NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC)

☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

Course Code: 16CS704

Course Name: Machine Learning

Spam Filter using Naive Bayes Algorithm

Semester: 7th   Section: C

Submitted by:

Saurabh D Rao	4NM16CS132
Sharath Kumar A	4NM16CS133

Submitted to:

Mrs. Divya Jennifer D'Souza

Department of Computer Science and Engineering

Date of submission: October 14, 2019

Signature of Course Instructor

Table of Contents

Abstract

For the last few years the upsurge in the volume of unwanted emails called spam has created an intense need for the development of more dependable and robust antispam filters. Machine learning methods of recent are being used to successfully detect and filter spam emails. We present a systematic review of some of the popular machine learning based email spam filtering approaches. Our review covers survey of the important concepts, attempts, efficiency, and the research trend in spam filtering. The preliminary discussion in the study background examines the applications of machine learning techniques to the email spam filtering process of the leading internet service providers (ISPs) like Gmail, Yahoo and Outlook emails spam filters. Discussion on general email spam filtering process, and the various efforts by different researchers in combating spam through the use machine learning techniques was done.

Literature Survey

The article written by Dr. Sebastian [1] describes how Naïve Bayes Algorithm can be utilized to classify a given email message as spam or ham. Bag of Word model is used which was constructed after pre-processing an email message through stop-word removal, stemming and lemmatizing. Natural Language Toolkit [2] is an open source program developed to perform various text processing tasks. This package defines methods for stop word removal, lemmatization and stemming. The article [3] is based on n-gram model for spam filtering where they found out that by using a n-gram of size 3 and 4 would produce better results when compared to the normal unigram model.

The web article [4] shows a step by step expressions and values for deciding if a sentence is sports or non-sports related. Here they implemented a count based approach and used this as prior for calculating the conditional probabilities. They also show how to make use of Laplace Smoothing to prevent probabilities to always be zero. Another interesting web article [5] which uses the concept of logarithms to prevent underflow while working with large denominators. On applying the logarithm for each term of the Bayes Theorem, the multiplication would be converted to addition and division would be a simple subtraction of the logarithms of each terms.

Methodology

Naïve Bayes Classifier

Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective naive comes from the assumption that the features in a dataset are mutually independent.

In probability theory and statistics, Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

Bayes' theorem is stated mathematically as the following equation.

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

Where, A and B are events and P(B)≠ 0.

- $P(A|B)$ is the conditional probability : the likelihood of event A occurring given that B is true.
- $P(B|A)$is also a conditional probability : the likelihood of event B occurring given A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other; this is known as marginal probability.

Dataset

In the current project we make use of text messages dataset which consists of spam and ham (regular messages) along with its corresponding tags. The dataset consists of 2047 samples of text which specifies the message and the category. There are two categories *spam* and *ham*.

Procedure

Pre-processing:

The text from the dataset cannot be used directly as it may contain some words and conjunctions that are useless for the classification. Thus pre-processing of the words is necessary to retain only the useful words. The following steps are involved

1) *Removing contractions:*
   A contraction is a word or phrase that has been shortened by dropping one or more letters. In writing, an apostrophe is used to indicate the place of the missing letters.

2) *Removing stop words:*
   In natural language processing, useless words (data), are referred to as stop words. Stop Words: A stop word is a commonly used word such as "the", "a", "an", "in".

3) *Perform stemming:*
   Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma.

Feature Engineering

The important part is to find the features from the data to make machine learning algorithms works. In this case, we have text. We need to convert this text into numbers that we can do calculations on. We use word frequencies. That is treating every document as a set of the words it contains. Our features will be the counts of each of these words. Then, we need to convert the probability that we wish to calculate into a form that can be calculated using word frequencies.

In our case, the probability that we wish to calculate can be calculated as:

$$P(spam|play\ now\ and\ win\ a\ car) = \frac{P(play\ now\ and\ win\ a\ car|spam)\ \times P(spam)}{P(play\ now\ and\ win\ a\ car)}$$

Similarly probabilities are calculated for every category.

Since there is an assumption that every word is independent of one another. Now we look at the individual words in the sentence rather than the whole sentence. Here we rewrite the probability we wish to calculate accordingly.

$$\begin{aligned} &P(play\ now\ and\ win\ a\ car|spam) \\ &\qquad = P(play|spam) \times P(now|spam) \times P(and|spam) \times P(win|spam) \\ &\qquad \times P(a|sapm) \times P(car|spam) \end{aligned}$$

Prior to fitting the model and using machine learning algorithms for training, we need to think about how to best represent a text document as a feature vector. A commonly used model in *Natural Language Processing* is the so-called *bag of words* model which is shown below.

$$P(w_1, \cdots, w_n | Y = y) = \prod_{i=1}^{n} P(w_i | Y = y)$$

Since we are calculating the overall probability of the class by multiplying individual probabilities for each word, we would end up with an overall probability of 0 for the positive class. So we make use of smoothing algorithm such as Laplace smoothing.

Laplace Smoothing

We modify our conditional word probability by adding 1 to the numerator and modifying the denominator as such:

$$P\ (\ wi\ |\ cj\ )\ =\ [\ count(\ wi, cj\ )\ +\ 1\ ]\ /\ [\ \Sigma w \in V(\ count\ (\ w, cj\ )\ +\ 1\ )\ ]$$

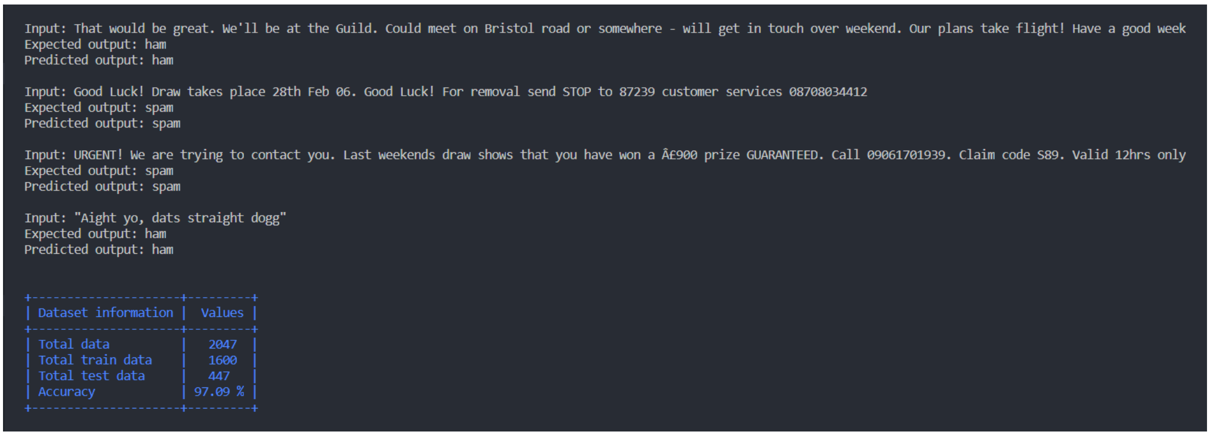
This can be simplified to

$$P\ (\ wi\ |\ cj\ )\ =\ [\ count(\ wi, cj\ )\ +\ 1\ ]\ /\ [\ \Sigma w \in V(\ count\ (\ w, cj\ )\ )\ +\ |V|\ ]$$

where |V| is our vocabulary size.

Results

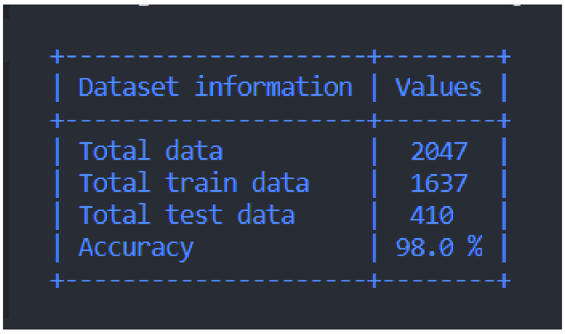
For the present experiment the Naïve Bayesian model is trained on the text messages dataset which consists of text messages along with its category. The following figure shows the accuracy obtained while classifying the new input samples into our implementation of the Naïve Bayes algorithm.



```
Input: That would be great. We'll be at the Guild. Could meet on Bristol road or somewhere - will get in touch over weekend. Our plans take flight! Have a good week
Expected output: ham
Predicted output: ham

Input: Good Luck! Draw takes place 28th Feb 06. Good Luck! For removal send STOP to 87239 customer services 08708034412
Expected output: spam
Predicted output: spam

Input: URGENT! We are trying to contact you. Last weekends draw shows that you have won a Â£900 prize GUARANTEED. Call 09061701939. Claim code S89. Valid 12hrs only
Expected output: spam
Predicted output: spam

Input: "Aight yo, dats straight dogg"
Expected output: ham
Predicted output: ham


+---------------------+---------+
| Dataset information |  Values |
+---------------------+---------+
| Total data          |   2047  |
| Total train data    |   1600  |
| Total test data     |   447   |
| Accuracy            | 97.09 % |
+---------------------+---------+
```

Figure 1: Accuracy and the classification of some of the sample texts using our implementation.

The following figure shows the accuracy obtained by using the inbuilt Naïve Bayes classifier available in the Scikit-learn library.



```
+---------------------+--------+
| Dataset information | Values |
+---------------------+--------+
| Total data          |  2047  |
| Total train data    |  1637  |
| Total test data     |  410   |
| Accuracy            | 98.0 % |
+---------------------+--------+
```

Figure 2: Accuracy of classification using the algorithm built into sklearn

It is observed that the accuracy of the classification of our implementation of Naïve Bayes algorithm is close to that of the one that is found in the machine learning libraries.

Conclusion

Text messages are part of our day to day lives. No days goes by without us sending messages or receiving them. Receiving the messages can be from the people we know or from some websites we have signed up in. It is crucial to know which of these messages are spam and which are ham. In this project we have successfully implemented the Naïve Bayes algorithm using for the classification of the messages into spam or ham. This implementation can be further refined to provide a better accuracy.

References

[1] D. S. Raschka, "Naive Bayes and Text Classification," 4 October 2014. [Online]. Available: https://sebastianraschka.com/Articles/2014_naive_bayes_1.html.

[2] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," 17 May 2002. [Online]. Available: http://www.nltk.org/.

[3] I. KANARIS, K. KANARIS, I. HOUVARDAS and E. STAMATATOS , "WORDS VS. CHARACTER N-GRAMS FOR ANTI-SPAM FILTERING," *International Journal on Artificial Intelligence Tools ,* pp. 1-20, 2006.

[4] B. Stecanella, "A practical explanation of a Naive Bayes classifier," 25 May 2017. [Online]. Available: https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/.

[5] H. H. Nguyen, "Algorithms for Text Classification," 4 February 2019. [Online]. Available: https://towardsdatascience.com/algorithms-for-text-classification-part-1-naive-bayes-3ff1d116fdd8.