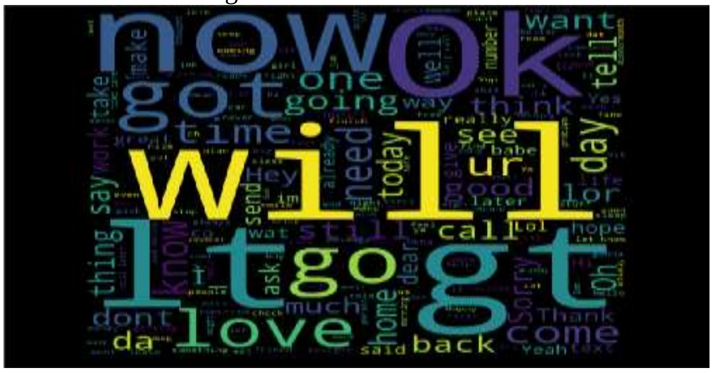
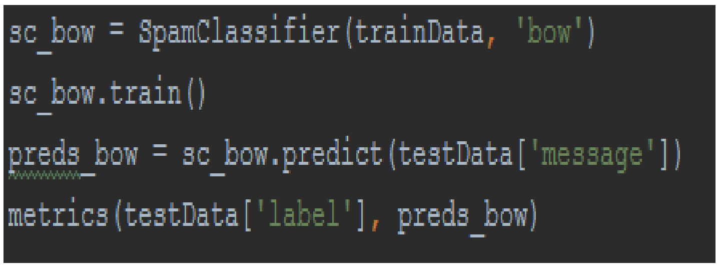


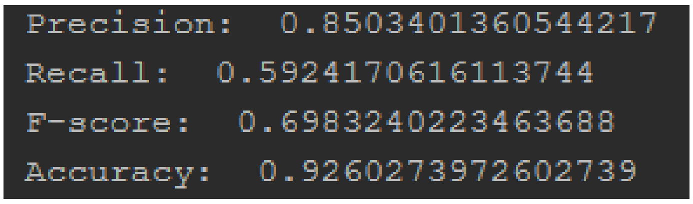
Fig.4. – Ham Visualization

The output will tell whether the given message is a Spam message or Ham message. In case of Spam message "True" is printed while for Ham message "False" is printed. The precision value, recall rate, F-score and accuracy of the dataset will be displayed, as depicted infigure 5& 6 for BOW and figure 7 – 10 for TF-IDF,



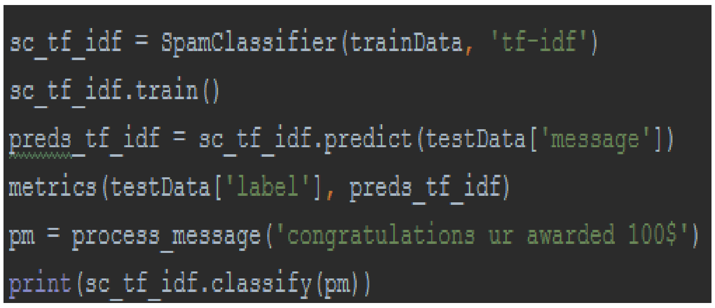
```
sc_bow = SpamClassifier(trainData, 'bow')
sc_bow.train()
preds_bow = sc_bow.predict(testData['message'])
metrics(testData['label'], preds_bow)
```

Fig. 5. – BOW



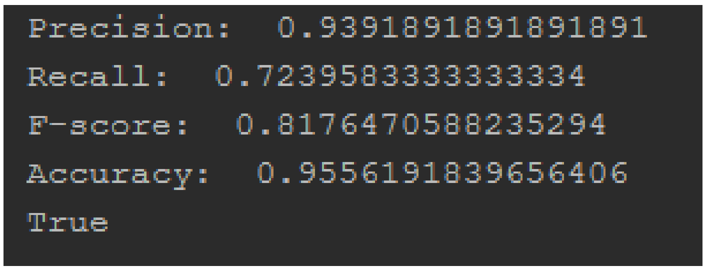
```
Precision:  0.8503401360544217
Recall:  0.5924170616113744
F-score:  0.6983240223463688
Accuracy:  0.9260273972602739
```

Fig.6. – Results for BOW



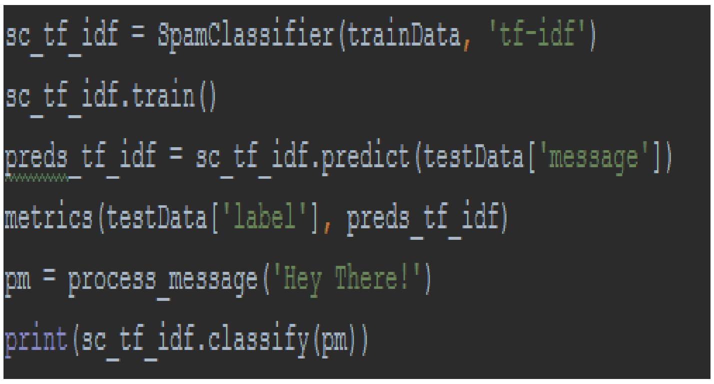
```
sc_tf_idf = SpamClassifier(trainData, 'tf-idf')
sc_tf_idf.train()
preds_tf_idf = sc_tf_idf.predict(testData['message'])
metrics(testData['label'], preds_tf_idf)
pm = process_message('congratulations ur awarded 100$')
print(sc_tf_idf.classify(pm))
```

Fig.7. – TF-IDF



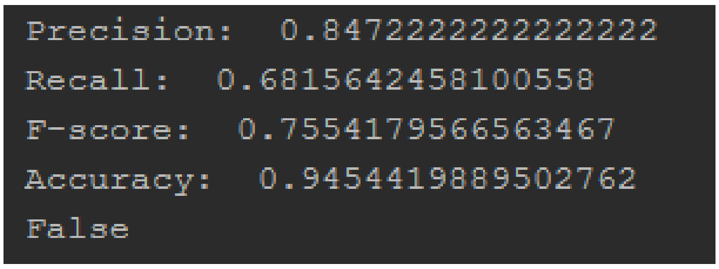
```
Precision:  0.9391891891891891
Recall:  0.7239583333333334
F-score:  0.8176470588235294
Accuracy:  0.9556191839656406
True
```

Fig.8. – Results for TF-IDF for Spam



```
sc_tf_idf = SpamClassifier(trainData, 'tf-idf')
sc_tf_idf.train()
preds_tf_idf = sc_tf_idf.predict(testData['message'])
metrics(testData['label'], preds_tf_idf)
pm = process_message('Hey There!')
print(sc_tf_idf.classify(pm))
```

Fig.9. – TF-IDF



```
Precision:  0.8472222222222222
Recall:  0.6815642458100558
F-score:  0.7554179566563467
Accuracy:  0.9454419889502762
False
```

Fig.10. – Results for TF-IDF for Ham

V. CONCLUSIONS AND FURTHER WORK

Spam emails are the biggest problem for the web data. This paper presents a Naïve Bayesian approach to deal with this problem. The classifier trained and tested the considered dataset from Kaggleand showed that it was effective in catching the spam content at higher accuracy and precision than the conventional spam filtering approaches.Although, it’s notpossible to achieve 100% accurate results, butthere is very much scope for identifying mail as spam emails or legitimate mails for text as well as multimedia messages. One direction is to strengthen the current inimical model to more user friendly model. The classifier is trained under the assumption that the distribution of Spam Letter(s) is/are constant over the time while realistically it is likely malicious spammers will change their pattern over the course of time. In order to tackle this, the program should also evolve and retrain at regular intervals and success rate should be tracked after each iteration.

REFERENCES

[1]. Chae, M. K., Abeer Alsadoon, P. W. C. Prasad, and Sasikumaran Sreedharan. "Spam filtering email classification (SFECM) using gain and graph mining algorithm." In Anti-Cyber Crimes (ICACC), 2017 2nd International Conference on, pp. 217-222. IEEE, 2017.

[2]. Alurkar, Aakash Atul, Sourabh Bharat Ranade, Shreeya Vijay Joshi, Siddhesh Sanjay Ranade, Piyush A. Sonewar, Parikshit N. Mahalle, and Arvind V. Deshpande. "A proposed data science approach for email spam classification using machine learning techniques." In Internet of Things Business Models, Users, and Networks, 2017, pp. 1-5. IEEE, 2017.

[3]. Neelavathi, C., and S. M. Jagatheesan. "Improving Spam Mail Filtering Using Classification Algorithms With Partition Membership Filter." (2016).

[4]. Feng, Weimiao, Jianguo Sun, Liguo Zhang, Cuiling Cao, and Qing Yang. "A support vector machine based naive Bayes algorithm for spam filtering." In Performance Computing and Communications Conference (IPCCC), 2016 IEEE 35th International, pp. 1-8. IEEE, 2016.

[5]. Kumar, Santosh, Xiaoying Gao, Ian Welch, and Masood Mansoori. "A machine learning based web spam filtering approach." In Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on, pp. 973-980. IEEE, 2016.

[6]. Roy, Kaushik, Sunil Keshari, and Surajit Giri. "Enhanced Bayesian spam filter technique employing LCS." In Computer, Electrical & Communication Engineering (ICCECE), 2016 International Conference on, pp. 1-6. IEEE, 2016.

[7]. Almeida, Tiago A., and Akebo Yamakami. "Compression‐based spam filter." Security and Communication Networks 9, no. 4 (2016): 327-335.

[8]. Pfeffer, Avi. Practical Probabilistic Programming. Manning Publications Co., 2016.

[9]. Vipin, N. S., and M. Abdul Nizar. "Efficient on-line SPAM filtering for encrypted messages." In Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on, pp. 1-5. IEEE, 2015.

[10]. Iyer, Akshay, Akanksha Pandey, Dipti Pamnani, Karmanya Pathak, and Jayshree Hajgude. "Email Filtering and Analysis Using Classification Algorithms." International Journal of Computer Science Issues (IJCSI) 11, no. 4 (2014): 115.

[11]. Rekha, S. Negi. "A Review on Different Spam Detection Approaches." International Journal of Engineering Trends and Technology (IJETT) 11, no. 6 (2014): 315-318.

[12]. Ba, Jimmy, Volodymyr Mnih, and Koray Kavukcuoglu. "Multiple object recognition with visual attention." arXiv preprint arXiv:1412.7755 (2014).

[13]. Mohammed, Sabah, Osama Mohammed, Jinan Fiaidhi, Simon James Fong, and Tai Hoon Kim. "Classifying Unsolicited Bulk Email (UBE) using Python Machine Learning Techniques." (2013).

[14]. Mali, Asmeeta. "Spam Detection Using Baysian with Pattren Discovery." International Journal of Recent Technology and Engineering (IJRTE) ISSN (2013): 2277-3878.

[15]. Ann Nosseir , Khaled Nagati and Islam Taj-Eddin,“Intelligent Word-Based Spam Filter Detection Using Multi-Neural Networks”, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013

[16]. Lin, Po-Ching, Ping-Hai Lin, Pin-Ren Chiou, and Chien-Tsung Liu. "Detecting spamming activities by network monitoring with Bloom filters." In Advanced Communication Technology (ICACT), 2013 15th International Conference on, pp. 163-168. IEEE, 2013.

[17]. Chiou, Pin-Ren, Po-Ching Lin, and Chun-Ta Li. "Blocking spam sessions with greylisting and block listing based on client behavior." In Advanced Communication Technology (ICACT), 2013 15th International Conference on, pp. 184-189. IEEE, 2013.

[18]. Geerthik. S and Anish .T.P, “Filtering Spam: Current Trends and Techniques”, International Journal of Mechatronics, Electrical and Computer Technology Vol. 3(8), Jul, 2013, pp 208-223, ISSN: 2305-0543 © Austrian E-Journals of Universal Scientific Organization.

[19]. Wisaeng, K. "A comparison of different classification techniques for bank direct marketing." International Journal of Soft Computing and Engineering (IJSCE) 3, no. 4 (2013): 116-119.

[20]. Freeman, David Mandell. "Using naive bayes to detect spammy names in social networks." In Proceedings of the 2013 ACM workshop on Artificial intelligence and security, pp. 3-12. ACM, 2013.

Aditya Gupta "Spam Filter using Naïve Bayesian Technique "International Journal of Computational Engineering Research (IJCER), vol. 08, no. 06, 2018, pp. 26-32.