PROJECT FOR EMAIL SPAM DETECTION

- by AbssZy

Table of Content

I. INTRODUCTION

In recent years social websites have become important components of the web. With their success, however has come a growing influx of spam. If left unchecked, spam threatens to undermine resource sharing, interactivity and openness. This project focuses on various countermeasures that are based on detection, demotion and prevention. Although various countermeasures have been proposed for email and web spam but still there are challenges.

Spam is a commonly said unwanted mail through the internet. In this trend Spam is became a major issue of this world to control legitimately. The large amount of spam will affect the bandwidth problems and mere nuisance to be considered. The issue in these spam mails is to mixed with the new mails and categorization is difficult to produce and it is very tough task to categorize the spam mails and legitimate mails. Through our project we attempt to solve this problem and categorize the emails efficiently. In the past few years, the increasing number of spams is making Internet usage more and more difficult.

According to a study of 2006, the messages that were spam, constituted about 40% of the total incoming messages in China. This situation is not being any better and hence it is the need of the hour to take steps to stop or at least reduce this overflow of spam messages around the Internet. Some of the present anti-spam approaches include (1) white-listing genuine senders and black-listing the senders who are potential spammers. (2) Hand crafted rule filtering approach (3) filtering the messages based on content. Although it's true that white-listing reduce the risk of accidentally blocking non-spammers from sending messages, but the problem with this approach is that they have to be used with other techniques, to differentiate between spam senders and genuine senders who are not there in the white-list. Since most spammers use fake addresses, black- listing has also lost its effectiveness. In case of hand-crafted rules, spammers can easily modify their messages in order to prevent triggering these hand-crafted rules. Content-based filtering is based on the idea that distinctive features in a message can be used for filtering the spam messages. It is further divided into rule-based approach and statistical-based approach. A rule-based approach follows the idea of expressing the domain knowledge in terms of a set of heuristic rules. But this leads to an extremely high cost as it requires to maintain a huge set of heuristic rules. In statistical approach, the differences among messages are expressed in terms of likelihood of certain events. It may be observed that a statistical model may work well in one corpus but not in another one which has different characteristics. Hence, we need more efficient spam detecting techniques.

1. AIM

Here we classify emails of two types – ham or spam, ham is generally to denote emails that are useful i.e. not spam and spam are the unwanted emails. Using certain set of words, we can classify an email as spam or ham from a given dataset, like certain words like (Won, lottery etc. are considered to be spam) and word likes (Meet me, see you tomorrow etc. are considered to be ham).

2. OBJECTIVE

To detect spams, this work proposes a spam detection approach using Naive Bayesian (NB) classifier (Multinomial and Bernoulli Naïve Bayesian) and Logistic Regression, where this classifier identifies email messages as being spam or legitimate, based on the content (i.e. body) of these messages. Each email is represented as a bag of its body's words (features). Certain preprocessing (Tokenization, stop word removal, and stemming) was needed to drop out any redundant data. In this project instead of the body we see the subject of these emails and classify them to increase accuracy level, certain statistics were suggested to extend NB algorithm with.

II. SYSTEM REQUIREMENT

Hardware requirements

a. Laptop

b. RAM 8GB+

c. VRAM 1050 Nvidea +

d. SSD 128GB

Software requirements

a. Jupyter Notebook

b. Python IDLE

c. Email Subject Dataset

III. PROJECT MODULE

Spamming Techniques

1.Botnets allows spammers to use command-and-control servers, or C&C servers, to both harvest email addresses and distribute spam.

2.Snowshoe spam is the technique of using a wide range of IP addresses and email addresses with neutral reputations to distribute spam widely.

3.Another method spammers use is blank email spam. This involves sending email with an empty message body and subject line. The technique could be used in a directory harvest, an attack against an email server that seeks to validate email addresses for a distribution list by identifying invalid bounced addresses. In this type of attack, the spammer does not need to enter text into the email. In other instances, seemingly blank emails may hide certain virusesand worms that can be spread through HTML code embedded in the email.

Spammers have developed methods to confuse the nature of their unwanted email or find a way to bypass spam filters. Because spam-filtering programs often search for certain patterns or words in the subject lines and message bodies of email, spam emails often contain misspelled words or extra characters.

With image spam, the text of a message is stored as a JPEG or GIF file and placed into the email body. The text is often computer-generated and unintelligible to human readers. This method attempts to avoid detection from text-based spam filters. Some newer filters can read images and locate text in them; however, this can inadvertently filter out non spam emails that happen to contain images featuring text.

IV. METHODOLOGY

Things Needed:

1.Dataset to train test-sets.

2.Code.

2.1.

2.1.1 Building a Spam filter.

2.2.

2.2.1. Preparing text data.

2.2.2. Creating word dictionary.

2.2.3. Feature extraction process.

2.2.4. Training and Testing classifier.

After that here we have Multinomial NB, Bernoulli NB and Logistic Regression so we use all these three to give us an output if the code is spam or ham, if occurrences of spam is more than ham then it is ham else it is spam.

Here we use three methodologies so that we can get an accurate result

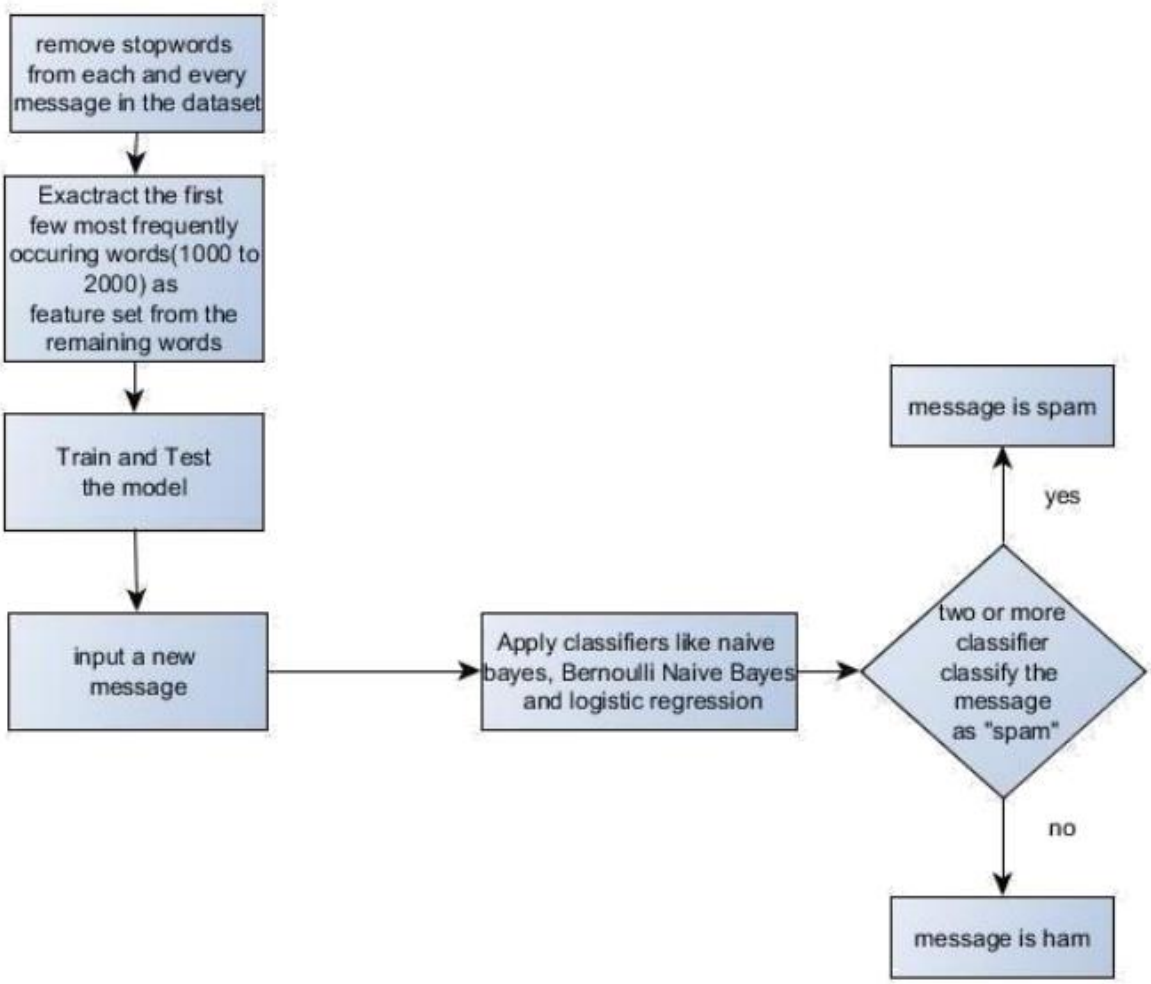
Methodology 1: Multinomial Naive Bayes: It will classify a document based on the counts it finds of multiple keywords i.e it cares about counts for multiple features that do occur.

Methodology 2: Bernoulli Naïve Bayes: It can only focus on a single keyword, but will also count how many times that keyword does not occur in the document. i.e. it cares about counts for a single feature that do occur and counts for the same feature that do not occur

Note: that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one

Methodology 3: Logistic Regression: It is a predictive analysis; it gives binary values (either 1 or 0). Thus, being most suitable for this project (1 for Spam and 0 for Not spam/ Ham). It is a classification algorithm used to assign observations to a discrete set of classes. It transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Block Diagram



Algorithms Used:

1. Naïve Bayes

The Naïve Bayes algorithm is the most used algorithm and is derived from Bayesian Decision Theory. According to Bayes Theorem and theorem of total probability, the probability that a document d that has vector x=<x1,……,xn> belongs to category c is:

$$P(c_j \mid d_x) = \frac{P(c_j)P(d_x \mid c_x)}{\sum_{j \in \{spam, ham\}} p(c_j)P(d_x \mid c_j)}$$

This classifier is based on the premise that x1…….xn are conditionally independent given the category c. Although this assumption is not true for most of the real-world

task, Naïve Bayes often performs classification very accurately. P(ci) can be easily calculated using the frequencies of the training corpus. Naïve Bayes has two generative model, multivariate Bernoulli event model (MBM) and Multinomial Model (MM). Both of these use the Naïve Bayes assumption. The P(dx | cj) on the two models is different. A test indicated that there is not much difference. Actually, Multivariate Bernoulli event model is more effective than Multinomial model if the vocabulary size is small. Also, it is easier to perform.

2. Multinomial Naïve Bayes

This generative model calculates the conditional probability of a class when input data is given. This algorithm assumes that input features are independent of each other. The Multinomial Naïve Bayes is a version of Naïve Bayes which is mostly used for documentation and text classification. It is true that there are more complicated models for discrete data classification, but Multinomial Naïve Bayes is still good enough and robust in terms of performance. Also, it allows training from mini-batches of data where maximum likelihood (ML) estimations can be obtained. There is a common idea to smooth the ML estimate using a regulation parameter. (α)

3. Bernoulli Naïve Bayes

In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence features are used rather than term frequencies. If Xi is a boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class Ck is given by

$$p(x \mid c_k) = \prod_{i=1}^{n} p_{ki}^{xi} (1 - p_{ki})^{(1-x_i)}$$

where Pki is the probability of class Ck generating the term Xi. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

4. Logistic Regression

Logistic regression is a probabilistic linear classifier, parameterized by a weight matrix w and a bias b. It enables the system to estimate categorical results with the help of a group of independent variables. The classifier equation for logistic regression model is given as

$$y = \text{sgn}(w^T x + b) \tag{1}$$

where y ε{-1,1} denoting the output class recognized for the input **x** fed to the system. The above classifier equation is rewritten using augmented weight matrix θ as

$$y = sgn(\theta^T x) \tag{2}$$

The augmented weight matrix is obtained during the training phase of logistic regression model as explained below. Let {xj, yj} for j = 1,2,…*m* denote the training data set, where yj is the target output for training data xj. The weight matrix is first initialized to 1, i.e., θ = **1**. The equation for weight updates is given as

$$\theta_j(n) = \theta_j(n-1) + \alpha.\, \vartheta_j \tag{3}$$

Where α is the learning rate of the model and υ_j is given as

$$\vartheta_j = \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \tag{4}$$

Where m denotes the number of samples available in training dataset, j € {1,2,….,m} and $h_\theta(x)$ is the logistic function given as

$$h_\theta(x) = \frac{1}{(1+e^{-\theta Tx})} \tag{5}$$

The average cost function $J(\theta)$ for logistic regression model is given as

$$J(\theta) = (1/m)\sum_{i=1}^{m} (\text{Cost}(h_\theta(x^{(i)}), y^{(i)})) \tag{6}$$

Where $\text{Cost}(h_\theta(x^{(i)}), y^{(i)})$ is given as

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if y} = 1 \\ -\log(1-h_\theta(x)) & \text{if y} = 0 \end{cases} \tag{7}$$

In order to obtain minimum average cost for the logistic regression model designed, the Gradient Descent method is employed to obtain the iterative expression for J( ) as

$$J(\theta) = J(\theta) + (-1/m)(\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)})) \tag{8}$$

V. DATASET DETAILS

Link : https://github.com/AbssZy/emailspamdetection/blob/main/smsspamcollection.zip

Publicly available since 02 October 2020.

Total set of 5574.

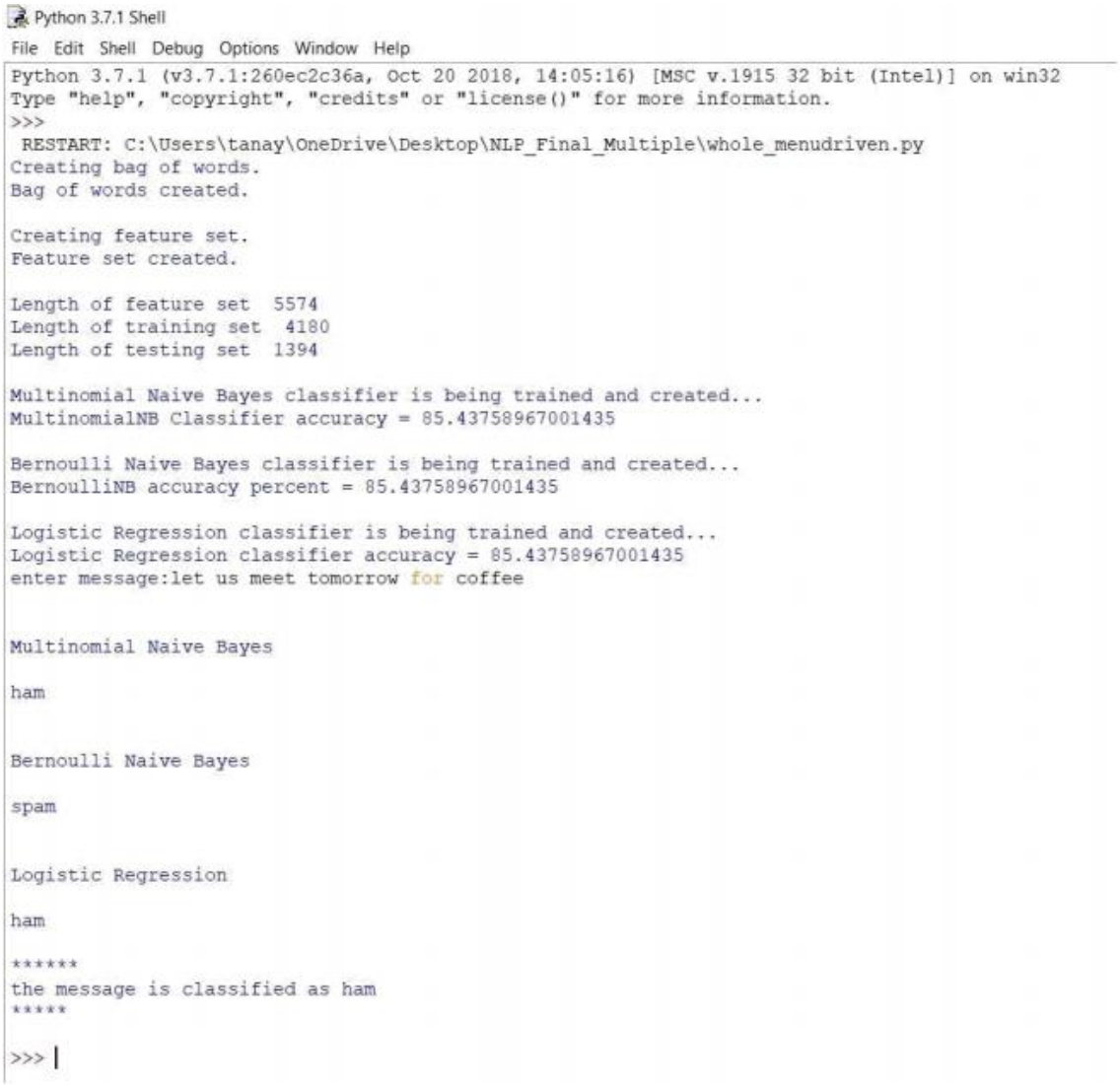
VI. CHALLENGES FACED

- Implementing 3 different algorithms on the same dataset to give accurate results.
- Finding the right dataset.
- Getting good accuracy.

VII. RESULTS

Output:

Example of Ham



Example of Spam


Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\tanay\OneDrive\Desktop\NLP_Final_Multiple\whole_menudriven.py
Creating bag of words.
Bag of words created.
Creating feature set.
Feature set created.
Length of feature set 5574
Length of training set 4180
Length of testing set 1394
Multinomial Naive Bayes classifier is being trained and created...
MultinomialNB Classifier accuracy = 86.22668579626973
Bernoulli Naive Bayes classifier is being trained and created...
BernoulliNB accuracy percent = 86.22668579626973
Logistic Regression classifier is being trained and created...
Logistic Regression classifier accuracy = 86.22668579626973
enter message:hurry you have won 1000$ as jackpot
Multinomial Naive Bayes
ham
Bernoulli Naive Bayes
spam
Logistic Regression
ham
******
the message is classified as spam
*****
>>>

VII. CONCLUSION

Complexity Analysis:

The algorithms used in this project are Naïve Bayes algorithm, Multinomial Naïve Bayes algorithm and Logistic Regression.

Their complexity analysis is as follows:

Naïve Bayes Classifier and Multinomial Bayes algorithm:

Time complexity:

Training time: O(|D|L(avg)+ |C||V|)

Where L(avg) is the average length of a document in D.

Test time: O(|C|L(t))

Where L(t) is the average length of a test document.

Space complexity: O(Nd)

Bernoulli Naïve Bayes:

T (n) = O (nmc)

Here 'm' is the number of parameters; 'c' is for classes

Logistic Regression:

Time complexity = O(N)

Therefore, it works really fast and is very efficient.

VIII. REFERENCE

[1] SANTOSH KUMAR, XIAOYING GAO AND IAN WELCH, "NOVEL FEATURES FOR WEB SPAM DETECTION" IN 2016 IEEE 28TH INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE

[2] Wuxain Zhang and Hung-Min Sun," Instagram Spam Detection" in 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing

[3] Nasim eshraqi,Mehrdad Jalali and Mohammad Hossein Moattar," Spam Detection In Social Networks: A Review" in 2015 International Congress on Technology, Communication and Knowledge (ICTCK)

[4] Himank Gupta, Mohd. Saalim Jamal, Sreekanth Madisetty and Maunendra Sankar Desarkar, "A Framework for Real-Time Spam Detection in Twitter" in2018 10th International Conference on Communication Systems & Networks (COMSNETS)

[5] Shafi'i Muhammad Abdulhamid, Muhammad Shafie Abo Latif, Haruna Chiroma, Oluwafemi Osho, Gaddafi Abdul-Salaam, Adamu I, Abubakar and

Tutut Herawan,"A Review on Mobile SMS Spam Filtering Techniques" in IEEE Access 10.1109/ACCESS.2017.2666785

[6] Ray Hunt and James Carpinter "Current and new developments in spam filtrating." In 2016 14th IEEE International Conference on Networks.

[7] Draško Radovanović, Božo Krstajić,"Review Spam Detection using Machine Learning" in 2018 IEEE 23rd International Scientific-Professional Conference on Information Technology (IT)

[8] Qin Luo, Bing Liu, Junhua Yan, Zhongyue He "Research of a Spam Filtering Algorithm Based on Naïve Bayes and AIS" in 2010 IEEE International Conference on Computational and Information Sciences.

[9] Shrawan Kumar Trivedi "A Study of Machine Learning Classifiers for Spam Detection" in2016 IEEE 4th International Symposium on Computational and Business Intelligence.

[10] Zhiyang zia, Wei Gao, Weiwei Li, Youmi Xia "Research on web spam detection based on Support vector machine" in 2012 IEEE International Conference on Communication on Systems and Network Technologies.

[11] Nitin Jindal, Bing Liu, "Analyzing and detecting review spam" in 2007 Seventh IEEE International Conference on Data Mining (ICDM 2007).

[12] M.Sasaki, H.Shinnou, "Spam detection using text clustering" in 2005 International Conference on Cyberworlds(CW'05).

[13] Surendra Sedhai, Aixin Sun, "Semi-supervised spam detection in twitter stream", in Volume 5, Issue:1, March 2018 IEEE Transactions on Computational Social Systems.

[14] Y.Rebahi, D Sisalem, "SIP Spam Detection", in 29-31 Aug 2006 International Conference on Digital Telecommunications (ICDT'06).

[15] Qian XU, Evan Wei Xiang, Qiang Yang, Jiachun Du, Jieping Zhong, “SMS Spam Detection using non-content features”, in Volume 27, Issue 6, Nov-Dec 2012, IEEE Intelligent Systems.

[16] J. Ioannidis, "Fighting spam by encapsulating policy in email addresses", Network and Distributed System Security Symposium, Feb 6–7 2003.

[17] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz, "A Bayesian approach to filtering junk email," July 1998.

[18] R. Jennings, The Cost of Spam, 2009:

Web page: http://www.ferris.com/research- library/industry-statistics, 2009.

[19] F. Benevenuto, G. Magno, T. Rodrigues, V. Almeida, "Detecting spammers on Twitter", Proc. Collaboration Electron. Messaging Anti-Abuse Spam Conf. (CEAS), vol. 6, pp. 12, 2010.

[20] R. S. Sexton, R. E. Dorsey, J. D. Johnson, "Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation", Decision Support Systems, vol. 22, no. 2, pp. 171-185, 1998.