

**Assessment 1: Data Mining Practice and Analysis**

**Emails classification with Naïve Bayer**

Author: Luu Thanh Tu (#13716589) & Bertocco Thomas-Killian (#13758322)

Lecturer: Kwang Lim

CP5634 Data Mining| SP52 | 2019

27 September 2019

Table of Contents

[Abstract 3](#_Toc20323099)

[Introduction 4](#_Toc20323100)

[Algorithms 5](#_Toc20323101)

[The Multilayer perceptron (MLP) 5](#_Toc20323102)

[The C4.5 decision tree classifier 6](#_Toc20323103)

[The Support Vector Machine (SVM) 8](#_Toc20323104)

[The k-Nearest Neighbors (KNN) 10](#_Toc20323105)

[Literature review. 12](#_Toc20323106)

[Dataset 14](#_Toc20323107)

[Methodology 15](#_Toc20323108)

[The algorithm 15](#_Toc20323109)

[Data preprocessing 15](#_Toc20323110)

[Remove upper-case characters 15](#_Toc20323111)

[Remove numbers 16](#_Toc20323112)

[Remove symbols 16](#_Toc20323113)

[Remove stop words 16](#_Toc20323114)

[Tokenize word 17](#_Toc20323115)

[Remove streaming 17](#_Toc20323116)

[Remove Lemmatization 17](#_Toc20323117)

[Part of speech tagging (POS) 17](#_Toc20323118)

[Chunking (shallow parsing): 18](#_Toc20323119)

[Generating features 20](#_Toc20323120)

[Use the Naïve Bayes classifier 21](#_Toc20323121)

[Use the J48 classifier 25](#_Toc20323122)

[Use the k-NN 26](#_Toc20323123)

[Support vector machine 27](#_Toc20323124)

[Evaluate the result 27](#_Toc20323125)

[Graphs 32](#_Toc20323126)

[Future work 36](#_Toc20323127)

[Conclusion 37](#_Toc20323128)

[Reference: 38](#_Toc20323129)

# Abstract

In the fourth industrial revolution, the internet has evolved significantly thanks to its vast number of users. According to Internet World Stats, there are 4,536,248,808 internet users in 2019, and this number is anticipated to be double or triple in the future. The internet also has many negative aspect that are difficult to fix. One of them is email spamming. For many people, email is an important communication channel that allows them to talk, work and discuss with their friends, families and collogues. However, to read these important emails, they must find them between tons of spam. Hence, email spam makes people feel more annoyed and disturbed when they open their mailboxes and realize that they have thousands of unwanted emails. To avoid this problem, there are different techniques are used to detect and sort emails, such as Random Forest, Naïve Bayesian, Support Vector Machine (SVM) and Neutral Network. This paper will use different methods of the Naive Bayes algorithm to evaluate results. Finally, we will compare the results to WEKA.

# Introduction

Sending mail has begun from a hundred years ago. At that time, the act of sending and receiving email was very time-consuming and costly. In contrast, since the occurrence of e-mail, a sender can send millions of advertisements at no cost, and in a second. Therefore, many unwanted massive emails are spreading widely. Many of us spend at least 5 or 10 minutes every day to sort and remove spams. Moreover, spams can create other problems for businesses. For example, a company that receive many spams may see its bandwidth decrease. As a result, many researches have been conducted find a solution to spams.

Nevertheless, these researches have encountered several problems with filtering spam emails. Thanks to machine learning, it is now easier to classify and cluster, but some fields and characteristics are still difficult to process. Spams dataset relates to text mining, and it belongs to the Natural Language Processing (NLP) which is the way that computers describe human languages. Analyzing human language never been considered an easy work because every language has their own rules, grammar, and slangs. Therefore, emails filtering and prediction consist of determined the deviation and error rate that reduces the percentage of accuracy which decides spam and ham emails. This is why the algorithm will train on a raining dataset first.

The objectives of this research are:

1. Preprocess a dataset.
2. Deploy Naïve Bayes algorithms to filter spam of two datasets.
3. Evaluate the results.

# Algorithms

There are numerous algorithms that are useful for either filtering spam email or spam email. We will use some of them.

## The Multilayer perceptron (MLP)

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. This algorithm must have at least three layers: an input layer, a hidden layer, and an output layer. Multilayer perceptron algorithm is structured as a neutral network. Both the hidden layer and output layer are neutron that has a nonlinear activation function while input code is not. MLP is a supervised learning technique called backpropagation algorithm. The number of layers in an MLP is calculated by the number of hidden layers plus 1. That is, when counting the number of layers of an MLP, we do not count the input layers. The number of layers in an MLP is usually denoted as L.

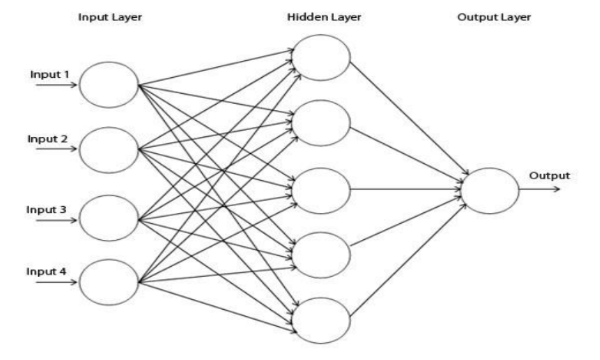


Fig 1: 3 layers of Multilayer perceptron.

As mentioned above, a linear activation function exists in all neurons of multilayer perceptron, which decides the weight of input to produce output. There are two common activation functions which are:

y(vi) = tanh(vi) and y(vi) = (1 + e vi) -1

Multilayer perceptron can be applied to classify spam and ham email. Before conducting MLP spam email filtering, there are a few steps that need to be done:

1. Initialization: pick the synaptic weights and thresholds from a uniform distribution
2. Presentations by training examples: present the network by epochs of training examples.
3. Forward computation: Signals are computed by proceeding forward though the network by layer by layer basis.
4. Backward computation: calculate the local gradients of the netwrk
5. Iteration: iterate the forward and backward computations until the chosen stopping criterion is met.

Because MPL is a neuron classifier that means it is very complex and very hard to implement.

## The C4.5 decision tree classifier

C4.5 Decision tree learning is a method for approximating discrete-valued functions, in which the learned function is represented by a decision tree. Learned trees can also be-represented as sets of if-then rules to improve human readability. These learning methods are among the most popular of inductive inference algorithms and have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants. C4.5 Decision tree learning is a heuristic, one-step look ahead (hill-climbing), non-backtracking search through the space of all possible decision trees (Polat, K., & Güneş, S. (2009)).

There are two steps to implement C4.5 decision tree classifier:

* Create a C4.5 decision tree model.
* Train the model.

In order to train the model, the tree is turned to a smaller subset based on attribute value prefixed. The formula to divide the C4.5 decision tree into a smaller subset is defined as below:

(x, Y) = (x1, x2, x3, …., xk, Y)

When we are splitting the C4.5 decision tree model, there are three criteria to create leaf node:

1. If all cases are of the same class, the tree is a leaf, and so the leaf is returned labeled with this class.
2. For each attribute, calculate the potential information provided by a test on the attribute and the gain in the information that would result from a test on the attribute
3. Depending on the current selection criterion, find the best attribute to branch on.

Spam email filtering with C4.5 decision tree algorithm is the act of splitting single words from text and classify them base of ham and spam nodes. Each ham and spam nodes consist of ham and spam words to the corresponding sub-nodes.

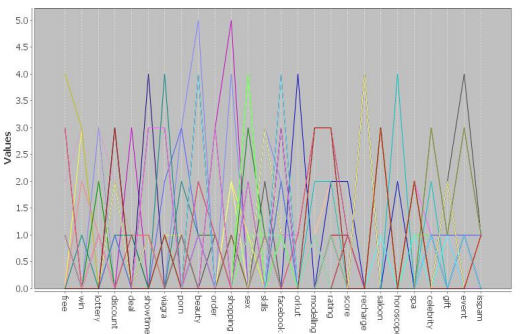


Figure 3: C4.5 decision tree model.

C4.5 decision tree model works well with the small number of email because it is very easy to understand. However, this model will not work well if there are thousands of emails to classify because it will create billion nodes that means it is very hard to manage.

## The Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised machine learning algorithm which provides accurate discrimination in high feature space (Amayri, O., & Bouguila, N. (2010)). SVM is a discriminative classifier which makes a decision based on hypothesis with linear equality and inequality constraints. During training phase, SVM improves by narrowing down boundaries for classification.

There are three passes to implement SVM algorithm:

1. The first pass is performed using conventional batch training of SVM, with n labeled examples, that generates the discriminant function F(x). (Sanghani, Gopi & Kotecha, Ketan. (2018))
2. The second pass comprises a series of testing phases in which small batches of incoming unlabeled e-mails are given to identify true labels.
3. The third pass is carried out by activating incremental training whenever any one of the two performance criteria – accuracy and false positive (FP) rate is violated.

SVM formulas:

The optimization problem for a soft-margin SVM is:

Where w is the normal vector of the separating hyperplane in feature space and C > 0 is a regularization parameter controlling, (x, y) is the data in the dataset.

The lagrangian form of the dual problem is: 

he normal vector of

the separating plane w is calculated as:

he normal vector of

the separating plane w is calculated as:

he normal vector of

the separating plane w is calculated as:

he normal vector of

the separating plane w is calculated as:



The normal vector of the separating plane w is calculated as:

The kernel function of x with every support vector: 

SVM divides the dataset into two vectors: spam and ham. The learning process of SVM has deviation for classifying personalized spam and spam email. SVM improves itself by using a heuristic function which re-calculates feature’s information.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Datasets | Inbox | Incremental Training with updated Feature Set | | | SVM Batch Training | | |
| Accuracy | Precision | Recall | Accuracy | Precision | Recall |
| ENRON | ENRON1 | 96.28 | 0.97 | 0.95 | 94.18 | 0.95 | 0.93 |
| ENRON2 | 96.01 | 0.98 | 0.90 | 87.90 | 0.93 | 0.62 |
| ENRON3 | 96.76 | 0.98 | 0.93 | 93.83 | 0.96 | 0.89 |
| ENRON4 | 98.19 | 0.98 | 0.98 | 96.49 | 0.98 | 0.96 |
| ENRON5 | 97.45 | 0.94 | 0.99 | 96.14 | 0.90 | 0.99 |
| ENRON6 | 96.86 | 0.94 | 0.98 | 91.58 | 0.92 | 0.92 |
| ECUE | ECUE 1 | 97.32 | 0.99 | 0.96 | 91.63 | 0.98 | 0.92 |
| ECUE 2 | 96.52 | 0.99 | 0.92 | 86.18 | 0.99 | 0.85 |

Table 4 The classification results for Personalized e-mail spam filtering

Table 1. The classification results for Personalized e-mail spam filtering

Table 4 The classification results for Personalized e-mail spam filtering

## The k-Nearest Neighbors (KNN)

KNN is the non-parametric, instance-based learning, or lazy learning where the function is accurate in a local instance, and all the computation will suspend until classification phase. The result of KNN is a class membership while the object which classifies member is calculated from the distance of its neighbors. Due to the nature of the feature vectors, with features as positive integer values, the Euclidean distance metric will be used for both methods.

KNN formula: (x2−x1) ²+(y2−y1) ²+(z2−z1) ² …… (n2-n1) ²

Emails which are defined as spam must satisfy this by this rule:

Where K is the number of nearest neighbors, D is the total distance between the query points and all neighbors, dk is the distance between the query point and the k the neighbors. t is a fixed threshold value (0 < t ≤ 1). The choice of ''k'' bases on the data. If the larger values of ''k'' reduce the effect of noise on the classification, then it makes boundaries between classes less distinct.

There are four steps to deploy the k-NN model:

1. Compute Euclidean from defined point to sample points.
2. Order samples taking for account calculated distances.
3. Choose optimal ''k'' nearest neighbor. If k = 1 => object is the nearest neighbor.
4. Calculate an inverse distance weighted average with the ''k''-n

To build a training model, we need to store the feature vectors and class labels of the training samples.

The accuracy of the k-NN classifier depends on choosing approximately parameter. These parameters are k (number of nearest neighbors), p (Maximum training points), t (Classification threshold), e (Approximation error bound). Since best parameters are chosen then noise data will be limited.

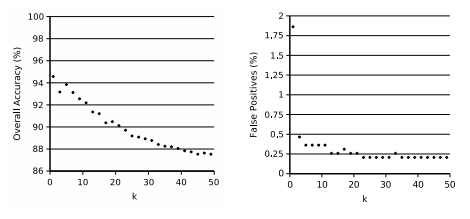


Fig. 5. Effects of the varying error bound

k-NN classifier can release very poor accuracy when it has incorrect k.

# Literature review.

Bayesian inference is a simple type of inference that the computer relies on the probabilities of the pre-existing data of the problem to find the probability of the output for subsequent inputs (Puga, J. L., Krzywinski, M., & Altman, N. (2015)). Unlike other theory, the result of Bayesian depends on assumption and knowledge. For example, predicting the result of the football match bases on the Bayes theory requires the opinion about both teams as well as the statistic which indicates the power of these teams. The improvement and accuracy of Bayes theory bases on the amount of observed data that the Bayes theory can learn.

P(A, B)

Probability of happening A on its own, regardless of B. denoted by P (A) and read as the probability of A. This is called marginal probability or a priori probability, it is "a priori" in the sense that it is not interested in any information about B.

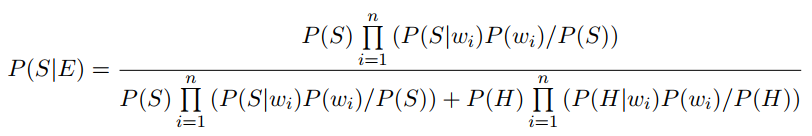
Probability of B itself, regardless of A. Denoted by P (B) and read as "probability of B". This quantity is also called normalizing constant, because it is always the same, regardless of the fact that A is trying to know.

The probability that B happens when A knows. Denote as P (B | A) and read as "probability of B if there is A". This quantity is called likelihood when B knows that A has occurred. Note that there is no confusion between the possibility of B knowing A and the probability of A knowing B.

Paul Graham’s Method is a different methodology to deploy the Naïve Bayes algorithm. This method is conducted in the dataset, which has 2 types: ham and spam. All the attributes are considerate as token (alphanumeric characters, dashes, apostrophes, and dollar signs and others are token separators) except digits tokens, and Html comments. Then, it does a list of words frequency in both ham and spam, by the Bayes theorem and the probability that each token is in a spam, or a ham. This method can reduce the false positives rate by doubling the number of each token in hams. After getting the result, it determines if the token is spam or ham (if it is closer to the 0 value or not).

However, this methodology has a problem because it is conducted in a dataset that has similar spams and hams. This method will have difficulties to see the slight difference between them.

There is a method that can solve this problem:



Where P(S| is known in Paul Graham’s Method. This method can improve the accuracy.

# Dataset

Spam email dataset is a combination of data which we collect from 2 sources:

The first one is: <http://www2.aueb.gr/users/ion/data/enron-spam/>

The second is: <http://www.enron-mail.com/email/>

There are also some email data that we took from our real mailboxes.

Those datasets include 2 types of emails:

We have two types of datasets:

* Test-mails dataset: This dataset includes 2 types: hams and spams. Spams are unwanted emails, and hams are normal or important emails. There are 100 emails for each category.
* Training dataset: Similar to the test dataset but it has fewer emails than test mails. Only 50 emails for each category.

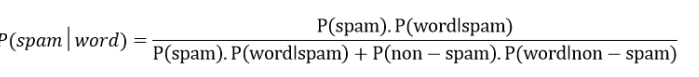
# Methodology

## The algorithm

For this research, we will focus on the Naïve Bayes algorithm.

The Naïve Bayes theory is a simple version of the Bayes theory that calculates a list of probabilities by counting the frequencies and combinations of words in a given dataset. The Naïve Bayes classifier use a bag of its word for the frequency of the word to train the classifier.

The formula of Naïve Bayes is described as below:



In which:

* P(spam|word) is the probability which indicates the word occur in spam
* P(spam) is the probability of spam email
* P(word|spam) is the probability that the specific word appears in spam message.
* P(non − spam) is the probability that occurs word is ham.
* P(word|non − spam) is the probability that the specific word appears in ham email.

There are three steps to implement Naïve Bayes classifier:

* Data preprocessing.
* Feature selection
* Classifier

## Data preprocessing

In order to apply the Naïve Bayes algorithm, all data must be converted into strings and must not have any special characters. Otherwise, the result or accuracy of the algorithm will be affected.

For preprocessing step, eight procedures will be conducted:

### Remove upper-case characters

Dataset has both lower- and upper-case words. However, when conducting a bag of its words then it will classify upper-case word have a different meaning from the lower-case word. For example, Free and free have similar meaning, but the machine will consider them as different words. To avoid this case, we use the str.lower() function provided by python.

Example: The email address of the Data Mining (CP5634) teacher is: teacher@jcu.sg -> the email address of the data mining (cp5634) teacher is: [teacher@jcu.sg](mailto:teacher@jcu.sg).

Remove numbers

To enhance the probabilities of the Naïve Bayes classifier, non-alphabet characters have to be removed.

In this case, all the numbers are removed.

Example: the email address of the data mining (cp5634) teacher is: teacher@jcu.sg -> the email address of the data mining (cp) teacher is: [teacher@jcu.sg](mailto:teacher@jcu.sg).

### Remove symbols

To enhance the probabilities of the Naïve Bayes classifier. The non-alphabet characters have to be removed. In this case, all the character ([!”#$%&’()\*+,-./:;<=>?@[\]^\_`{|}~]:) are removed.

Example: the email address of the data mining (cp) teacher is: [teacher@jcu.sg](mailto:teacher@jcu.sg). -> the email address of the data mining cp teacher is teacher jcu sg

### Remove stop words

Some words are very common in English, and they cannot be considered as a setting to separate hams and spams, because they are very common in both of them. In this step, we will remove these common words. This step can be conducted in Weka. Here is a list of some of them:

Example: the email address of the data mining (cp) teacher is: [teacher@jcu.sg](mailto:teacher@jcu.sg). -> email address data mining cp teacher teacher jcu sg

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | him | themselves | be | and | against | Out | How | Only | now |
| me | his | What | been | but | between | on | all | Own |  |
| my | himself | which | being | if | into | off | any | Same |  |
| myself | she | who | have | or | through | over | both | So |  |
| we | her | whom | has | because | during | under | each | than |  |
| our | hers | this | had | as | before | again | few | Too |  |
| ours | herself | that | having | until | after | further | more | Very |  |
| ourselves | it | these | Do | while | above | then | most | S |  |
| You | its | those | does | of | below | once | other | t |  |
| Your | itself | am | did | at | to | here | some | can |  |
| yours | they | is | doing | by | from | there | such | will |  |
| yourself | then | are | a | for | up | when | no | just |  |
| yourselves | their | was | an | with | down | where | nor | don |  |
| He | theirs | were | the | about | In | why | not | should |  |

EXAMPLES OF NON FILTERED AND FILTERED EMAIL (WRITE, NO PICS)

### Tokenize word

As mentioned in the beginning: Naïve Bayes classifier will calculate the probabilities that a word appears in the dataset. There are different ways to tokenize the word. Either turn string to separate word or turn a word into a number.

EXAMPLES OF NON FILTERED AND FILTERED EMAIL (WRITE, NO PICS)

### Remove streaming

Like the lower- and upper-case, Ing and non-ing have a similar meaning, but they can be considered as different words. To avoid this case, we remove all the ing words.

EXAMPLES OF NON FILTERED AND FILTERED EMAIL (WRITE, NO PICS)

### Remove Lemmatization

Like the lower- and upper-case. S,es and non-s,es have a similar meaning, but they can be considered as different words. To avoid this case, we remove all the s,es words.

EXAMPLES OF NON FILTERED AND FILTERED EMAIL (WRITE, NO PICS)

### Part of speech tagging (POS)

The classifying process can be easier when it knows the structure of sentences. For example, Noun, Verb, Adjective. To do that we use the POS classifier.

Here is the list of POS.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC coordinating conjunction | JJS adjective, superlative ‘biggest’ | PDT predeterminer ‘all the kids’ | TO to go ‘to‘ the store. | WDT wh-determiner which |
| CD cardinal digit | LS list marker 1) | POS possessive ending parent‘s | UH interjection | WP wh-pronoun who, what |
| DT determiner | MD modal could, will | PRP personal pronoun I, he, she | VB verb, base form take | WP$ possessive wh-pronoun whose |
| EX existential there (like: “there is” … think of it like “there exists”) | NN noun, singular ‘desk’ | PRP$ possessive pronoun my, his, hers | VBD verb, past tense took | WRB wh-adverb where, when |
| FW foreign word | NNS noun plural ‘desks’ | RB adverb very, silently, | VBG verb, gerund/present participle taking |  |
| IN preposition/subordinating conjunction | NNP proper noun, singular ‘Harrison’ | RBR adverb, comparative better | VBN verb, past participle taken |  |
| JJ adjective ‘big’ | NNPS proper noun, plural ‘Americans’ | RBS adverb, superlative best | VBP verb, sing. present, non-3d take |  |
| JJR adjective, comparative ‘bigger’ | PDT predeterminer ‘all the kids’ | RP particle give up | VBZ verb, 3rd person sing. present takes |  |

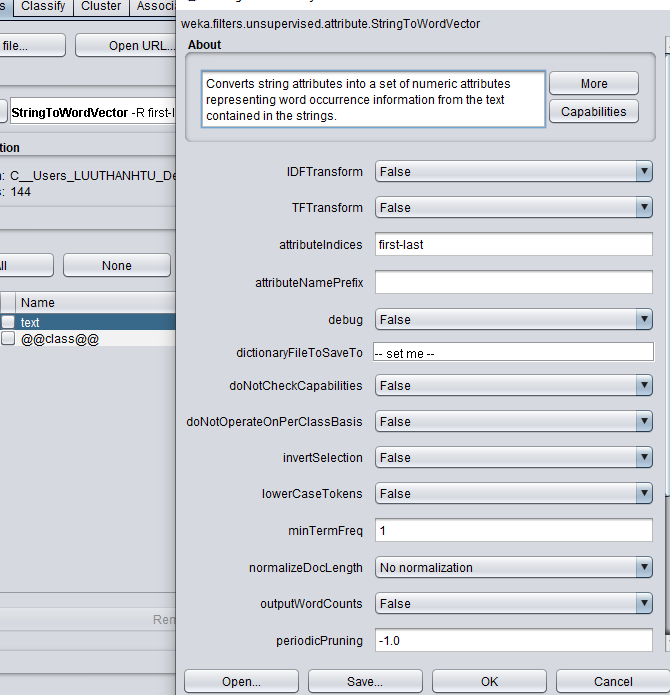
EXAMPLES OF NON FILTERED AND FILTERED EMAIL (WRITE, NO PICS)

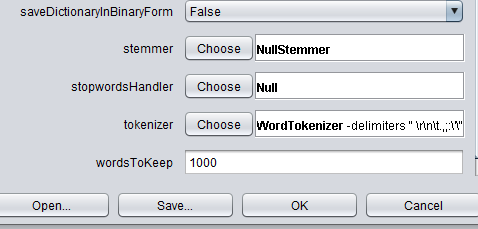
### Chunking (shallow parsing):

We can do further step to enhance the classifier which can link the POS structure to a correct grammar structure.

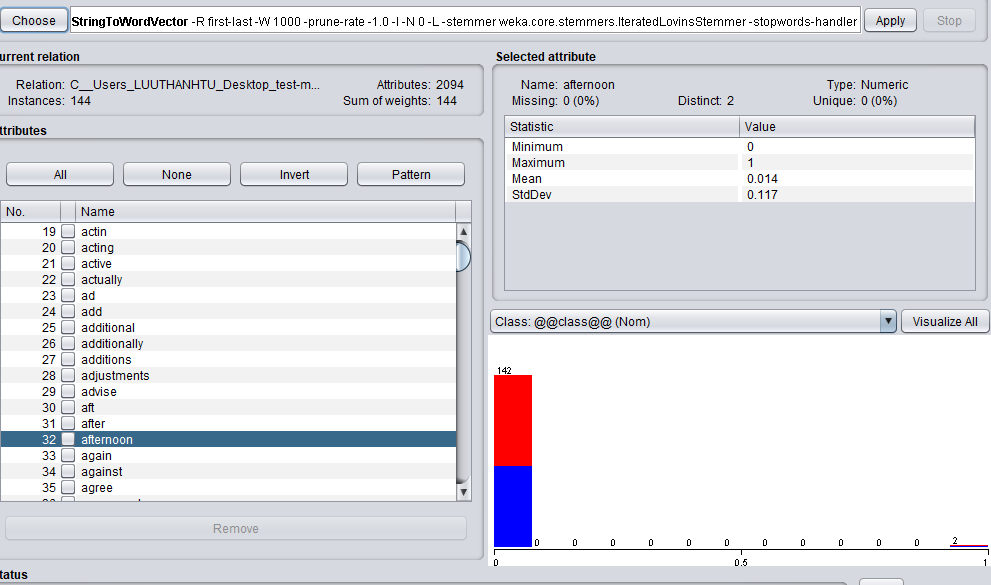
EXAMPLES OF NON FILTERED AND FILTERED EMAIL (WRITE, NO PICS)

**NOTE**: Remove stop word, Tokenization, and Steaming steps can be done in Weka (prepossessing => Filter => unsupervised => attribute => StringtoWordvector)





Moreover, we change the data correspond to the class (ham and spam). For example, the word ‘afternoon’ is classified by class by this:



## Generating features

After the unnecessary data is removed, we have to choose which feature will display the data. Weka provides three features: BestFirst, Greedystepwise, and Ranking.

**BestFirst**: Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Sets the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward or start with the full set of attributes and search backward or start at any point and search in both directions (BestFirst, (n.d.)).

**Greedystepwise**: Performs a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected Class (GreedyStepwise, (n.d.)).

**Ranking**: This feature will display words in ranking level.

In this assignment, we choose to use the BestFirst feature.

## Use the Naïve Bayes classifier

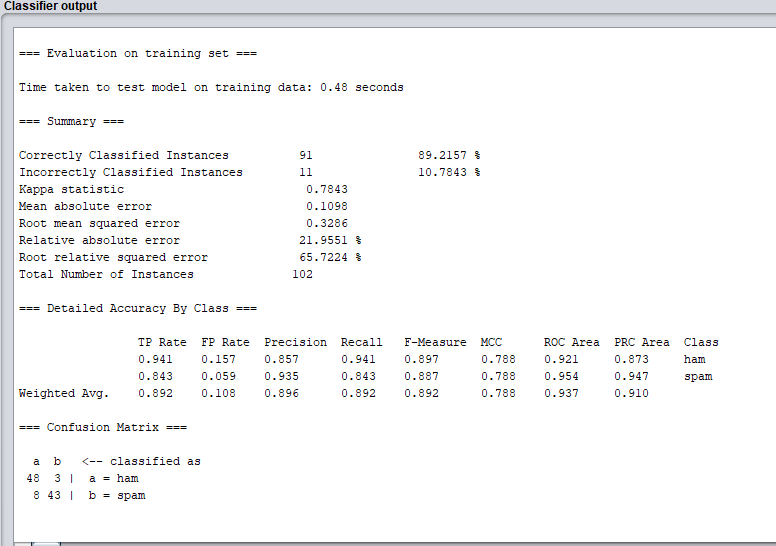
Naïve Bayes has three types of classifiers: normal Naïve Bayes, Naïve Bayes multinomial, Naïve Bayes multinomial text. Each types output different results.

The Normal Naïve Bayes will go through every single word. It will calculate the accuracy of these words.

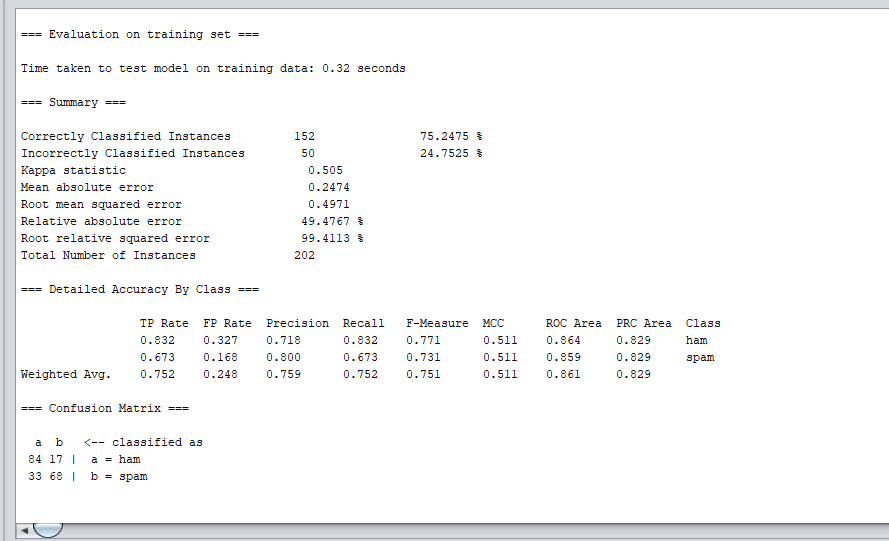
The Naïve Bayes multinomial will count the occurrence and relative frequency of a word in the dataset.

The Multinomial Naive Bayes for text data operates directly (and only) on String attributes. Other types of input attributes are accepted but ignored during training and classifications (Class NaiveBayesMultinomialText, (n.d.)).

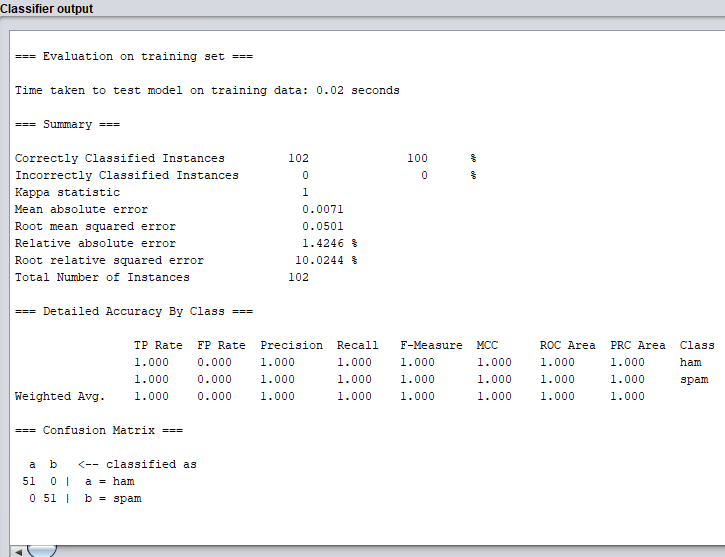
Naïve Bayes for training dataset:



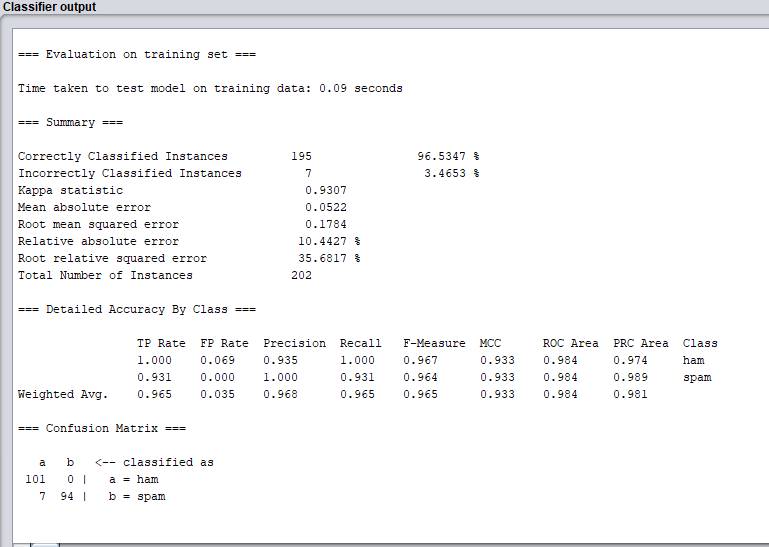
Naïve Bayes for testing dataset:



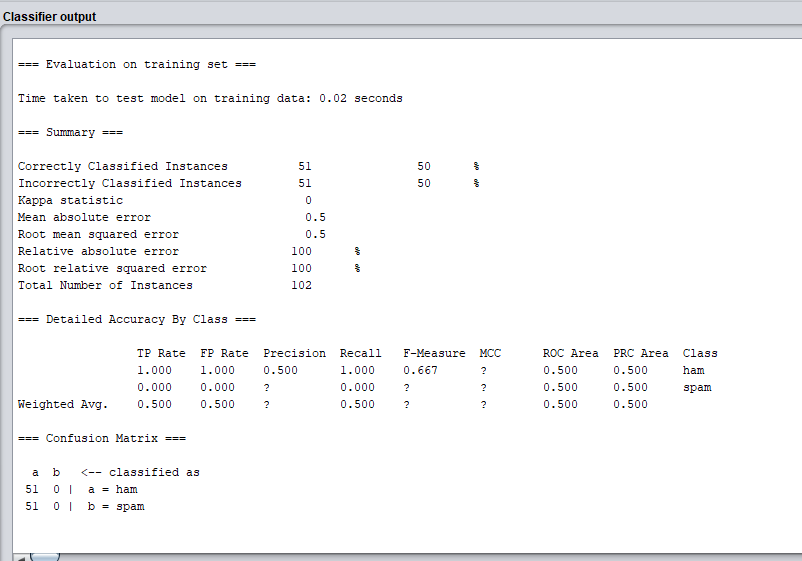
Naïve Bayes multinomial with a training dataset:



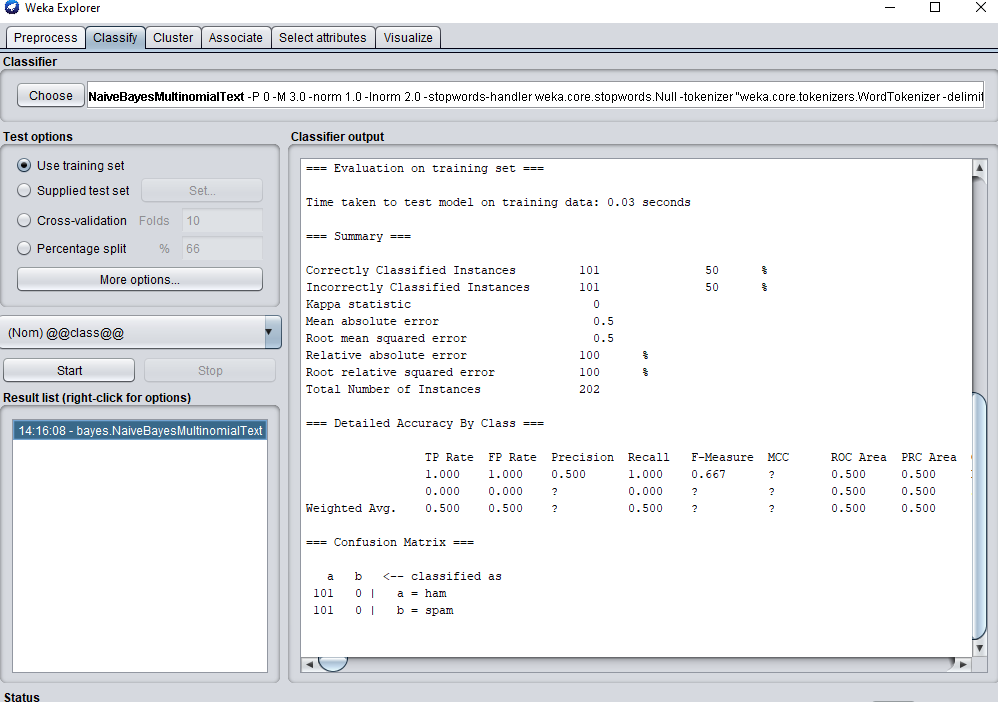
Multinomial Naïve Bayes for test-mails:



Multinomial Naïve Bayes text for training dataset:



Multinomial Naïve Bayes text for test mails dataset:



There are five parameters that we need to consider:

Accuracy: The percentage to define spam and ham in a dataset.

Accuracy

Recall (sensitive): is the percentage of relevant instances that have been received over the total amount of relevant instances.

Recall

Precision (also called positive predictive value) is the percentage of relevant for spam email

Precision

F-measure: the weighted average of precision and recall rate.

F-measure

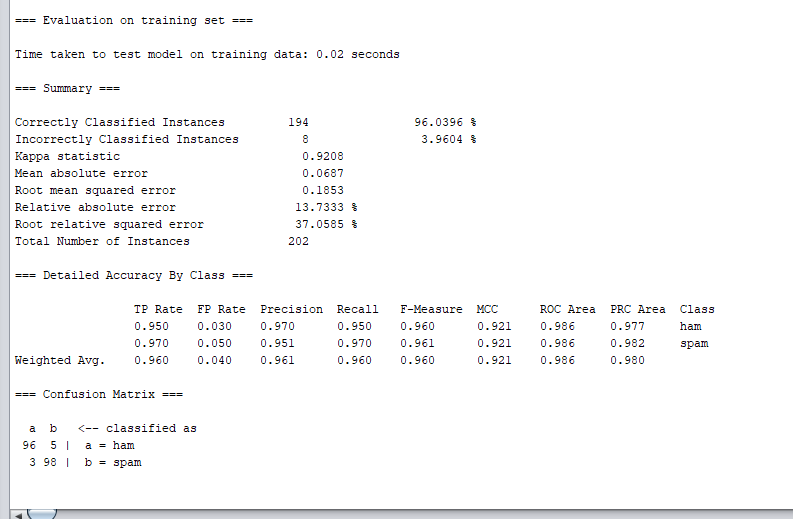
Where TP is the actual spam email in the dataset.

TN is the actual of ham email in the dataset.

FN is the mismatched rate of ham email.

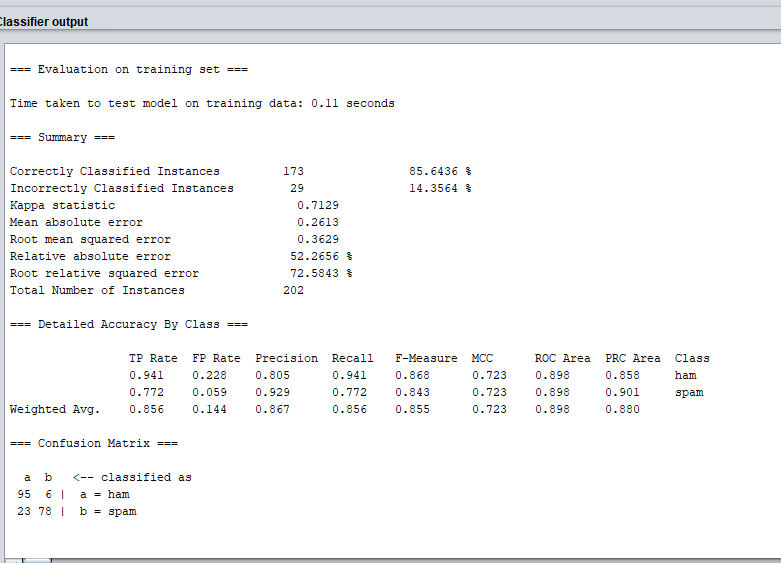
FP is the mismatched rate of spam email.

## Use the J48 classifier

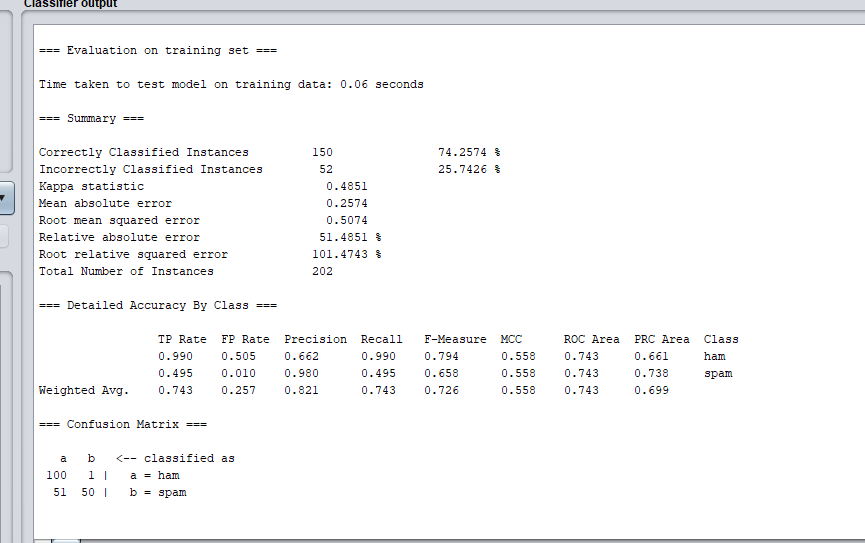


## Use the k-NN

k-NN with k = 2



## Support vector machine



## Evaluate the result

**The result of Naïve Bayes for training dataset**

The number of correct spam mails is 91/102 (89.2177%)

The number of incorrect emails is 11/102 (10.7843%)

Accuracy = 0.667

Recall = 0.903

Precision

F-measure = 0.674

**The result Naïve Bayes for a testing dataset**

There are 152/202 of correct spam emails (75.2475 %)

There are 50/202 of incorrect spam emails (24.7525%).

Accuracy = 0.667

Recall = 0.802

Precision

F-measure = 0.666

**The result of multinomial Naïve Bayes for training dataset**

The number of correct spam emails is 102/102 (100%)

The number of incorrect spam emails is mail is 0/102 (0%).

Accuracy = 0.667

Recall = 1

Precision

F-measure = 0.666

**The result of multinomial Naïve Bayes for testing dataset**

The number of correct emails is 195/202 (96.5347%)

The number of incorrect emails is 7/202 (3.4653%)

Accuracy = 0.667

Recall = 0.96

Precision

F-measure = 0.727

**The result of multinomial Naïve Bayes text for the training dataset**

The number of correct emails is 51/102 (50 %)

The number of incorrect emails is 51/102 (50 %)

Accuracy = 0.667

Recall = 0.667

Precision

F-measure = 0.667

**The result of multinomial Naïve Bayes text for test-mails dataset**

The number of correct emails is 194/202 (50 %)

The number of incorrect emails is 101/202 (50 %)

Accuracy = 0.667

Recall = 0.667

Precision

F-measure = 0.667

**The result of J48 for the test-mails dataset**

The number of correct emails is 194 /202 (96.0396%)

The number of incorrect emails is 8 /202 (3.9604%)

Accuracy = 0.667

Recall = 0.961

Precision

F-measure = 0.666

**The result of J48 for test-mails dataset**

The number of correct emails is 194 /202 (96.0396%)

The number of incorrect emails is 8 /202 (3.9604%)

Accuracy = 0.667

Recall = 0.961

Precision

F-measure = 0.666

**The result of k-NN for test-mails dataset**

The number of correct emails is 173 (85.6436%)

The number of incorrect emails is 29/202 (14.3564 %)

Accuracy = 0.667

Recall = 0.8744

Precision

F-measure = 0.6661

**The result of SVM for test-mails dataset**

The number of correct emails is 150 (74.2574 %)

The number of incorrect emails is 52/202 (25.7426%)

Accuracy = 0.667

Recall = 0.538

Precision

F-measure = 0.555

### Graphs

**Between 3 types of Naïve Bayes**

Recall rate of test-mails:

Precision in test-mails dataset:

Recall rate of training-mails:

Precision in test-mails dataset:

Throughout three Naïve Bayes classifiers, if we compare four evaluation measurements with the similarity in accuracy and F-measure, it is obvious that the multinomial Naïve Bayes is the best for classifying spams. Since we already filtered the dataset, the multinomial Naïve Bayes text is less accurate than other Naïve types.

As seen before, the accuracy of these classifiers is almost equal, however, we can observer that the Multinomial Naïve Bayes has better prediction results. Although j48 decision tree produces positive result, it will not work accurately if the dataset expands.

SVM classifier produces more balanced results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Multinomial Naïve Bayes | J48 | K-NN | SVM |
| Recall | 0.96 | 0.961 | 0.8744 | 0.538 |
| Precision | 0.508 | 0.51 | 0.538 | 0.573 |
| F-measure | 0.727 | 0.666 | 0.6661 | 0.555 |

## Future work

As mentioned, he spam email dataset had only 200 emails in total. In the future, we would like to increase the population to thousands or millions for spams and hams. Moreover, the accuracy of the dataset is around 86% to 96% for large scale dataset. To improve the result, we need to build a better training model, and find the ideal settings to increase accuracy.

# Conclusion

E-mail spam filtering is an important issue in the world of network security and machine learning techniques. Naive Bayes classifier shows better potential in this process of filtering e-mail spam. The research also shows the great relationship between the scale of data and the performance of Naïve Bayes classifier. Moreover, among four types of classifier, Naïve Bayes classifier shows great result to filter spam email.

# Reference:

Ali S. A., Souratib N.,K. Proposed efficient algorithm to filter spam using machine learning techniques. Pacific Science Review A: Natural Science and Engineering ,18(2), 145-149 https://doi.org/10.1016/j.psra.2016.09.017

Class BestFirst, (n.d.). Retrieved from: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/BestFirst.html

Class GreedyStepwise, (n.d.). Retrieved from: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/GreedyStepwise.html

Class NaiveBayesMultinomialText, (n.d.). Retrieved from: http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayesMultinomialText.html

INTERNET USAGE STATISTICS The Internet Big Picture (n.d.). Retrieved from: https://www.internetworldstats.com/stats.htm

Almeida, T. A., Almeida, J., & Yamakami, A. (2011). Spam filtering: How the dimensionality reduction affects the accuracy of naive bayes classifiers. Journal of Internet Services and Applications, 1(3), 183-200. doi:10.1007/s13174-010-0014-7

https://towardsdatascience.com/naive-bayes-intuition-and-implementation-ac328f9c9718

https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908

Amin, Muhammad & Ali, Amir. (2018). Application of Multilayer Perceptron (MLP) for Data Mining in Healthcare Operations. Retrieved from: https://www.researchgate.net/publication/329277388\_Application\_of\_Multilayer\_Perceptron\_MLP\_for\_Data\_Mining\_in\_Healthcare\_Operations

A.Sesha Rao A.S., Avadhani P.,S., Chaudhuri B., N., (2016). A Content-Based Spam E-Mail Filtering Approach Using Multilayer Percepton Neural Networks. International Journal of Engineering Trends and Technology (IJETT) – Volume-41 Number-1 - https://pdfs.semanticscholar.org/fdf4/301dcb1384b6b05e0343159561f614ab4526.pdf

Polat, K., & Güneş, S. (2009). A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. Expert Systems with Applications, 36(2), 1587-1592. doi:10.1016/j.eswa.2007.11.051

Sarit C., Bikromadittya M., (June 2012). Spam Mail Filtering Technique using Different Decision Tree Classifiers through Data Mining Approach - A Comparative Performance Analysis. International Journal of Computer Applications (0975 – 888) Volume 47– No.16, 26. Retrieved from: https://pdfs.semanticscholar.org/2fbc/3021ecba5eadfaea280fc08b55c5ecd24aae.pdf

Amayri, O., & Bouguila, N. (2010). A study of spam filtering using support vector machines. Artificial Intelligence Review, 34(1), 73-108. doi:10.1007/s10462-010-9166-x

Sanghani, Gopi & Kotecha, Ketan. (2018). SUPPORT VECTOR MACHINE FOR PERSONALIZED E-MAIL SPAM FILTERING. Retrieved from: https://www.researchgate.net/publication/329116665\_SUPPORT\_VECTOR\_MACHINE\_FOR\_PERSONALIZED\_E-MAIL\_SPAM\_FILTERING

Trudgian C. D. & Yang R. Z., Spam Classification Using Nearest Neighbor Techniques. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.8884&rep=rep1&type=pdf

Ananthi S., Sathyabama S., (n.d.). SPAM FILTERING USING K – NN. Retrieved from: https://pdfs.semanticscholar.org/30fc/65335ced0e750cc03bfb0c8f981ad4011d4b.pdf

Puga, J. L., Krzywinski, M., & Altman, N. (2015). Bayes' theorem: Incorporate new evidence to update prior information. Nature Methods, 12(4), 277+. Retrieved from https://link-gale-com.elibrary.jcu.edu.au/apps/doc/A408649556/AONE?u=james\_cook&sid=AONE&xid=b005f447