TF-IDF classification based Multinomial Naïve Bayes model for spam filtering

MSc Internship
MSc Cybersecurity

Alan Chavez
Student ID: x19137516@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Alan Chavez ……………………………………………………………………………………

Student ID: ……………x19137516…………………………………………………………………………………..……

Programme: ………… MSc Cybersecurity………… **Year:** …………………………..

Module: ………………Internship……………………………………………………………………….………

Supervisor: ……………Niall Heffernan…………………………………………………………………….………

Submission Due Date: ………………17 of August 2020……………………………………………………….………

Project Title: …………… Text frequency classification based Multinomial Naïve Bayes model for spam filtering ……………………………………………………………….…

Word Count: …………4809……..……… **Page Count**……………11………………………………..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: …………………………Alan Chavez ………………………………………………………

Date: …………………………17/08/2020…………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

TF-IDF classification based Multinomial Naïve Bayes model for spam filtering

Alan Chavez
MSc Cybersecurity
National College of Ireland
Dublin, Ireland
X19137516@student.ncirl.ie

1 Abstract

Email spam, better known as unwanted email messages, is the practice of sending unsolicited electronic messages with different intentions, commonly commercial purposes, or trying to commit criminal actions. Despite the numerous anti-spam measures nowadays, spam still being a problem all over the internet due to the low-cost and high impact that represents elaborate a spam campaign. Many different solutions exist to categorize incoming messages such as white list, grey list, blacklist, Machine Learning, Rule-based filtering, etc. However, no one definitively. A possible reason is since spammers are high resilient, once a spam filtering method is compromised spammers adapt to it. The aim of the present work has the objective of detecting in a more effective way spam email with the Multinomial Naïve Bayes approach, in addition to text sanitation and TF-IDF. Results given by the proposed model gives an accuracy improve than Multinomial Naïve Bayes by its own.

2 Introduction

The high volume of spam traveling through the internet has reached unmeasurable proportions lately. According to a securelist[1] in the Q1-2014 the calculate spam on the internet was 63.5%, in February Q1-2019[2] it drops to 55% maintaining the label as a serious threat for the internet security and their users.

Virus and spamming go along the way, a virus is a malicious payload designed to attack any possible flaw in the system having different purposes from data collection to denial of service and everything in between. Furthermore, Spam represents a problem for the ISP (Internet Service Provider) and final users.

Spam messages and spam controls have evolved over the last decade. Different techniques and methods have been applied to try to face spam such as white, grey, and blacklist [3]. However, different approaches published in academic works seems to be limited to facing spam messages. The principal reason is due to the techniques disclosed. Once the technique is revelated, spammers tend to adapt it and make it obsolete. There is not a bulletproof method to stop junk mail, and that is why the purpose of the present paper is the analysis of spam filters, explain the history and background to understand the origin, and why represents a threat. These different perspectives and discussions lead to the challenge of refining an automatic classifier that can identify illegitimate emails from unwanted sources more efficiently and accurately.

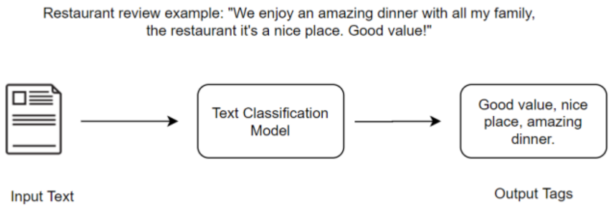
Technically, an electronic message is spam if the message is irrelevant and out of context to the recipients because it may apply to many other people, or if the recipients have not granted any explicit permission to receive it [23]. Others simply describe spam as a junk email, from a bulk message sent through the internet [24]. The relation between "SPAM" the canned meat and unsolicited mails is not all clear, different resources mention due to a can met sketch where a group of Vikings shouts "Spam, Spam, Spam" repeatedly, all over the restaurant catching all the attention and "blocking" all kind of communication. According to Quigley R. [25] the first spam electronic message was from 1978, Quigley refers to Gary Thuerk a marketer for the Digital Equipment as the responsible for sending the first-ever spam mail. Sending out his message to 400 of the 2600 ARPAnet. In addition to some researches, on the internet is possible to find information about the first spam ever. The first spam message was transmitter over telegraph wires in 1864, curiously some advertisement was involved, the full text can be seen in Appendix A.1.

2.1 Spam recognition techniques

To face the challenging threat, different commercial and open source products exist in the market to satisfy different users' necessities. The basic and more used spam filters are based on the most used spammers' phrases and words. Other simple approach cover whitelist, blacklist, and gray list, a new approach where the filter assumes the spammer attempt to send a batch of spam messages just once, the receiving mail rejects the unknown users' email sending a failure message to the origin server and just if the email is received again the gray list filter will add the sender's email and allow emails from that user [29].

Different approaches fit different users. In the first approach, the flaw relies on the assumption that the spammers will act like spammers and they will not try something different like changing the vocabulary to avoid the filters, or hiding their messages in pictures. The Blacklist problem is the fact that you must know all the harmful email senders to perform full protection. On the other hand, whitelist risk is the overlook of a relevant message if the sender is not a previous register as part of the trusted sources.

Text classification is the duty of assigning a set of predefined values, weights, or categories to a text. Text classifiers are often used to categorized, organized, or structure huge amounts of text. The classification is done in two different ways: automatic and manual classification. The manual classification is done by a human who interprets the content of a text and classified them already predefined. The method means high accuracy but it is time-consuming and expensive if the text is a big sample. To evolve into a faster and less cost-consumer way, natural language processing and machine learning form part of the automatic classification option.


Restaurant review example: "We enjoy an amazing dinner with all my family, the restaurant it's a nice place. Good value!"
Text Classification Model
Good value, nice place, amazing dinner.
Input Text
Output Tags

Different spam filter techniques are explained in this paper. However, it is important to highlight the relevance of the state-of-the-art spam filter, explaining their different approaches and the gap they are trying to control the spam problem.

- **Case Base Spam Filter Method:** Case base filtering probably the most used method. Approach where the classification is based on the content of itself. Method where all received emails are collected from the users' inbox, spam, and non-spam. After the collection, emails are processed to extract and evaluate their data. Then, the data is classified into two vector sets. Finally, the machine learning algorithm uses the datasets obtained before to train and decide if the incoming email is non-spam or spam [4].
- **Content-Based Filtering Technique:** Content-based filtering is used to create automatic filtering based on rules, classifying emails using machine learning, such as Support Vector Machine (SPV) [13], Artificial Neural Networks (ANN) [9], Naïve Bayesian [19]. This method analyses the words features, frequency, and distribution of phrases and the email words based on the generated rules before, to classify all incoming emails [5].
- **Rule-Based Spam Filtering Technique:** Ruled based spam filter use rules to evaluate different patterns based on the common expressions on each email. Every time a pattern is identified within an email the email score increased, and for every time the pattern did not match, the score is deducted. The email is considered unwanted once the message reaches the trigger value. On the bright side, the rules can be modified according to the necessities. On the downside, spammers easily adapt introducing new messages to avoid filtering rules [6].
- **The previous Likeness Based Spam Filtering Technique:** This approach used machine learning methods to categorize emails based on their data training similitude. The different email features are used to create a multi-vector, which helps to qualified new incoming messages. The new messages are subsequently categorized due to the training data and allocated into their corresponding instance [7].
- **Adaptive Spam Filtering Technique:** During this method, the incoming emails are divided into various groups represented by tokens. Tokens represent different sentences, words, or strings. The token is used to compare the similarity ration with receiving emails to decide whether the incoming email is spam or not [8].

Many academicians and researchers have already proposed different spam classification techniques approach used to classify data. Techniques such as decision tree[11], support vector machine (SVM)[9], artificial neural networks (ANN)[10], and Naïve Bayes algorithms [11]. Content-based filtering technique has been presented as the most popular filtering option dealing with spam, content-based filtering technique identify patrons in particular phrases, symbols, words, and grammar. The frequency at which these particular patrons appear in emails determines the probabilities for each characteristic in the email, after which is compared against the trigger value. If the email exceeds this measure, automatically ring the trigger value and is classified as a spam message [6]. Lately, the most recurrent topic for researchers and academics has become Naïve Bayes (NB) classifiers. The probable reasons are the high accuracy demonstrated classifies data model, and also the easy comprehension compares to other approaches. NB, Artificial Neural Network (ANN) and Support Vector Machine (SVM) share something in common, all represent Machine Learning (ML) classification option with different advantages.

Support Vector Machine (SVM) is an already tested and trusted state of the art classification technique facing unwanted mails. SVM is a discriminative classifier that takes decisions using the learned hypothesis [13]. SVM is a supervised learning model that analyses the statistical properties of text classification to identify patterns between the variables predefined [14]. The downside of SVM is that in high volumes of data the efficacy decrease. I believe SVM is a powerful state of the art option against spam, a functional tool when the separation between classes is clear. Which can decrease in effectiveness if the data set is enough big to create noise, overlapping the target classes? To face the size limitation, different academic papers claim Decision Tree (DT) a suitable option for bigger datasets.

Decision Tree (DT) is a proven machine learning algorithm approach against spam messages. Unlike SVM, decision tree (DT) can handle in a better way large datasets and require fewer data training. A decision tree (DT) is a flow chart (with a tree shape), where an internal node represents an attribute, an arm represents a decision rule, and the leaf node represents the solution or outcome [16][17]. The great DT worth value is the capacity to assign an open to interpretation value to decisions and results. On the other hand, nonparametric DT values let different interpretation which tends to overfit of training data, limiting their classification accuracy [18]. Compare to ANN, DT is a white box Machine Learning algorithm, helping to follow the decision path to its end and share the decision taken with all the teamwork. In contrast, ANN is a black box type algorithm where the algorithm is not visible or available. DT is a discriminative machine learning: learning from explicit boundaries between classes. That means, training a model to distinguish the correct output between all possible outputs. While Naïve Bayes (NB) represents a generative machine learning model where a model learns the parameters to maximize the joint probability.

Naïve Bayes (NB) algorithm is another machine learning approach widely used for email spam filtering. Applying the Bayes' theorem on text classification, each message represents a vector with attributes. Attributes binaries represented into the electronic messages [19]. No rules or weights have to be optimized[9]. NB has different benefits over other spam filtering approaches such as real-time predictions, scalability, simplicity, and good accuracy with high dimensional data [15], compared to the fewer data needed to assume and classify an object by mapping the features classifying it individually. In [22] authors mention BETSY as a real-world spam filtering example that uses NB to realize the classification. BETSY is a program that classifies text based on previously trained material. NB has many advantages and is the reason why different academics have chosen NB to elaborate spam filter studies. However, NB assumes that all inputs are independent. If the case is the opposite the accuracy of the NB classifier can be compromised. A possible reason to pick Artificial Neural Networks (ANN) over NB is that with the appropriated structure, ANN can handle the correlation and dependence between input variables.

Artificial Neural Networks (ANN) is a computational model meant to simulate the functions of a human brain [9][12]. ANN is built like the human brain, with nodes (neurons) interconnected between them. Each neuron is made up of a cell body responsible for processing data by carrying information towards and away (inputs and outputs) from the brain. The principal feature of this approach is the creation of new structures to process the information.

	Pros	Cons
Support Vector Machine	- More effective in higher dimensional spaces - Effective algorithm when classes are separated	- Not optimum performance when data has much noise - Larger datasets represent a large amount of time
Decision Tree	- Handling missing values - Easy visualization	- Overfitting - Higher time to train
Naïve Bayes	- Real-time decisions - Multi-class prediction	- Assumptions - If inputs are dependent, the outcome can be compromised
Artificial Neural	- Adaptive networks	- Not every data is suitable for

Network	- Effective recognizing patterns	ANN - The outcome quality depends on the training quality phase

3 Literature review

There is an increased interest in the worldwide enthusiasm on email spam filtering. Inside this section, different papers are presented enlisting the highlights as well as the issues found to be addressed. In the paper title “An Artificial Neural Nets for Spam e-mail Recognition” details an ANN spam filter approach where two ANN algorithm was applied, backpropagation and optical backpropagation, comparing accuracy, recall, false negative, positive and complexity. Applying attributes composed from the favorite spammer descriptive feature patterns. They conclude which ANN configuration performed in the best way with fewer errors. However, the papers date from 2006 and for obvious reason no cover recent articles [26]. A more up to date Naïve Bayesian approach is presented in [20], authors describe the general spam characteristics with a comparative table between spam characteristics and the percentage of searched messages. The Bayesian algorithm throws an 85% accurate system, which they comment is acceptable for spam filtering. But what happened if just 10% or 15% of the training samples are spam?. In conclusion, NB algorithms are popular because are easy to understand, use, and refine. If we do not forget in mind that requires frequent maintenance refining the algorithm.

Youn S and McLeod D. customized an adaptive ontology spam classification filter. Using SVM and NB to classify over 4500 emails and 55 features. The authors conclude that a filter method can evolve based on the users' background. Using a text-oriented email dataset, improved performance can be achieved. The paper highlight the opportunity to improve accuracy by pruning the tree and using a better classification algorithm [27]. A similar solution was proposed in [22] where authors mention that each user has a unique email collection, and different users would have a different email into their spam folder. Finally, they declared that highly effective filtering can be reached making the user's interface information available of the mail client. Seems an interesting filtering approach which in some cases is already in use, when a user identifies a marketing email that has the option to mark it as spam, the filter algorithm would learn from this choice preventing from receiving similar emails. Joachims T. [21] in his publication gives some proves on how SVM development classification can be applying to achieve a more accurate result than using more popular methods like Naïve Bayes (NB). Gapta S. mention in [15] that high volumes of data could represent an SVM efficacy decrease due to the complexities of the processed data. In contrast to Sanghani G, Dr. Kotecha k. [13] who declare that SVM can reach a high performance despite a big dataset due to the learning procedure which narrows the decisions between classifications.

Wang R, Youssef A, Elhakeem A. in [28] present two heuristic algorithms seeking the spam filtering improvement. Using Artificial Immune System (AIS) and Tabu Search (TS). The proposed TS algorithms reduced the email dimension and notable improvement with the 94.5% of effectiveness. However, heuristic filters needed to have often maintenance updating rules to keep it up to date which means time-consuming.

4 Research methodology

Electronic messages generally look different from the solicited to the unexpected mails. However, spammers have found methods to avoid spam filters, delivering illegitimate emails pretending to be legitimate emails. That is the reason why I decided to elaborate my spam filtering approach with Naïve Bayes, instead of a method as decision tree which decides if an incoming email is spam based on rules, rules that can be easily avoided if are known by the spammer. The Naïve Bayes approach unlike rule-based filtering fragments the incoming messages in a bag of words to then eliminate the noisy words and then elaborate comparison between the new incoming text and the already known spam words. Facilitating a matrix word visualization and elaborating a frequency map useful in futures tasks and feedback analysis.

The uncomplicated Naïve Bayes algorithm is used to find association words related to spam from the training emails. Based on these association words, a five-phase spam filtering method is explained below:

Checking on each emails' words as a solution to prevent illegitimate messages from crossing the spam filter is efficient, but realize the task manually would represent a high energy and time consumption. Taking advantage of the ML features and to automate the process, is necessary to transform the previously classified label-data into binary data which the NB algorithm can read to make the categorization faster and efficient.

A mechanism is developed to identify all incoming words from the text (emails), converting a collection of words into a matrix. Each message is represented by a row and each word (token) being a column, the row-column values represent the frequency of the occurrence of each word or token in the message. After the tokenization, Bag of Words is applied to have the measure of all words included in the messages regardless of their position.

Before the vectorization, the data is sanitized, eliminating noisy and repetitive words. Reducing the dataset enough to let the model focus in the relevant text features.

The *CountVectorizer* gets the frequency of the words in the text.

TF-IDF provides the weighting to the text. Assigning the importance based on the appearance in the message. Also, it checks how relevant the word is for the corpus.

Finally, A multinomial (MB) training model is designed to improve the method. MB requires discrete features (in our case, word counts for text classification). We trained our model with the training data (80% of the total dataset, already split) and the final step is the evaluation of the 20% left data to make the predictions in tenfold cross-validation representing a real-case spam scenario. Therefore, the additional feedback from the analysis of misjudging emails can potentially improve the method to identify spams.

5 The proposed Spam filtering approach / Design specification

The proposed spam filtering approach categorizes emails into legitimate emails or spam messages with NB help. The approach is briefly provided below:

The information source that feeds the proposed NB spam filter is the “SMS Spam Collection Data Set" available in the Machine Learning Repository [30]. The dataset contains 5574 samples, 4825 non-spam messages called "ham", and 749 messages labeled as "spam". The data is separated into two columns, the first one categorizing the email between "spam" or "ham", and the second column contains the message text to be classified.

- Database pre-processing: Since Scikt-learn (Python library used for classification) manages numerical values, Scikt-learn transforms the strings “spam” and “ham” values to binary values “0” and “1”.

- Tokenization: is the process where sentences, phrases or a paragraph are split into individual words. Each one of the words is called a token

- Bag of Words: Bag of Words (BoW) is used to extract features from text to be used in ML algorithms. BoW involves a collection of known words and a measure for the known words. TF-IDF (term frequency–inverse document frequency)providing more weight understanding the context giving more weight to the words that appear more in the set of data.

- Word standardization: Reducing inflectional forms, derivationally related forms, low case standardization, stop words parameter. All steps mentioned before help to regularized words after BoW to avoid common English words, ignore punctuation, etc.

- Multinomial (MB) Naïve Bayes training model: Ideal training model since the classifier is suitable for classification with discrete features (words in a text)

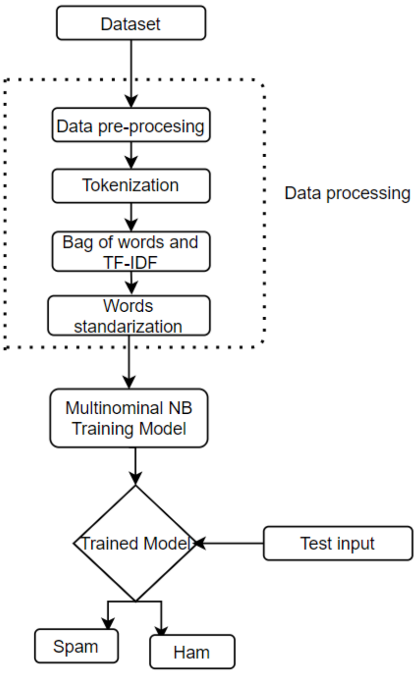


Fig 1. Project Model

5.1 Naïve Bayesian Theorem

Naïve Bayesian theorem classifier assumes that all the predictors in the provided dataset are independent of each other, that is the reason why is called “naïve”. The formula for the conditional probability is:

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

Where:

- P(A): Is the probability that the hypothesis “A” is true (regardless of the data). This is known as the prior probability of “A”.
- P(B): Data probability (regardless of the data). This is known as the prior probability.
- P(A|B): Is the probability that the hypothesis “A” given the data “B”. This is known as the prior probability.
- P(A|B): is the probability of the data "A" given that hypothesis B was true. This is known as the posterior probability.

$$Posterior = \frac{likelihood * prior}{evidence}$$

In other words, the prior P(A) and the **evidence** P(B) mean to the probabilities of A and B being **independent** of each other, while the **posterior** and **likelihood** are **conditional probabilities** of A given B and the other way around.

5.2 Naïve Bayes Spam classification

$$P(Spam|x) = \frac{p(x|Spam) * P(Spam)}{P(x)}$$

Where "x" represents an incoming vector of words coming from the emails (spam or ham):

$$\dot{X} = [w1, w2, w3, w4 \ldots., wn]$$

The assumption that the NB classifier makes is that the incoming words are independent of each other. The result is that the **likelihood** is the product of individual probabilities seeing each word in the dataset spam or ham.

Likelihood, **evidence**, and **prior** represent necessary variables in the NB theorem to proceed to obtain **posterior** probabilities. Probabilities that will provide the possibilities of a new incoming email classified as spam based on a particular set of words.

Multinomial (MB) Naïve Bayes classification model

Multinomial (MB) NB is one of the three types of NB models. Multinomial (MB) is used for discrete counts. Applies when the data features have discrete frequency counts. (In our exercise we count the words in the body of the email), then is necessary to use word counts in the email's body.

Considering $\dot{X} = [w1, w2, w3, w4, \ldots \ldots, wn]$ as vector of n distinct feature in the dataset, and V = [v1,v2,v3,v4, ... Vn] express all possible classes belonging to NB Multinominal model:

$$P(vk|w1, w2, w3, w4, \ldots., wn) = \frac{P(w1, w2, w3, w4, \ldots, wn|vk) * P(vk)}{P(w1, w2, w3, w4, \ldots, wn)}$$

$P(vk|w1, w2, w3, w4, \ldots., wn)$ represent the **posterior** probability of classes, while *vk*P(vk)* express that the sample probability belongs to class “vk”. **Prior** probability is represented by $P(w1, w2, w3, w4, \ldots, wn)$which is a constant. $P(vk|w1, w2, w3, w4, \ldots., wn)$ express the **likelihood** for$\dot{X} = [w1, w2, w3, w4, \ldots \ldots., wn]$ to belong to category *vk*.

The multinomial algorithm needs to process two possible questions to be able to classify new messages:

$$P(Spam|w1, w2, w3, \ldots wn) \infty P(spam) * \pi \prod_{i=1}^{n} P\,(wi|Spam)$$

$$P(Ham|w1, w2, w3, \ldots wn) \infty P(Ham) * \pi \prod_{i=1}^{n} P\,(wi|Ham)$$

Where

$$P(wi|Spam) = \frac{Nwi|Spam + \alpha}{NSpam + \alpha * Nvolabulary}$$

$$P(wi|Ham) = \frac{Nwi|Ham + \alpha}{NHam + \alpha * Nvolabulary}$$

- Nwi represents the frequency in the sample category w. Nwi is the total of all features counts into class w.
- α is smoothing prior and equal to 1, it is called Laplace smoothing and helps to prevent zero probabilities.
- n is the total of different features in the training dataset.

It is important to mention:

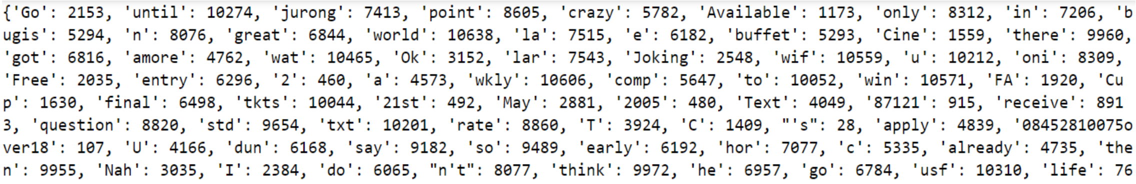
- NSpam represents the total number of words in all the spam messages, not the number of spam messages or not equal to the total number of unique words in spam messages.
- NHam represents the total number of words in all the non-spa messages, not the numbers of non-spam messages or the number of unique words identified in non-spam messages.

6 Evaluation

Python 3.7 was used for experimentation. Scikit-learn to avoid the NB math implementation from scratch and the Multinomial NB algorithm (explained above) as a training model due to the data features handle in the paper.

The algorithm was tested over a corpse of messages categorized as "ham" and "spam". The goal is to demonstrate the high levels of accuracy that can be reached based on text classification. Using the ham and spam dataset, an ML model (NB) is trained to learn to categorized spam or ham emails automatically. A model which later, can be used to classify new incoming messages (unlabelled).

The dataset contained 4827 ham and 747 email messages. Though the data processing the messages are split into individual words to be tokenized. The tokenized data is converted into vectors to allow ML to deal with them. During the vectorization, BoW transforms the vectors counting how many times does a word appears in each message, assigns the weighting, and normalizes vectors.

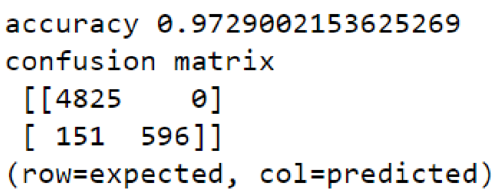


```
{'Go': 2153, 'until': 10274, 'jurong': 7413, 'point': 8605, 'crazy': 5782, 'Available': 1173, 'only': 8312, 'in': 7206, 'b
ugis': 5294, 'n': 8076, 'great': 6844, 'world': 10638, 'la': 7515, 'e': 6182, 'buffet': 5293, 'Cine': 1559, 'there': 9960,
'got': 6816, 'amore': 4762, 'wat': 10465, 'Ok': 3152, 'lar': 7543, 'Joking': 2548, 'wif': 10559, 'u': 10212, 'oni': 8309,
'Free': 2035, 'entry': 6296, '2': 460, 'a': 4573, 'wkly': 10606, 'comp': 5647, 'to': 10052, 'win': 10571, 'FA': 1920, 'Cu
p': 1630, 'final': 6498, 'tkts': 10044, '21st': 492, 'May': 2881, '2005': 480, 'Text': 4049, '87121': 915, 'receive': 891
3, 'question': 8820, 'std': 9654, 'txt': 10201, 'rate': 8860, 'T': 3924, 'C': 1409, "'s": 28, 'apply': 4839, '08452810075o
ver18': 107, 'U': 4166, 'dun': 6168, 'say': 9182, 'so': 9489, 'early': 6192, 'hor': 7077, 'c': 5335, 'already': 4735, 'the
n': 9955, 'Nah': 3035, 'I': 2384, 'do': 6065, "n't": 8077, 'think': 9972, 'he': 6957, 'go': 6784, 'usf': 10310, 'life': 76
```

Fig2. BoW count

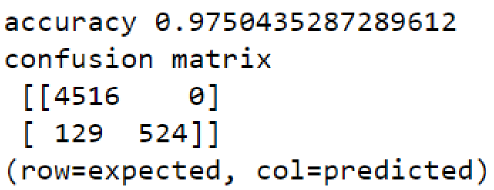
Some authors (Řehůřek R, 2014)[31] and (Lara, 2019)[32] have found value using the NB theorem deploying spam filters. Lara deploys a four phases model using BoW as the spearhead to collect all words contained within the message and, thereby, understand the messages' context. Laras' approach claims to reach an accuracy of over 98% evaluating on the same training data. However, as Řehůřek mention in his publication is not reliable to evaluate the accuracy on the same data. Řehůřek's approach has four steps, highlighting the training model, where a 10 equally size subsets are taken to compute the accuracy. Then the data is trained on 9 parts, and compute the accuracy on the last part.

Due the analysis that I have been done, I suggest changes in the way how text is managed before the tokenization. Implementing a set of conditions, ensuring the sanitization of the text that would impact positively on the words before the training model. The analysis In the next section is followed by the Řehůřek [31] proposed and compared against my Proposed Model. Highlighting the differences taking advantage from the fact that both models follow the Multinomial Naïve Bayes model .



```
accuracy 0.9729002153625269
confusion matrix
 [[4825    0]
 [ 151  596]]
(row=expected, col=predicted)
```

Fig3. Řehůřek classifies overall



```
accuracy 0.9750435287289612
confusion matrix
 [[4516    0]
 [ 129  524]]
(row=expected, col=predicted)
```

Fig4. Proposed model classify overall

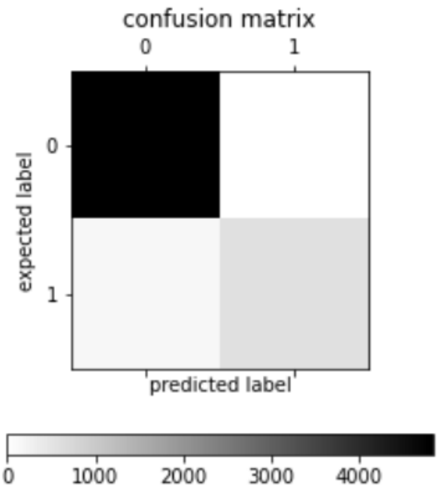


Fig5. Řehůřek Confusion Matrix

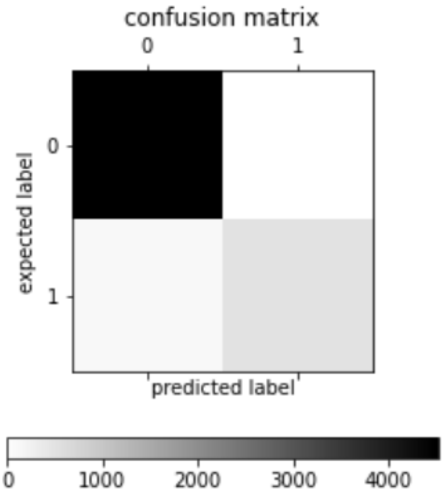
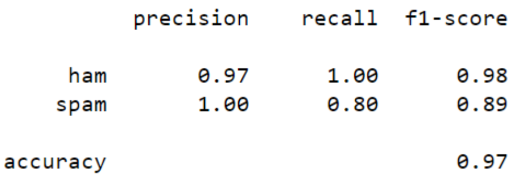


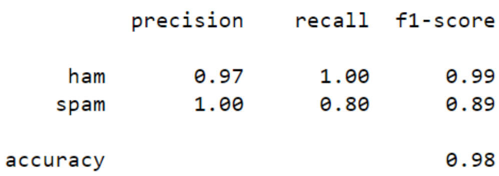
Fig6. Proposed Model Confusion Matrix

In both cases, the confusion matrix conclude an accuracy of 97% of performance, from where we can obtain precision, recall, and f1 score



```
             precision    recall  f1-score

         ham       0.97      1.00      0.98
        spam       1.00      0.80      0.89

    accuracy                           0.97
```

Fig7. Řehůřek Precision, recall, and f1-score score

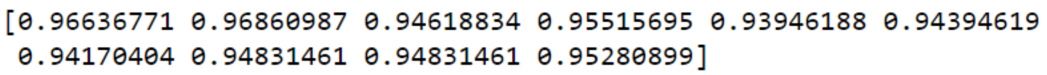


```
              precision     recall   f1-score

         ham       0.97      1.00       0.99
        spam       1.00      0.80       0.89

    accuracy                            0.98
```

Fig8. Proposed Model Precision, recall, and f1-

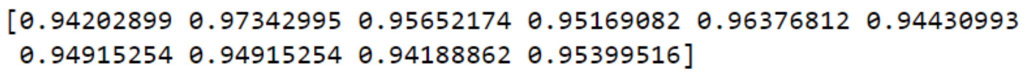
A 97% sounds an almost ideal scenario and is because the accuracy was evaluated on the same training data. To improve to a more real-scenario, the data is split into training and test data. A 20% test size is taken into consideration. Furthermore, the model has to be related just to training data. While the test data is used only for training purposes.
Following the Řehůřek testing environment and through repeated tests, I divided my training set into ten equally sizes parts just as Řehůřek did. A solution that provided the best performance-configuration for the algorithm.

Obtaining relative the same scores in a range between 0.93 and 0.96 proves the proposed model as stable.



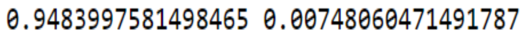
```
[0.96636771 0.96860987 0.94618834 0.95515695 0.93946188 0.94394619
 0.94170404 0.94831461 0.94831461 0.95280899]
```

Fig9. Řehůřek accuracy

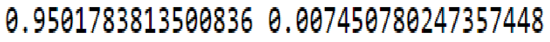


```
[0.94202899 0.97342995 0.95652174 0.95169082 0.96376812 0.94430993
 0.94915254 0.94915254 0.94188862 0.95399516]
```

Fig10. Proposed Model Accuracy



```
0.9483997581498465 0.00748060471491787
```



```
0.9501783813500836 0.007450780247357448
```

Fig9. Řehůřek scores

Fig10. Proposed scores

As we can observe above, in both approaches the accuracy is located under 98% reached by Lara. However, as we mention before probably the Lara method under a different training scenario would react differently.

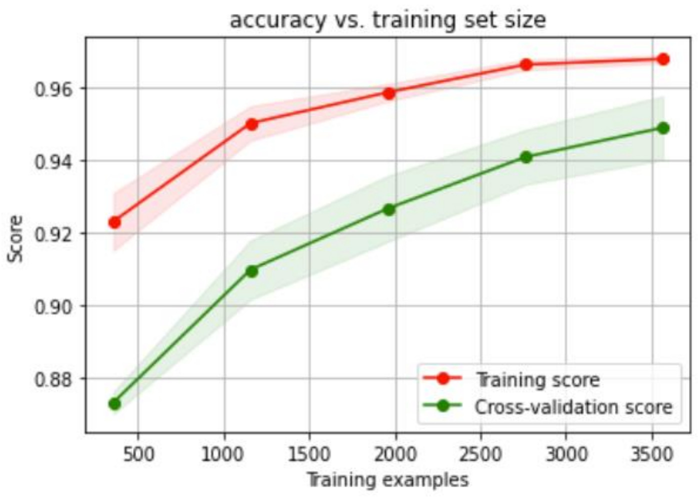


Fig13. Řehůřek accuracy vs training set size

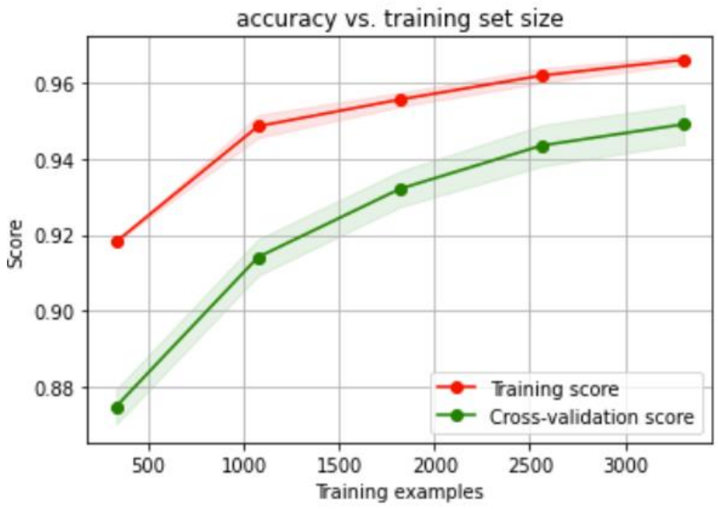


Fig14. Proposed Model accuracy vs training set size

From the above testing, it is possible to conclude the similar results obtained by the two tested models. However, The proposed model is slightly more stable in its accuracy tests. The reason is because before the vectorization the data is sanitized with a series of “rules” over the text, decreasing the noisy words letting the model focus on the context words and comparation. Finally Despite neither of both approaches model reaches 98% of accuracy, 94% to 95% is an acceptable result for an experimental model. If we observe the fig.13 and 14 graphs, a common issue shared by both approaches is the accuracy rate. Where the two different evaluation models are closely matched at the final phase, with a relative difference at the initial phase. This is because the model is not complex enough to capture all features given in the data. Two possible solutions would be: use more training data, due to the rapid popularity obtained by NB and their algorithms, people approach their utilities updating bigger sets of data on the web. Secondly, use a more complex model, to extract more features from the data.

7 Conclusion

Spam filtering Multinomial Naïve Bayes model based on text classification was set up. My goal was to observe and compare the incoming email text behavior to understand their patrons, and base on the text sanitization reach a higher accuracy percentage. After the research done during the elaboration of this paper, I am convinced that the text classification process plays a crucial area in spam filtering. A good text mechanism classification, along with the words’ sanitization can improve the classification spam effectiveness and help with future works to understand the spam behave.

This technique can be used over different datasets with high spam detection accuracy, the reason is the model. The model was designed to identify, split, sanitize, categorized, and count the text frequency. The proposed algorithm is an improvement for the Multinomial Naïve Bayes on its own. The reason is that when the difference between spam and ham are identified. In short, Naïve Bayes has won popularity due to the easy interpretation and the fact that is possible to tune algorithms to achieve a better performance compared to some other more complex black box models as Neural Networks.

8 References

[1] Kaspersky (2014). Spam and Phishing Statistics Report Q1-2014. [Online]. Available: https://usa.kaspersky.com/resource-center/threats/spam-statistics-report-q1-2014 [Accessed July. 30, 2020]

[2] Vergelis M, Shcherbakova T, Sidorina T. (2019). Spam and phishing in Q1 2019. [Online]. Available: https://securelist.lat/spam-and-phishing-in-q1-2019/88830/ [Accessed July. 30, 2020]

[3] https://www.techsoupcanada.ca/en/learning_center/10_sfm_explained

[4] Cunningham P, Nowlan N, Delany S, Haahr M. (2003). A Case-Based Approach to Spam Filtering that Can Track Concept Drift. [Online]. Available: https://www.researchgate.net/publication/2474902_A_Case-Based_Approach_to_Spam_Filtering_that_Can_Track_Concept_Drift [Accessed Sept. 2, 2020]

[5] Christina V, Karpagavalli S, Suganya G. (2010). Email Spam Filtering using Supervised Machine Learning Techniques. [Online]. Available: https://www.researchgate.net/publication/50235326_Email_Spam_Filtering_using_Supervised_Machine_Learning_Techniques [Accessed Sept. 2, 2020]

[6] Luo Q, Lui B, Yan J, He Z. (2011). Design and Implement a Rule-Based Spam Filtering System Using Neural Network. [Online]. Available: https://ieeexplore.ieee.org/document/6086218 [Accessed Sept. 2, 2020]

[7] Malarvizhi R, Saraswathi K. (2013). Content-Based Spam Filtering and Detection Algorithms- An Efficient Analysis & Comparison. [Online]. Available: https://www.ijettjournal.org/volume-4/issue-9/IJETT-V4I9P198.pdf

[8] Pelletier L, Almahana J, Choulakian V. (2004). Adaptive filtering of spam. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1344731 [Accessed Sept. 5, 2020]

[9] Himani B, Mahesh H. (2012). A review on support vector machine for data classification [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1039.2508&rep=rep1&type=pdf [Accessed Sept. 5, 2020]

[10] Ahmed A, Sara Al A, Ghada Al J, Ghayda H, Samy S A, (2018). Email Classification Using Artificial Neural Network [Online]. Available: https://philpapers.org/archive/ALGECU-3.pdf[Accessed Sept. 6, 2020]

[11] Christina V, Karpagavalli S, Suganya G. (2010) Email Spam Filtering using Supervised Machine Learning Techniques [Online]. Available: http://www.enggjournals.com/ijcse/doc/IJCSE10-02-09-151.pdf [Accessed Sept. 7, 2020]

[12] Puniškis D, Laurutis R, Dirmeikis R. (2006) An Artificial Neural Nets for Spam E-mail Recognition [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.4328&rep=rep1&type=pdf [Accessed Sept. 10, 2020]

[13] Sanghani G, Dr. Kotecha k. (2017). Support Vector Machine for personalized e-mail spam filtering [Online]. Available: http://www.iaeme.com/MasterAdmin/uploadfolder/IJARET_08_06_011-2/IJARET_08_06_011-2.pdf [Accessed Sept. 10, 2020]

[14] Mgpullen (2016). Spam filter using a support vector machine [Online]. Available: https://mgpullen.wordpress.com/2016/09/16/spam-filtering-using-a-support-vector-machine/ [Accessed Sept. 10, 2020]

[15] Gapta S. (2018). Pros and cons of various machine learning algorithms [Online]. Available: https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6 [Accessed Sept. 17, 2020]

[16] Navlani A. (2018). Decision tree classification in Python [Online]. Available: https://www.datacamp.com/community/tutorials/decision-tree-classification-python [Accessed Sept. 17, 2020]

[17] Subasi A, Alzahrani S, Aljuhani A, Aljedani M. (2018). Comparison of Decision Tree Algorithms for Spam E-mail Filtering [Online]. Available: https://ieeexplore.ieee.org/document/8442016 [Accessed Sept. 17, 2020]

[18] Chidanand A, Frederick D, Sholom W. (2001). Method for improved accuracy of decision tree-based text categorization [Online]. Available: https://patents.google.com/patent/US6253169B1/en [Accessed Sept. 29, 2020]

[19] Androutsopoulos I, Koutsias J, Chandrinos K, Paliouras G, Spyropoulos C. (2000). An evaluation of Naive Bayesian anti-spam filtering [Online]. Available: https://philpapers.org/archive/ALGECU-3.pdf [Accessed Aug. 1, 2020]

[20] Roy S, Patra A, Sau S, Mandal K, Kunar S. 9 (2013). An efficient spam filtering techniques for email account [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.682.7216&rep=rep1&type=pdf#page=69 p63-73 [Accessed Aug. 1, 2020]

[21] Joachims T. (N.D.) Text Categorization with Support Vector Machines: Learning with Many Relevant Features [Online]. Available: https://link.springer.com/content/pdf/10.1007%2FBFb0026683.pdf [Accessed Aug. 4, 2020]

[22] Kang L, Chen R, Chen Y, and Cao W. (2018). Using Naïve Bayes Method to Classify Text-Based Email [Online]. Available: https://ieeexplore.ieee.org/document/8701849 [Accessed Aug. 5, 2020]

[23] IT & the Lawyer (2003). E-mail usage guidelines [Online]. Available: https://www.ghostdigest.com/articles/e-mail-usage-guidelines/52145 [Accessed Aug. 9, 2020]

[24] Teravainen T. (N.D.) Email spam [Online]. Available: https://searchsecurity.techtarget.com/definition/spam [Accessed Aug. 9, 2020]

[25] Quigley R. (2010). Today in history: The first Spam Email Ever Sent [Online]. Available: https://www.themarysue.com/first-spam-email/ [Accessed Aug. 10, 2020]

[26] Puniškis D, Laurutis R, Dirmeikis R, An Artificial Neural Nets for Spam e-mail Recognition [Online] Available: http://www.eejournal.ktu.lt/index.php/elt/article/view/10681 [Accessed Aug. 10, 2020]

[27] Seongwook y, Dennis M. (2007). Spam Email Classification using an Adaptive Ontology [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.206&rep=rep1&type=pdf [Accessed Aug. 10, 2020]

[28] Wang R, Youssef A, Elhakeem A. (2006). On Improving the Performance of Spam Filters Using Heuristic Feature Selection Techniques [Online]. Available: https://ieeexplore.ieee.org/document/1644610/citations?tabFilter=papers [Accessed Aug. 10, 2020]

[29] Sather T. (2015). Everything Marketers Need To Know About Engagement-Based Spam Filtering [Online]. Available: https://marketingland.com/everything-marketers-need-know-engagement-based-spam-filtering-123437 [Accessed Aug. 13, 2020]

[30] Machine Learning Repository (UCI). (2012). SMS Spam Collection Data Set [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection [Accessed Aug. 13, 2020]

[31] Řehůřek R. (2014). Practical Data Science in Python [Online]. Available: https://radimrehurek.com/data_science_python/ [Accessed Aug. 14, 2020]

[32] Lara C. (2019) Naïve Bayes Spam Classification. [Online]. Available: https://github.com/LeanManager/NLP_Technical_Founders/blob/master/NaiveBayes/NLP_Naive_Bayes.ipynb [Accessed Aug. 15, 2020]

9 Appendix

A.1

TO THE EDITOR OF THE TIMES

Sir, -- On my arrival home late yesterday evening a "telegram," by "London District Telegraph," addressed in full to me, was put in my hands. It was as follows: --

"Messrs. Gabriel, dentists, 27, Harley-street, Cavendish-square. Until October Messrs. Gabriel's professional attendance at 27, Harley-street, will be 10 till 5."

I have never had any dealings with Messrs. Gabriel, and beg to ask by what right do they disturb me by a telegram which is evidently simply the medium of advertisement? A word from you would, I feel sure, put a stop to this intolerable nuisance. I enclose the telegram, and am,

Your faithful servant,

Upper Grosvenor-street, May 30. M. P.