VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

# Artificial intelligence project

# SMS Spam Filteng Using Naïve boyes

## J Component
## Final Review

**By-Sakshi Rao Varam**

# Table of Contents

| Chapter No. | Contents |
|---|---|
| 1 | Abstract |
| 2 | Objective |
| 3 | SPAM |
| 4 | Naive Bayes spam filtering |
| 4.1 | Process |
| 4.2 | Mathematical foundation |
| 4.3 | Advantages |
| 4.4 | Disadvantages |
| 4.5 | Stages |
| 5 | Literature Review |
| 6 | Related Work |
| 7 | Data Set Description |
| 8 | Code |
| 9 | Output |
| 10 | Results and Accuracy |
| 11 | Conclusion |
| 12 | References |

# Abstract

The Short Message Service (SMS) have an important economic impact for end users and service providers. Spam is a serious universal problem that causes problems for almost all users. Several studies have been presented, including implementations of spam filters that prevent spam from reaching their destination. Naïve Bayesian algorithm is one of the most effective approaches used in filtering techniques. The computational power of smartphones are increasing, making increasingly possible to perform spam filtering at these devices as a mobile agent application, leading to better personalization and effectiveness. The challenge of filtering SMS spam is that the short messages often consist of few words composed of abbreviations and idioms. In this paper, I propose an anti-spam technique based on Artificial Immune System (AIS) for filtering SMS spam messages. The proposed technique utilizes a set of some features that can be used as inputs to spam detection model. The idea is to classify message using trained dataset that contains Phone Numbers, Spam Words, and Detectors. My proposed technique utilizes a double collection of bulk SMS messages Spam and Ham in the training process. I state a set of stages that help us to build dataset such as tokenizer, stop word filter, and training process. The results applied to the testing messages show that the proposed system can classify the SMS spam and ham with accurate compared with Naïve Bayesian algorithm.

# Objective

The objective of the project was to build a model that can accurately differentiate spam messages from ham ones. In doing so, I tried to showcase the ability of naive-bayes algorithm to accurately predict the message as spam or ham which will be used to detect unsolicited and unwanted SMS and prevent those messages from getting to a user's inbox. Like other types of filtering programs, a spam filter looks for certain criteria on which it bases judgments.

# SPAM

Today’s SPAM, also known as junk is a far cry from 1937 and has nothing to do with solving problems, and everything to do with creating them.

Spammers may argue that it’s not really a problem and that you can merely delete what you don’t want, but that is naïve in the extreme. Whether received as a private user, or as a business professional, SPAM is a cyber menace for a number of reasons:

- On the most basic level, receiving a large volume of SPAM SMS wastes valuable bandwidth and time. Private user is forced to individually delete their unwanted message and administrators have to fight similar problems but on a far larger scale in an attempt to keep our businesses operational.
- This wasted time and effort inevitably leads to a loss in productivity as valuable resources are in-efficiently allocated to non-profitable enterprises.
- SPAM is a prime means of transferring electronic viruses and malware infections, whether deliberately, or by accident as the direct result of generating mass SMS to and from a large number of recipients.
- SPAM is also not merely restricted to mass marketing schemes. Professional criminal organizations use it to instigate complex frauds and scams such as phishing attacks and “419” Nigerian fraud type scams that have made the news in recent years.
- Cost shifting can be an issue. It is incredibly cheap for the spammers to send hundreds of thousands of SMS an hour, but the cost of receiving them can be many times greater, both in terms of monetary outlay and also the cost of rectifying all of the other associated problems.
- Perhaps the greatest problem is the irritation factor. SPAM is extremely annoying and there is nothing worse than accessing your SMS provider to find countless SMS advertising products you don’t need, pornography you don’t want and scams you want to avoid.

## Naive Bayes spam filtering

Naive Bayes classifiers are a popular statistical technique of SMS filtering. They typically use bag of words features to identify spam SMS, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam SMS and then using Bayes' theorem to calculate a probability that an SMS is or is not spam.

Naive Bayes spam filtering is a baseline technique for dealing with spam that can tailor itself to the SMS needs of individual users and give low false positive spam detection rates that are generally acceptable to users. It is one of the oldest ways of doing spam filtering, with roots in the 1990s.

## Process

Particular words have particular probabilities of occurring in spam SMS and in legitimate SMS. For instance, most users will frequently encounter the word "Viagra" in spam SMS, but will seldom see it in other SMS. The filter doesn't know these probabilities in advance, and must first be trained so it can build them up. To train the filter, the user must manually indicate whether a new SMS is spam or not. For all words in each training SMS, the filter will adjust the probabilities that each word will appear in spam or legitimate SMS in its database. For instance, Bayesian spam filters will typically have learned a very high spam probability for the words "Viagra" and "refinance", but a very low spam probability for words seen only in legitimate SMS, such as the names of friends and family members.

After training, the word probabilities (also known as likelihood functions) are used to compute the probability that an SMS with a particular set of words in it belongs to either category. Each word in the SMS contributes to the SMS's spam probability, or only the most interesting words. This contribution is called the posterior probability and is computed using Bayes' theorem. Then, the SMS's spam probability is computed over all words in the SMS, and if the total exceeds a certain threshold (say 95%), the filter will mark the SMS as a spam.

As in any other spam filtering technique, SMS marked as spam can then be automatically moved to a "Junk" SMS folder, or even deleted outright. Some software implement quarantine mechanisms that define a time frame during which the user is allowed to review the software's decision.

The initial training can usually be refined when wrong judgements from the software are identified (false positives or false negatives). That allows the software to dynamically adapt to the ever-evolving nature of spam.

Some spam filters combine the results of both Bayesian spam filtering and other heuristics (pre-defined rules about the contents, looking at the message's envelope, etc.), resulting in even higher filtering accuracy, sometimes at the cost of adaptiveness.

## Mathematical foundation

Bayesian SMS filters utilize Bayes' theorem. Bayes' theorem is used several times in the context of spam:

- a first time, to compute the probability that the message is spam, knowing that a given word appears in this message
- a second time, to compute the probability that the message is spam, taking into consideration all of its words (or a relevant subset of them)
- sometimes a third time, to deal with rare words

## Computing the probability that a message containing a given word is spam

The formula used by the software to determine that, is derived from Bayes' theorem: -

$$\Pr(S|W) = \frac{\Pr(W|S)\cdot\Pr(S)}{\Pr(W|S)\cdot\Pr(S)+\Pr(W|H)\cdot\Pr(H)}$$

## Combining individual probabilities

$$p = \frac{p_1 p_2 \cdots p_N}{p_1 p_2 \cdots p_N + (1-p_1)(1-p_2)\cdots(1-p_N)}$$

Where ‘p’ is the probability that the suspect message is spam;

## Dealing with rare words

$$\Pr'(S|W) = \frac{s\cdot\Pr(S)+n\cdot\Pr(S|W)}{s+n}$$

## Advantages

One of the main advantages of Bayesian spam filtering is that it can be trained on a per-user basis. The word probabilities are unique to each user and can evolve over time with corrective training whenever the filter incorrectly classifies an SMS. As a result, Bayesian spam filtering accuracy after training is often superior to predefined rules.

It can perform particularly well in avoiding false positives, where legitimate SMS is incorrectly classified as spam. For example, if the SMS contains the word "Nigeria", which is frequently used in Advance fee fraud spam, a predefined rules filter might reject it outright. A Bayesian filter would mark the word "Nigeria" as a probable spam word, but would take into account other important words that usually indicate legitimate e-mail. For example, the name of a spouse may strongly indicate the e-mail is not spam, which could overcome the use of the word "Nigeria." Super simple, you're just doing a bunch of counts. If the NB conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models like logistic regression, so you need less training data. And even if the NB assumption doesn't hold, a NB classifier still often does a great job in practice. A good bet if want something fast and easy that performs pretty well. Its main disadvantage is that it can't learn interactions between features (e.g., it can't learn that although you love movies with Brad Pitt and Tom Cruise, you hate movies where they're together).

## Disadvantages

Depending on the implementation, Bayesian spam filtering may be susceptible to Bayesian poisoning, a technique used by spammers in an attempt to degrade the effectiveness of spam filters that rely on Bayesian filtering. A spammer practicing Bayesian poisoning will send out SMS with large amounts of legitimate text (gathered from legitimate news or literary sources). Spammer tactics include insertion of random innocuous words that are not normally associated with spam, thereby decreasing the SMS's spam score, making it more likely to slip past a Bayesian spam filter. However, with (for example) Paul Graham's scheme only the most significant probabilities are used, so that padding the text out with non-spam-related words does not affect the detection probability significantly.

Words that normally appear in large quantities in spam may also be transformed by spammers. For example, «Viagra» would be replaced with «Viaagra» or «V!agra» in the spam message. The recipient of the message can still read the changed words, but each of these words is met more rarely by the Bayesian filter, which hinders its learning process. As a general rule, this spamming technique does not work very well, because the derived words end up recognized by the filter just like the normal ones.

## Stages

- Environment Setting
- Convert a corpus to a vector format: bag-of-words approach
- Massage the raw message (sequence of characters) into vectors (sequences of numbers)
- split a message into its individual words and return a list
- remove punctuation

- remove very common words('the','a',etc)
- Vectorization
- vectoring our messages
- convert each message (represented as a list of tokens) into a vector
- count DF in the vector
- weight the counts(IDF)
- Normalize the vectors to unit length (L2 norm)

## Literature Review

Globally, short messaging service (SMS) is one of the most popular and also most affordable telecommunication service packages. However, mobile users have become increasingly concerned regarding the security of their client confidentiality. This is mainly due to the fact that mobile marketing remains intrusive to the personal freedom of the subscribers. SMS spamming has become a major nuisance to the mobile subscribers given its pervasive nature. It incurs substantial cost in terms of lost productivity, network bandwidth usage, management, and raid of personal privacy. Thus, in short spamming threatens the profits of the service providers. Mobile SMS spams frustrate the mobile phone users, and just like email spams, they cause new societal frictions to mobile handset devices. Email spam is sent or received via the World Wide Web, while the SMS mobile spam is typically broad casted via a mobile network.

Spam can be described as unwanted or unsolicited electronic messages sent in bulk to a group of recipients. The messages are characterized as electronic, unsolicited, commercial, mass constitutes a growing threat mainly due to the following factors:

1. the availability of low-cost bulk SMS plans
2. reliability (since the message reaches the mobile phone user)
3. low chance of receiving responses from some unsuspecting receivers
4. the message can be personalized. Mobile SMS spam detection and prevention is not a trivial matter

It has taken on a lot of issues and solutions inherited from relatively older scenarios of email spam detection and filtering. Unsolicited SMS text messages are a common occurrence in our daily life and consume communication time, bandwidth and resources. Although the existing spam filters provide some level of performance, the spams misinform receivers by manoeuvring data samples.

# Related Work

| SNO. | NAME | AUTHORS | TECHNOLOGY USED | CONCLUSION |
|---|---|---|---|---|
| 1 | Naïve Bayes Text Classifier | Haiyi Zhang, Di Li | Text categorization, Detectors, Bayesian methods, Probability, Classification algorithms, Inference algorithms | In this paper, a spam email detector is developed using naive Bayes algorithm. They use pre-classified emails (priory knowledge) to train the spam email detector. With the model generated from the training step, the detector is able to decide whether an email is a spam email or an ordinary email. |
| 2 | A novel framework for SMS spam filtering | Efnan Sora Gunal, Semih Ergin, Serkan Gunal, Alper Kursat Uysal | unsolicited SMS messages, feature selection, information gain, discriminative feature subsets, Bayesian-based classifier, SMS message categorization, real-time mobile application, Bayes methods, feature extraction, information filtering, mobile computing, pattern classification, real-time systems | A novel framework for SMS spam filtering is introduced in this paper to prevent mobile phone users from unsolicited SMS messages. The framework makes use of two distinct feature selection approaches based on information gain and chi-square metrics to find out discriminative features representing SMS messages. The discriminative feature subsets are then employed in two different Bayesian-based classifiers, so that SMS messages are categorized as either spam or legitimate. Moreover, the paper introduces a real-time mobile application for Android™ based mobile phones utilizing the proposed spam filtering scheme, as well. Hence, SMS spam messages are silently filtered out without disturbing phone users. Effectiveness of the filtering framework is evaluated on a large SMS message collection including legitimate and spam messages. |
| 3 | A Bayesian Classification Approach | Steven Kay, Paul M. Baggenstoss, Haibo He, Bo Tang | Feature selection, text categorization, class-specific features, PDF projection and estimation, naive Bayes, | In this paper, they present a Bayesian classification approach for automatic text categorization using class-specific features. Unlike conventional |

| | | | | |
|---|---|---|---|---|
| | Using Class-Specific Features for Text Categorization | | dimension reduction | text categorization approaches, their proposed method selects a specific feature subset for each class. To apply these class-specific features for classification, they follow Baggenstoss's PDF Projection Theorem (PPT) to reconstruct the PDFs in raw data space from the class-specific PDFs in low-dimensional feature subspace, and build a Bayesian classification rule. One noticeable significance of their approach is that most feature selection criteria, such as Information Gain (IG) and Maximum Discrimination (MD), can be easily incorporated into our approach. They evaluate their method's classification performance on several real-world benchmarks, compared with the state-of-the-art feature selection approaches. |
| 4 | Feature selection for text classification with Naïve Bayes | Jingnian Chen, Houkuan Huang, Shengfeng Tian, Youli Qu | Text classification, Feature selection, Text preprocessing, Naïve Bayes | This paper presents two feature evaluation metrics for the Naïve Bayesian classifier applied on multi-class text datasets: Multi-class Odds Ratio (MOR), and Class Discriminating Measure (CDM). Experiments of text classification with Naïve Bayesian classifiers were carried out on two multi-class texts collections. As the results indicate, CDM and MOR gain obviously better selecting effect than other feature selection approaches. |
| 5 | Feature selection for multi-label naive Bayes classification | Min-Ling Zhang, José M.Peña, Victor Robles | Multi-label learning, Naive Bayes, Feature selection, Principal component analysis, Genetic algorithm | In this paper, this learning problem is addressed by using a method called Mlnb which adapts the traditional naive Bayes classifiers to deal with multi-label instances. Feature selection mechanisms are incorporated into Mlnb to improve its performance. Firstly, feature extraction techniques based on principal component analysis are |

| | | | | |
|---|---|---|---|---|
| | | | | applied to remove irrelevant and redundant features. After that, feature subset selection techniques based on genetic algorithms are used to choose the most appropriate subset of features for prediction. Experiments on synthetic and real-world data show that Mlnb achieves comparable performance to other well-established multi-label learning algorithms. |
| 6 | Detecting Spam Zombies By Monitoring Outgoing Messages | Birru Devender, Korra Srinivas, Ch.Tulasi Ratna Mani | compromised machines in a network that are used for sending spam messages, spam zombie detection system, Sequential Probability Ratio Test (SPRT) | In this paper, They developed an effective spam zombie detection system named SPOT by monitoring outgoing messages in a network. SPOT was designed based on a simple and powerful statistical tool named Sequential Probability Ratio Test to detect the compromised machines that are involved in the spamming activities. SPOT has surpassed both the false positive and false negative error rates. It also minimizes the number of required observations to detect a spam zombie. In addition, They also showed that SPOT outperforms two other detection algorithms based on the number and percentage of spam messages sent by an internal machine, respectively. The main usage of the application is sender can identify the sending mails as either spam or not and weather his system is compromised system or an uncompromised one and the user defined thresholds algorithms which are CT and PT can support the dynamic behavior to detect the spam mails associated with different address locations. |
| 7 | Spam detection using text clustering | H. Shinnou, M. Sasaki | spam detection, text clustering, vector space model, centroid vectors, Ling-Spam test collection | They propose a new spam detection technique using the text clustering based on vector space model. Their method computes disjoint clusters |

| | | | | |
|---|---|---|---|---|
| | | | | automatically using a spherical k-means algorithm for all spam/non-spam mails and obtains centroid vectors of the clusters for extracting the cluster description. For each centroid vectors, the label (`spam' or `non-spam') is assigned by calculating the number of spam email in the cluster. When new mail arrives, the cosine similarity between the new mail vector and centroid vector is calculated. Finally, the label of the most relevant cluster is assigned to the new mail. By using their method, they can extract many kinds of topics in spam/non-spam email and detect the spam email efficiently. In this paper, they describe the our spam detection system and show the result of our experiments using the Ling-Spam test collection |
| 8 | Analyzing and Detecting Review Spam | Nitin Jindal, Bing Liu | product review spam detection, product reviews opinion mining, Web page spam, email spam, spam review categorization | In this paper, they study this issue in the context of product reviews.They will see that review spam is quite different from Web page spam and email spam, and thus requires different detection techniques. Based on the analysis of 5.8 million reviews and 2.14 million reviewers from amazon.com, they show that review spam is widespread. In this paper, they first present a categorization of spam reviews and then propose several techniques to detect them. |
| 9 | SMS Spam Detection Using Noncontent Features | Jieping Zhong, Jiachun Du, Qiang Yang, Evan Wei Xiang, Qian Xu | Support vector machines, Feature extraction, Classification algorithms, Electronic mail, Telecommunications, Short message services, Unsolicited electronic mail, Short Message Service, spam detection, SMS spam, social media spam, data | Short Message Service text messages are indispensable, but they face a serious problem from spamming. This service-side solution uses graph data mining to distinguish spammers from non spammers and detect spam without checking a message's contents. |

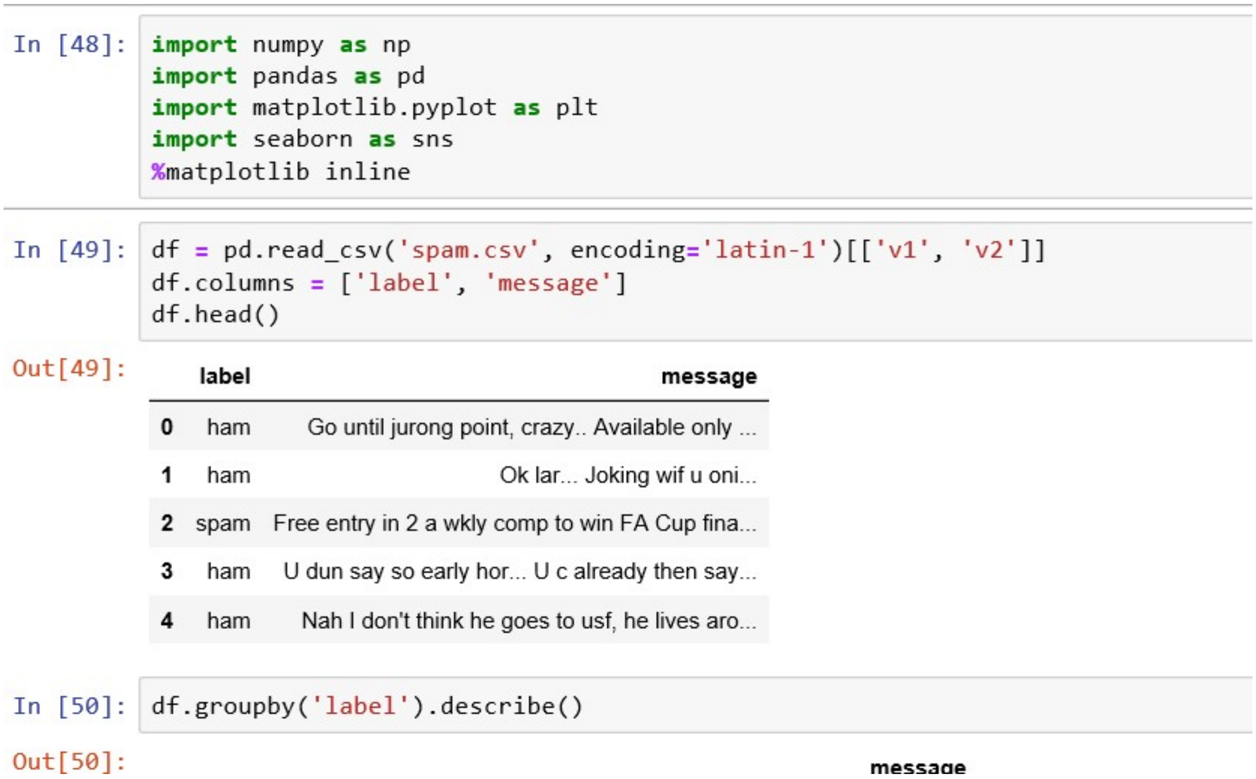
| | | | | |
|---|---|---|---|---|
| | | | mining | |
| 10 | 6 million spam tweets: A large ground truth for timely Twitter spam detection | Wanlei Zhou, Yang Xiang, Xiao Chen, Jun Zhang, Chao Chen | Twitter spam detection, spam Tweets, Google SafeBrowsing, online social networks, machine learning algorithm | To carry out a thorough evaluation, they collected a large dataset of over 600 million public tweets. They further labelled around 6.5 million spam tweets and extracted 12 light-weight features, which can be used for online detection. In addition, they have conducted a number of experiments on six machine learning algorithms under various conditions to better understand their effectiveness and weakness for timely Twitter spam detection. They will make our labelled dataset for researchers who are interested in validating or extending their work. |

## Data set description (with example given in table form)

A collection of 425 SMS spam messages was manually extracted from the Grumble text Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages. The Grumble text Web site is: http://www.grumbletext.co.uk/. -> A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available. The NUS SMS Corpus is available at: http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/. -> A list of 450 SMS ham messages collected from Caroline Tag's PhD Thesis.

|  | V1<br>Class | V2<br>messages |  |  |  |
|---|---|---|---|---|---|
|  | ham<br>87%<br>spam<br>13% | 5169<br>unique values | 43<br>unique values | 10<br>unique values | 5<br>unique values |
| 1 | ham | Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat... |  |  |  |
| 2 | ham | Ok lar... Joking wif u oni... |  |  |  |
| 3 | spam | Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's |  |  |  |
| 4 | ham | U dun say so early hor... U c already then say... |  |  |  |
| 5 | ham | Nah I don't think he goes to usf, he lives around here though |  |  |  |
| 6 | spam | FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, �1.50 to rcv |  |  |  |
| 7 | ham | Even my brother is not like to speak with me. They treat me like aids patent. |  |  |  |

# Code



In [48]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [49]:
```
df = pd.read_csv('spam.csv', encoding='latin-1')[['v1', 'v2']]
df.columns = ['label', 'message']
df.head()
```

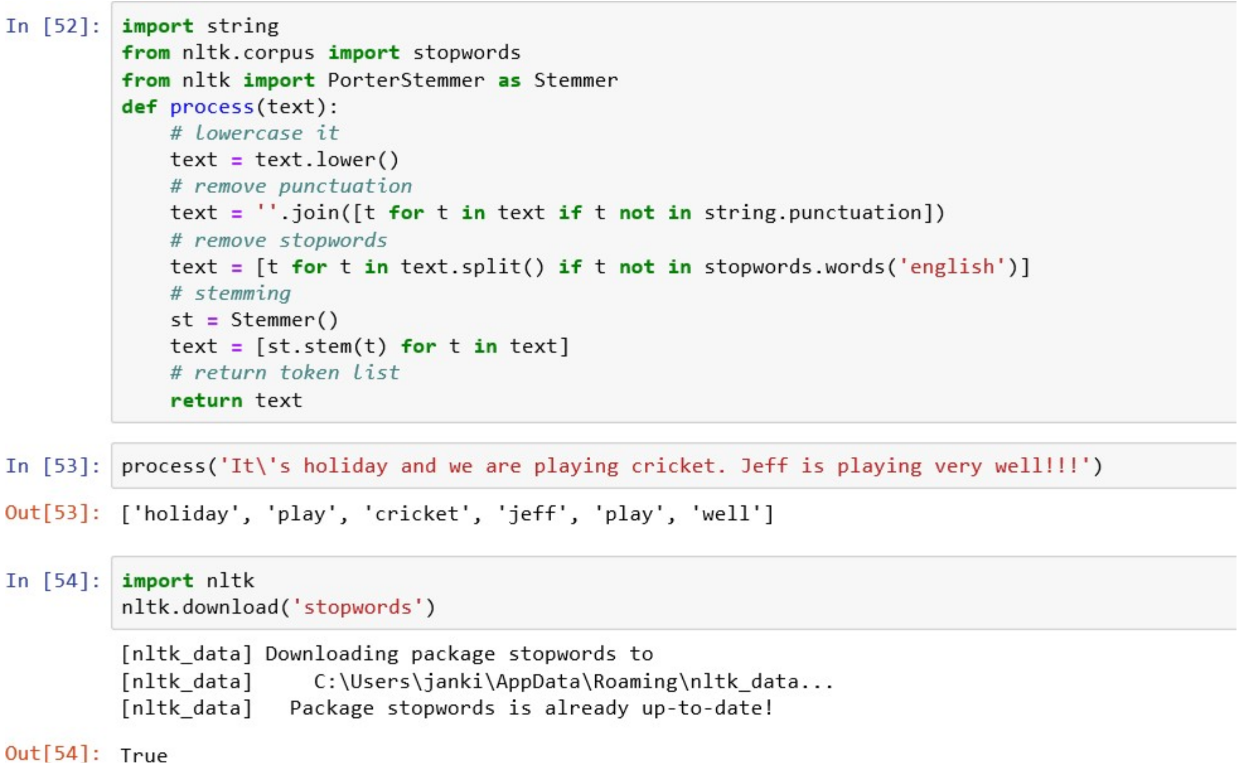
Out[49]:

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

In [50]:
```
df.groupby('label').describe()
```

Out[50]:

message



In [52]:
```
import string
from nltk.corpus import stopwords
from nltk import PorterStemmer as Stemmer
def process(text):
    # lowercase it
    text = text.lower()
    # remove punctuation
    text = ''.join([t for t in text if t not in string.punctuation])
    # remove stopwords
    text = [t for t in text.split() if t not in stopwords.words('english')]
    # stemming
    st = Stemmer()
    text = [st.stem(t) for t in text]
    # return token list
    return text
```

In [53]:
```
process('It\'s holiday and we are playing cricket. Jeff is playing very well!!!')
```

Out[53]:
```
['holiday', 'play', 'cricket', 'jeff', 'play', 'well']
```

In [54]:
```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\janki\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[54]:
```
True
```

```
In [55]: df['message'][:20].apply(process)
```

```
Out [ 55] : 0      [go, jurong, point, crazi, avail, bugi, n, gre...
            1                          [ok, lar, joke, wi*, u, oni]
            2     [free, eutri, 2, wkli, comp, win, fa, cup, fiu...
            3         [u, duu, say, earli, hor, u, c, alreadi, says
            4     [uah, dout, think, goe, usf, live, around, tho...
            5     [freemsg, hey, dari, 3, week, word, back, id, ...
            6     [even, brother, like, speak, treat, like, aid,...
            7     [per, request, mell, mell, oru, minnaminungint...
            8     [u1n ner, va1 u, network, c ustarn, sele ct, rece1v...
            9     [nob i1, 1I, month, u, r, ent it I, updat, 1at est ...
            16 [im, gonna, home, soon, dout, want, talk, stuf...
            11 [ s1x, chanc , min, c ash, 100, 26068, pound, txt ...

            13    [ive, search, right, word, thank, breather, p•...
            14                                   [date, sunday]
            15    [xxxmobilemovieclub, use, credit, click, wap,
            16                                 [oh, kim, watchj
            17    [eh, u, rememb, 2, spell, uame, ye, v, naughti...
            18    [fine, thatéo, way, u, *eel, thatéo, way, gota...
            19    [euglaud, v, macedouia, dout, miss, goalsteam,...
            Name: message, dtyoe: object
```

```
In [56a: from sk1earn.feature extraction.text import TfidfVectorizer
```

```
I n [ 57a : t£idTv  =  If  idfVect O ri  zer'(anaJy ze r=p rocess )
            data = tTid*v.fit transform(df['message'])
```

```
o int (rn e ss )
```

F r'ee entry i n 2 a wkly comp to min FA Cup I inal Ikts 21st May 2B05. *Tech* FA to 87't 21 to receive ent°y question(st d txt rate)T& C' s apply 08452816075ove rJ8 's

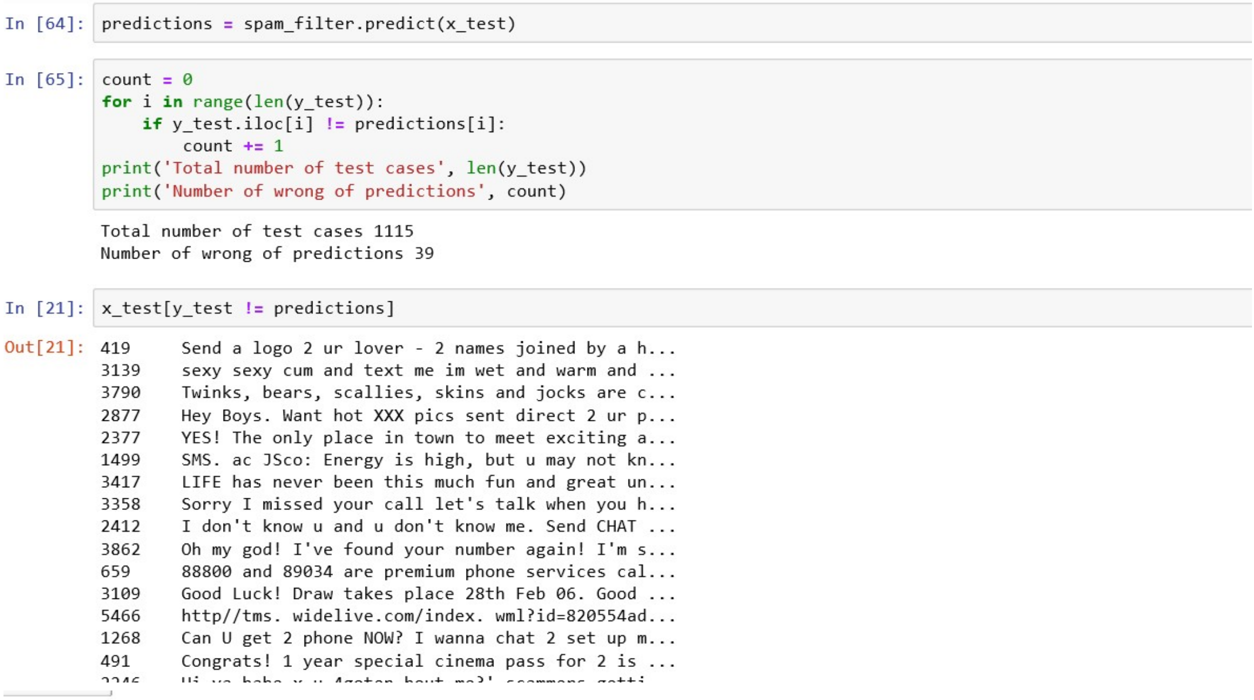
```
I n  [61] :  from skIearn.pipeline import Pipeline
             +- rom s klea rn. naive bayes Import Nu IU norma INB
             spam filter = Pipeline([
                  ( " we ci o ri zeo ' , Tf i d fVe ctor i ze ( ana Ip zen=pro ces s ) ) # wse soqes to wei ghbed Y F-PDF s core
                  ( " c las s if ten ' ,   Nu lt  i nomia INB () )                   # tro i n or in ar vectors ivi lb Eni ve 8eyes
```

```
I n  [ 62]:  from skIearn.model selection *mport train test split
             x train. x test, y train, y test = train test split(df['message'], df['label'], test size=0.20, random state = 21)
```

```
I n  [ 63]:  spam filter.fit(x train, y train)
```

```
Out [ 63] :  Pipeline(memory=None,
                   steps=[('vecto izer', TfidfVectorizer(analyzer=<function process at 0x092A6780>, biuary=False,
                      decode erro ='strict', dtype=‹class 'numpy.int64'>,
                      encoding='utf—8', input='content', lowercase=True, max df=1.0,
                      max features=None, min df=], ngram •ange=(1, 1), no m='l2',
                     .       vocabulary=None)), ('classi*ier', MultiuomialNB(alpha=l.0, class prior=Woue, fit prior=True))])
```

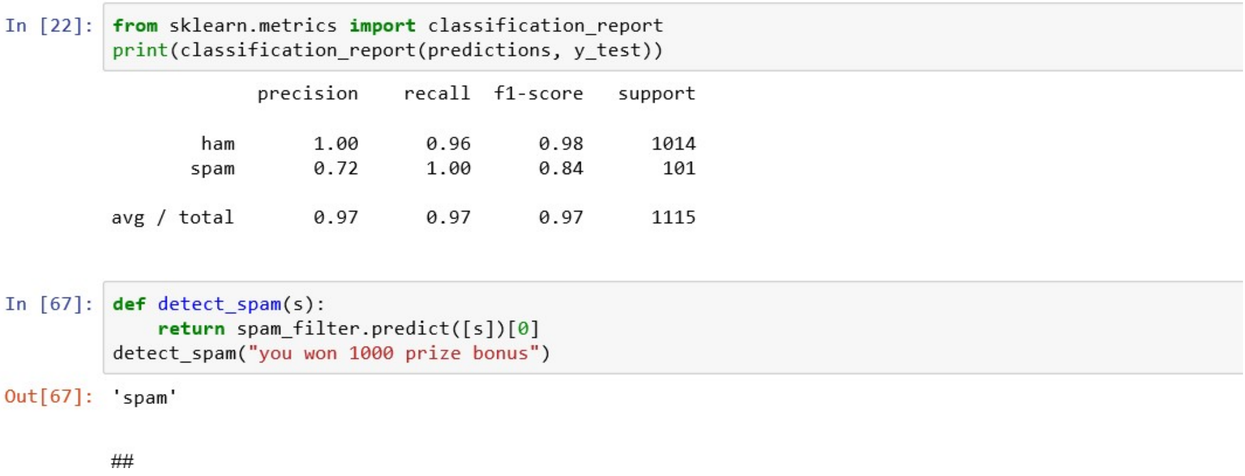
```
I n  [ 64]:  o redi ct io n s = s pan *i lt e r . p re di c I (x be st )
```



```
In [64]: predictions = spam_filter.predict(x_test)
```

```
In [65]: count = 0
         for i in range(len(y_test)):
             if y_test.iloc[i] != predictions[i]:
                 count += 1
         print('Total number of test cases', len(y_test))
         print('Number of wrong of predictions', count)

         Total number of test cases 1115
         Number of wrong of predictions 39
```

```
In [21]: x_test[y_test != predictions]

Out[21]: 419     Send a logo 2 ur lover - 2 names joined by a h...
         3139    sexy sexy cum and text me im wet and warm and ...
         3790    Twinks, bears, scallies, skins and jocks are c...
         2877    Hey Boys. Want hot XXX pics sent direct 2 ur p...
         2377    YES! The only place in town to meet exciting a...
         1499    SMS. ac JSco: Energy is high, but u may not kn...
         3417    LIFE has never been this much fun and great un...
         3358    Sorry I missed your call let's talk when you h...
         2412    I don't know u and u don't know me. Send CHAT ...
         3862    Oh my god! I've found your number again! I'm s...
         659     88800 and 89034 are premium phone services cal...
         3109    Good Luck! Draw takes place 28th Feb 06. Good ...
         5466    http//tms. widelive.com/index. wml?id=820554ad...
         1268    Can U get 2 phone NOW? I wanna chat 2 set up m...
         491     Congrats! 1 year special cinema pass for 2 is ...
         2246    Hi ya babe x u 4goten bout me?' scammers getti
```

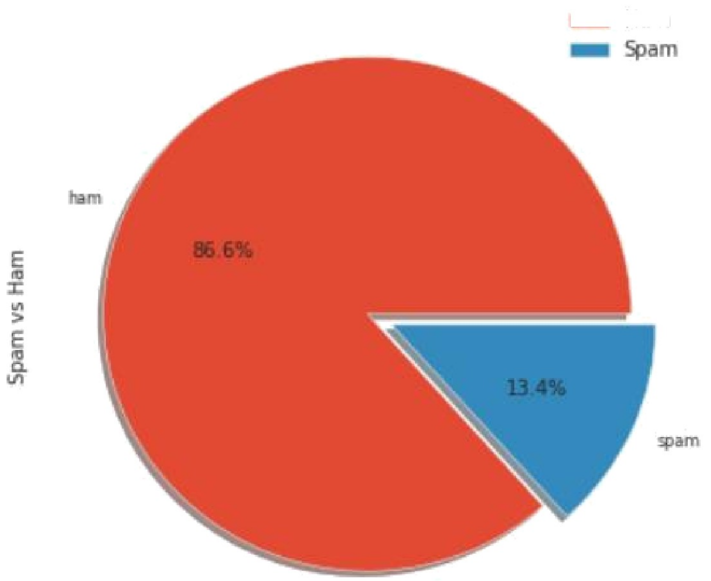


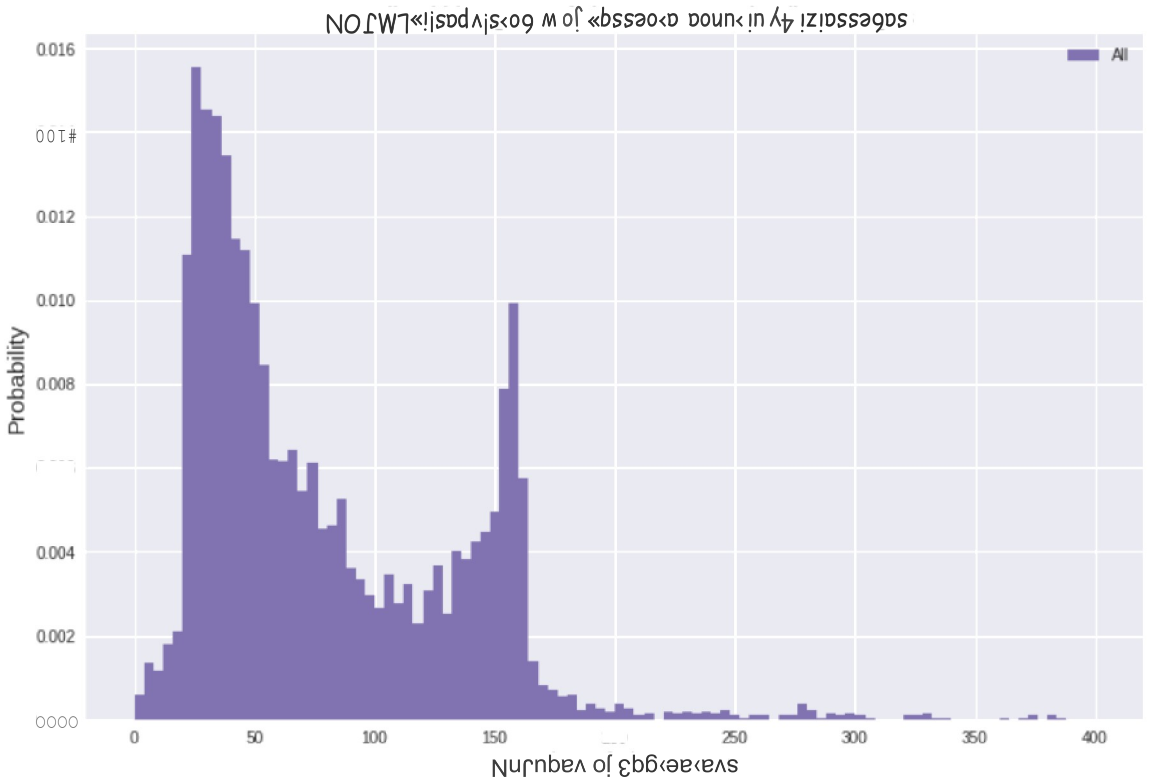
```
In [22]: from sklearn.metrics import classification_report
         print(classification_report(predictions, y_test))

                      precision    recall  f1-score   support

                 ham       1.00      0.96      0.98      1014
                spam       0.72      1.00      0.84       101

         avg / total       0.97      0.97      0.97      1115
```

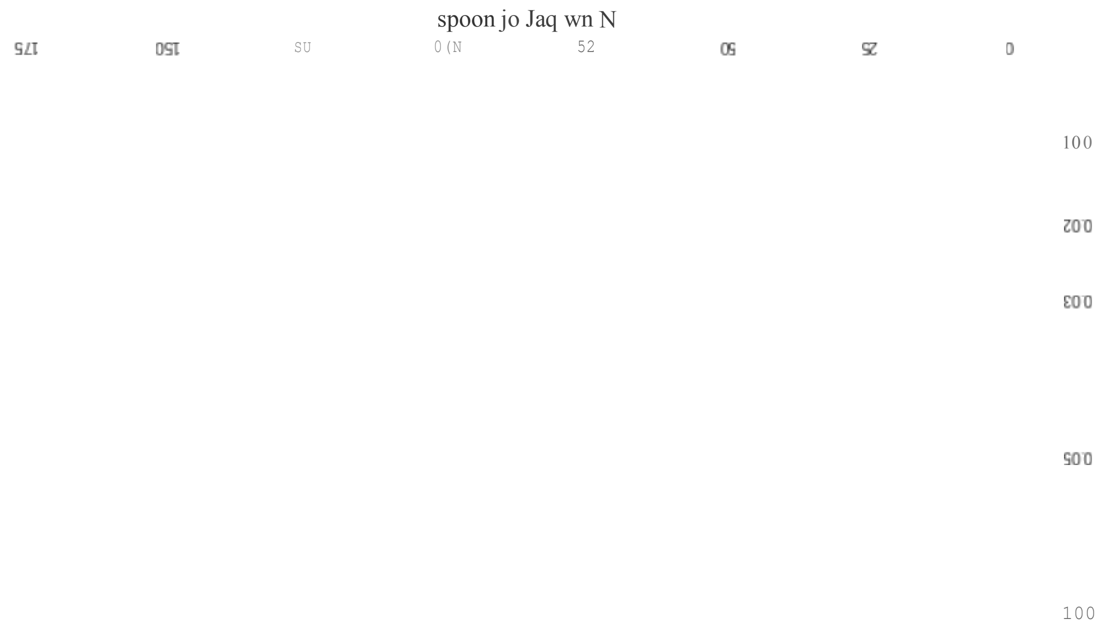
```
In [67]: def detect_spam(s):
             return spam_filter.predict([s])[0]
         detect_spam("you won 1000 prize bonus")

Out[67]: 'spam'

         ##
```

# Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
category    5572 non-null object
text        5572 non-null object
dtypes: object(2)
memory usage: 43.6+ KB
```

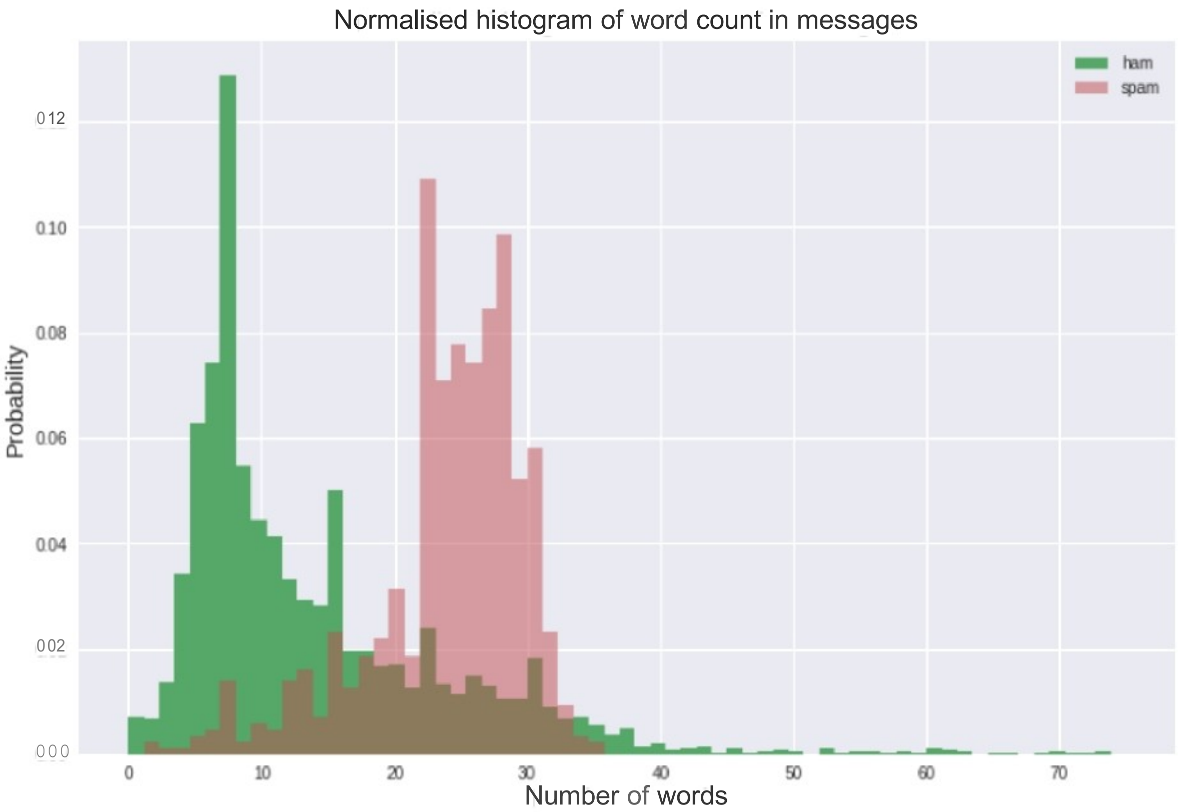

Spam
Spam vs Ham
ham
86.6%
13.4%
spam

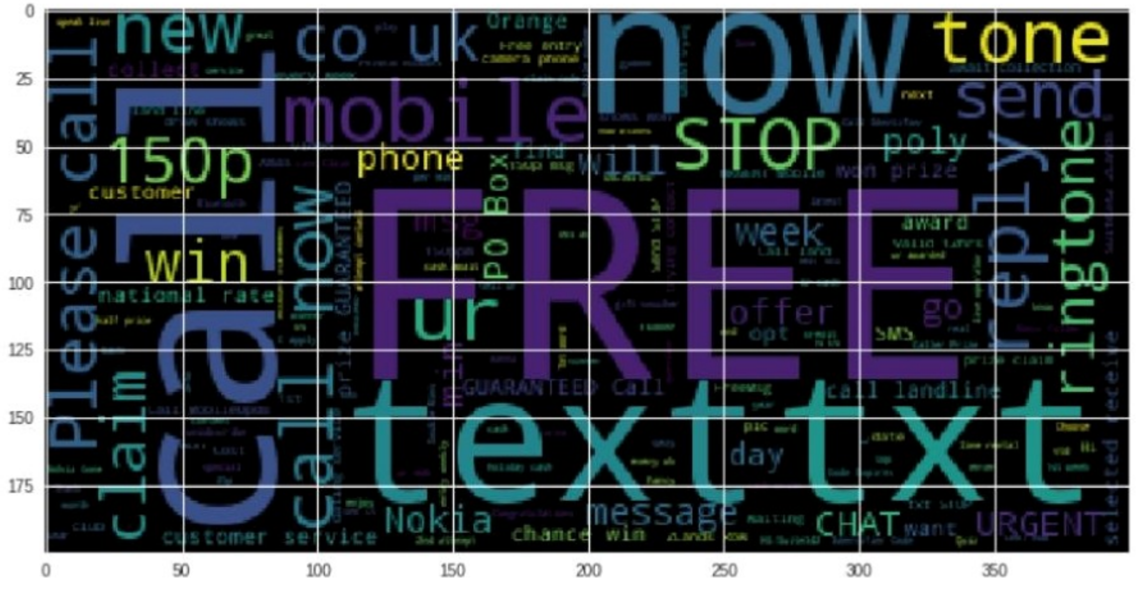

sa6essaizi 4y ui ›unoa a›oessq» jo w 6o›s!vpas!i»LMJON
All
0.016
#100
0.012
0.010
0.008
0.004
0.002
0000
Probability
0
50
100
150
250
300
350
400
sva›ae›gq3 jo vaquJnN

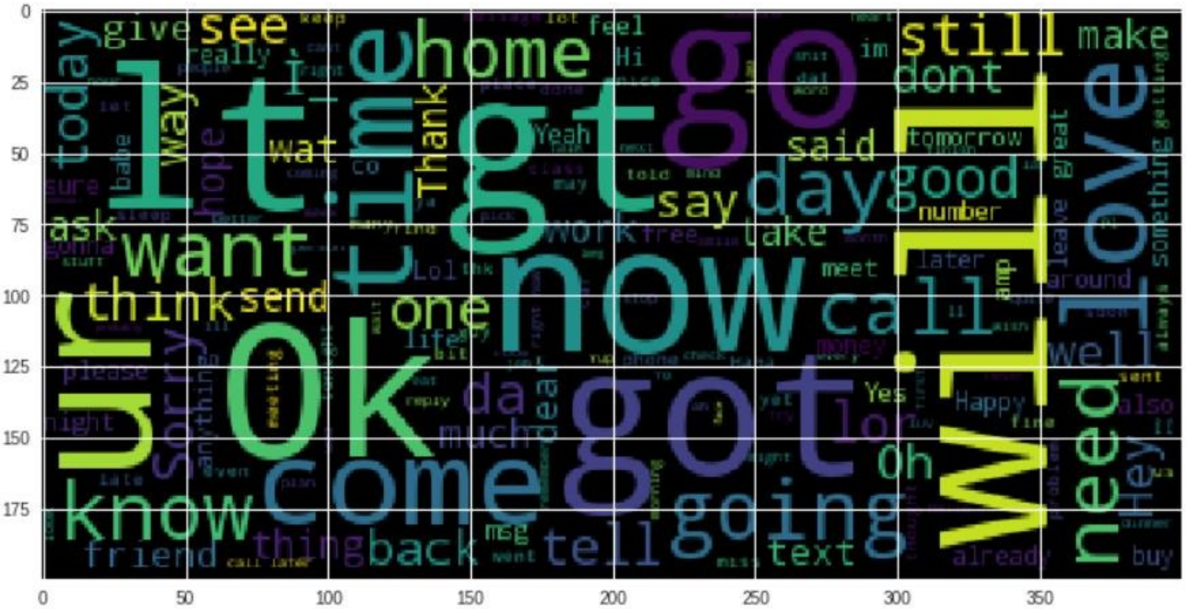


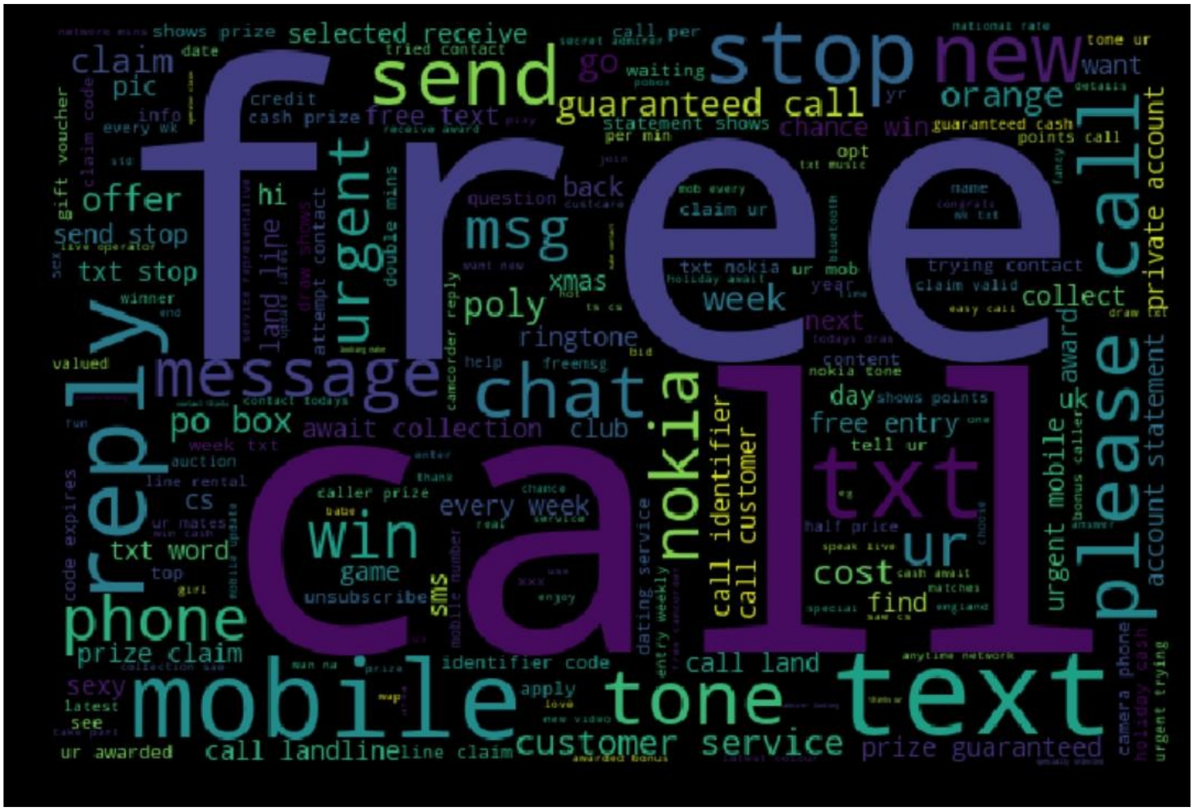
sa6essaw ||E U! Junoo poor jo uJev6oi !v pa !i + UJJON

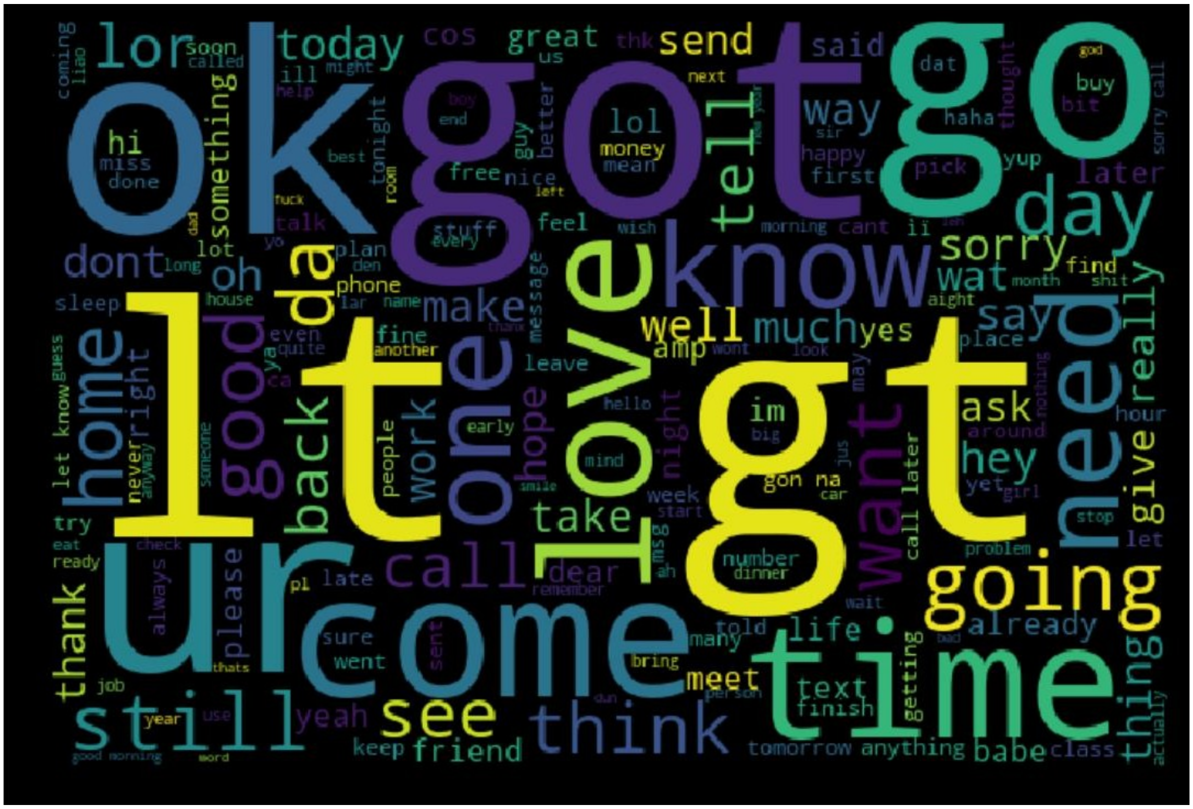
sabessaw u! ¡uno» aJ3eJeq3 jo iue BoTs! v pasi ewvoN


Normalised histogram of word count in messages
ham
spam
Probability
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
60
70
Number of words

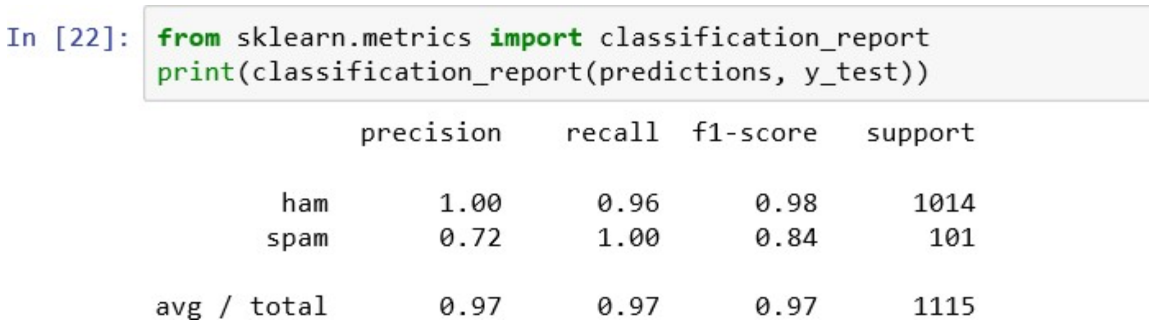

0
25
50
75
100
125
150
175
0
50
100
150
200
250
300
350
now
FREE
call
txt
text
mobile
tone
new
co uk
send
reply
STOP
150p
Please call
ringtone
win
claim
phone
Box
poly
Will
customer
week
award
offer
go
ur
call now
national rate
opt
SMS
prize claim
GUARANTEED Call
call landline
day
message
Nokia
CHAT
URGENT
customer service
chance win
want
Orange
won prize
selected receive


0
25
50
75
100
125
150
175
0
50
100
150
200
250
300
350
lt
gt
go
ok
ur
now
got
will
time
home
come
love
still
want
know
going
call
day
good
need
see
give
make
dont
today
way
said
tomorrow
say
ask
think
send
one
work
take
tell
back
text
friend
thing
sorry
Hey
Oh
well
later
meet
da
lor
much
dear
Yes
Happy
please
night
already
buy
Thank
wat
Yeah
really
feel
Hi
im
number
hope
babe
sure
around
something
great
leave
msg
life
money
free


free
call
mobile
text
txt
reply
stop
new
send
please call
tone
message
chat
nokia
urgent
msg
win
phone
claim
guaranteed call
orange
offer
customer service
prize guaranteed
call landline
selected receive
shows prize
free text
chance win
po box
await collection
poly
week
ringtone
ur
cost
find
game
every week
txt word
prize claim
call identifier
call customer
free entry
account statement
private account
urgent mobile
collect
back
hi
pic
send stop
txt stop
land line
sexy
sms
unsubscribe
identifier code
call land
apply
ur awarded
club
day
uk
award
go
want
credit
cash prize
xmas
next
latest
see
cs
top
gift voucher
valued
winner
auction
code expires
statement shows
waiting
trying contact
tried contact
opt
per min
call per
tone ur
points call
guaranteed cash
dating service
half price
shows points
tell ur
txt nokia
ur mob
claim ur
claim valid
content
double mins
attempt contact
question
year
date
info
every wk

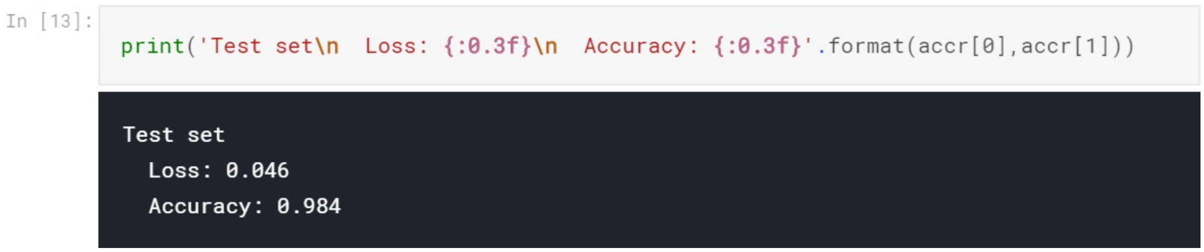


## Result and accuracy



```
In [22]: from sklearn.metrics import classification_report
         print(classification_report(predictions, y_test))
```

```
             precision    recall  f1-score   support

        ham       1.00      0.96      0.98      1014
       spam       0.72      1.00      0.84       101

avg / total       0.97      0.97      0.97      1115
```



```
In [13]: print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

```
Test set
  Loss: 0.046
  Accuracy: 0.984
```

# Conclusion

In this paper it has been shown that it is possible to achieve very good classification performance using a word-position-based variant of naive Bayes. The simplicity and low time complexity of the algorithm thus makes naive Bayes a good choice for end-user applications. This spam classifier is implemented by Naive Bayes Model, a simple but very efficient solution in spam classification problem. In brief, Naive Bayes treats every features independent from each other, making inference very efficient.

This code runs quite well with the accuracy of 98.31% on training sample and 97.81% on test sample. We also tried with Random Forest and XGBoost, but the accuracy was low, so the conclusion is that the Naïve Bayes is the best classifier till now.

# References

https://ieeexplore.ieee.org/abstract/document/4403192

https://ieeexplore.ieee.org/abstract/document/6246947

https://ieeexplore.ieee.org/abstract/document/7393866

https://www.sciencedirect.com/science/article/pii/S0957417408003564

https://www.sciencedirect.com/science/article/pii/S0020025509002552

http://www.irjes.com/Papers/vol5-issue5/D-71-75.pdf

https://ieeexplore.ieee.org/abstract/document/1587549

https://ieeexplore.ieee.org/abstract/document/4470288

https://ieeexplore.ieee.org/abstract/document/6133257

https://ieeexplore.ieee.org/abstract/document/7249453