HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOL OF INFORMATION TECHNOLOGY AND COMMUNICATION

-----


# MACHINE LEARNING & DATA MINING PROJECT REPORT

## Project name: Spam Email Filtering using Naïve Bayes Classification

**Supervisor** Nguyễn Nhật Quang

**Student names :** Lê Huy Hoàng 20194766
Nguyễn Vũ Minh 20194801

*Ha Noi, 06/2022*

## I. Description of the problem

We all know how frustrating email spamming might be. Having your mailbox full of spam makes every time your co-worker tells you to check your email become a hunt for a gem deep under the sea. For people who know not to care about those annoying junk mails, it just stops at being annoying. But others who are not familiar with the internet might find those spam interesting. Then they become a victim of scams or have virus infect their computers. Thus, email spam is a problem we can't just ignore and require a solution.

And so, in this project, we would like to utilize machine learning to identify spam emails and normal emails.

## II. Algorithm and Dataset

### 1. Algorithm

For the machine learning algorithm, we would like to use the Naïve Bayes classification approach.

Naïve Bayes classification is a simple probability algorithm based on the fact, that all features of the model are independent. In this problem, we will assume that every word in the email's content are independent and completely ignore the context (the order of words happen in a mail doesn't matter).

Our machine learning model will calculate the probabilities that the email is spam or not spam using the Bayes formula:

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

$$Bayes\ formula$$

Modify the formula a bit, we will then get:

$$P(Spam|Content) = \frac{P(Content|Spam) * P(Spam)}{P(Content)} \qquad (1)$$

Where:

| | |
|---|---|
| P(Spam\|Content) | ■ The probability the mail is spam knowing the content |
| P(Content\|Spam) | ■ The probability the mail have that content knowing it's a spam |
| P(Spam) | ■ The probability the mail is a spam regardless of its content |
| P(Content) | ■ The probability that a mail will have that content regardless whether it is spam or normal |

Do the same with normal email, we will also have:

$$P(Normal|Content) = \frac{P(Content|Normal) * P(Normal)}{P(Content)} \qquad (2)$$

In the spam email filtering, we have to classify an email to be spam or normal knowing its content. Which mean that we need to calculate $P(Spam|Content)$ and $P(Normal|Content)$. After that, we will compare the two probabilities to see which one come out on top, and the email will be classify as the one having the higher probabilities.

Now, we will go into detail how we calculate the parameters of the formula (the following is for $P(Spam|Content)$ but the same can be done for $P(Normal|Content)$:

- P(Spam)
  We will get this by calculate the probability of spam email in the dataset.
  $$P(Spam) = \frac{n_{Spam}}{n_{Spam} + n_{Ham}}$$
  Where $n_{Spam}$ – The number of spam mails in dataset
  $n_{Ham}$ – The number of normal mails in dataset
- P(Content|Spam)
  First, we divide the mail's content into separate independent words:
  Content = $w_1$ $w_2$ $w_3$ … $w_n$
  Then we can calculate the probability by:

$$P(Content|Spam) = \prod_{i=1}^{n} P(w_i \mid Spam)$$

Where $P(w_i \mid Spam)$ – The probability $w_i$ appear in the spam email list

$$P(w_i \mid Spam) = \frac{N_{w_i|Spam}}{N_{Spam}}$$

Where $N_{w_i|Spam}$ – The number of word $w_i$ appear in spam email

$N_{Spam}$ – The total number of word in all spam email

- P(Content)

  Since our main goal is to compute and compare $P(Spam|Content)$ and $P(Normal|Content)$, when we classify an email, $P(Content)$ in both (1) and (2) will be the same (the content used in both formulas are of that same email). So, we can exclude it out of the formula.

The formulas will have changes due to issues/difficulties that will be discussed in further part.

**2. Dataset**

The dataset we will be using is: https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv

The dataset contains 5172 rows with each row represent an email and 3002 columns in which the first column will contain Email name, the last column contains the label (1 for spam and 0 for normal) and the remaining 3000 columns are the 3000 most common words in all the emails.

We have prepared a list of stopwords to ignore when train the model

| | Number of emails | % in dataset |
|---|---|---|
| Spam email | 1500 | 29% |
| Normal email | 3672 | 71% |

## III. Results

### 1. Evaluation methods

We used k-fold cross-validation for our evaluation where k = 10.

Each subset is stratified sampling to keep the class distribution balance.

2. **Evaluation metrics**

| Spam | | Classified by the system | |
|---|---|---|---|
| | | Spam | Normal |
| True class | Spam | 1417 | 83 |
| | Normal | 214 | 3458 |

Confusion matrix for spam email

| Normal | | Classified by the system | |
|---|---|---|---|
| | | Normal | Spam |
| True class | Normal | 3458 | 214 |
| | Spam | 83 | 1417 |

Confusion matrix for normal email

$$Precision\ (Spam) = \frac{1417}{1417 + 214} = 86.88\ \%$$

$$Recall\ (Spam) = \frac{1417}{1417 + 83} = 94.47\ \%$$

$$Precision\ (Normal) = \frac{3458}{3458 + 83} = 97.66\ \%$$

$$Recall\ (Normal) = \frac{3458}{3458 + 214} = 94.17\ \%$$

$$=>\ Precision\ (Macro) = \frac{86.88 + 97.66}{2} = 92.27\ \%$$

$$=>\ Recall\ (Macro) = \frac{94.47+94.17}{2} = 93.32\ \%$$

$$=> F1 = \frac{2*92.27*93.32}{92.27+93.32} = 92.79\ \%$$

## IV. Issues / Difficulties

1. **Word appear only in one class**

In the dataset, some of the words only appear in spam and not in normal and vice versa.

Thus, making $P(w_i \mid Spam)$ $or$ $P(w_i \mid Normal)$ equal 0, resulting in $P(Spam|Content)$ $or$ $P(Normal|Content)$ = 0 and the prediction become faulty.

To fix this, we have to make sure that $N_{w_i|Spam}\ != 0$.

The solution we used in this machine learning system is to add an $\alpha$ number to all of $N_{w_i|Spam}$ (in our evaluation, we set $\alpha$ = 1). By doing this, $N_{Spam}$ will be increased by the amount we have added to $N_{w_i|Spam}$.

So, the final formula for $P(w_i \mid Spam)$ will be:

$$P(w_i \mid Spam) = \frac{N_{w_i|Spam} + \alpha}{N_{Spam} + \alpha * N_{Vocabulary}}$$

Where $N_{w_i|Spam}$ – The number of word $w_i$ appear in spam email

$N_{Spam}$ – The total number of word in all spam email

$N_{Vocabulary}$ – The total number of unique word in dataset

**2. Floating-point underflow**

Using the current formula, we realize that the result we got all become 0 since the number was too small that a 64-bits floating point number fail to present it.

Knowing that if a > b, then log(a) > log(b), we then choose to compute and compare log($P(Spam|Content)$) and log($P(Normal|Content)$) instead

In the end, we will need to calculate:

$$log\big(P(Spam|Content)\big) = log\big(P(Content|Spam) * P(Spam)\big)$$
$$= log\big(P(Spam)\big) + log\big(P(Content|Spam)\big)$$
$$= log\big(P(Spam)\big) + \sum_{i=1}^{n} \log(P(w_i \mid Spam))$$

And:

$$log\big(P(Ham|Content)\big) = log\big(P(Content|Ham) * P(Ham)\big)$$
$$= log\big(P(Ham)\big) + log\big(P(Content|Ham)\big)$$
$$= log\big(P(Ham)\big) + \sum_{i=1}^{n} log(P(w_i \mid Ham))$$

## V. Conclusions

After trying out the Naïve Bayes method for Email Spam Filtering, we are surprise to see an over 90% accuracy results. Still, the method still have a big problem of misclassify normal email as spam.