COURSE PROJECT REPORT

EMAIL SPAM FILTERING

1. **Group members:**

- ✓ Nguyễn Huy Hoàng – 20194433 – hoang.nh194433@sis.hust.edu.vn
- ✓ Phạm Như Thuấn – 20194456 – thuan.pn194456@sis.hust.edu.vn
- ✓ Lê Đức Anh – 20194416 – anh.ld194416@sis.hust.edu.vn

2. **Description of the problem:**

Electronic mail (email or e-mail) is a method of exchanging messages ("mail") between people using electronic devices and an email "spam" is an unsolicited bulk email. The problem is to classify emails as “spam” or “non-spam”.

- Input: Representation of the content of an email (e.g., a vector of keyword weights).
- Output: An assigned label of either “spam” or “normal”.

3. **Machine Learning Algorithm:** *Naïve Bayes classification*

- Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

a) Naïve Bayes Classifier

- Problem definition

  - A training set D, where each training instance x is represented as an n-dimentional attribute vector: (x_1, x_2, … ,x_n).
  - A pre-defined set of classes: C = {c_1, c_2, … , c_m).

- Given a new instance z, we want to find the most probable class for instance z.

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i | z)$$

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i | z_1, z_2, \ldots, z_n)$$

$$c_{MAP} = \arg\max_{c_i \in C} \frac{P(z_1, z_2, \ldots, z_n | c_i).P(c_i)}{P(z_1, z_2, \ldots, z_n)}$$ (By Bayes Theorem)

$$c_{MAP} = \arg\max_{c_i \in C} P(z_1, z_2, \ldots, z_n | c_i).P(c_i)$$

($P(z_1, z_2, \ldots, z_n)$ is the same for all classes)

- Assumption in Naïve Bayes classifier. The attributes are conditionally independent given classification

$$P(z_1, z_2, \ldots, z_n | c_i) = \prod_{j=1}^{n} P(z_j | c_i)$$

- Naïve Bayes classifier finds the most probable class for z.

$$c_{NB} = \arg\max_{c_i \in C} P(c_i).\prod_{j=1}^{n} P(z_j | c_i)$$

- The learning (training) phase (given a training set)

  For each classification (i.e., class label) $c_i \in C$

  - Estimate the prior probability: $P(c_i)$
  - For each attribute value x_j, estimate the probability of that attribute value given classification c_i: $P(x_j|c_i)$

- The classification phase (given a new instance)

  - For each classification $c_i \in C$, compute the formula

$$P(c_i).\prod_{j=1}^{n} P(x_j|c_i)$$

  - Select the most probable classification $c^{¿}$

$$c^{¿} = \arg\max_{c_i \in C} P(c_i).\prod_{j=1}^{n} P(x_j|c_i)$$

b) Distribution used to compute $P(x_j|c_i)$:

i. Gaussian Naïve Bayes:

- Gaussian Naïve Bayes implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

$$P(x_j|c) = P(x_j|\mu_c, \sigma_c^2) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(\frac{-(x_j-\mu_c)^2}{2\sigma_c^2}\right)$$

- The parameter μ_c, σ_c are estimated using maximum likelihood.

ii. Multinomial Naïve Bayes:

- Multinomial Naïve Bayes implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). The distribution is parametrized by vectors $\theta_c = (\theta_{c1}, \theta_{c2}, \ldots, \theta_{c1})$ for each class , where is the number of features (in text classification, the size of the vocabulary) and θ_{cj} is the propability $P(x_j|c)$ of feature I appearing in a sample belonging to class y.

- The parameters $\theta_c$is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{ci} = \frac{N_{ci} + \alpha}{N_c + \alpha n}$$

where $N_{ci} = \sum_{x \in T} x_i$is the number of times feature i appears in a sample of class y in the training set T, and $N_c = \sum_{i=1}^{n} N_{ci}$is the total count of all features for class y.

- The smoothing priors $\alpha > 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing.

iii. Bernoulli Naïve Bayes:

- Bernoulli Naïve Bayes implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate

Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a Bernoulli Naïve Bayes instance may binarize its input (depending on the binarize parameter).

- The decision rule for Bernoulli naive Bayes is based on

$$P(x_j|c) = P(j|c)x_j + (1 - P(j|c))(1 - x_j)$$

which differs from multinomial NB's rule in that it explicitly penalizes the non-occurrence of a feature i that is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature.

- In the case of text classification, word occurrence vectors (rather than word count vectors) may be used to train and use this classifier. Bernoulli Naïve Bayes might perform better on some datasets, especially those with shorter documents. It is advisable to evaluate both models, if time permits.

4. Dataset:

- General description:
  - Number of instance: 5729
  - Number of attributes: 2 (email content, nominal class label)
  - Missing attribute value: None
  - Class distribution:
    - Spam: 1368 (23.9%)
    - Non-spam: 4361 (76.1%)

- Attribute information:

  The first column of ‘dataset3.csv’ denotes whether the e-mail was considered spam (1) or not (0). The second column contain the content of the e-mail in text.

- Source:

  Spam-or-Ham-Email-Classification - Balakishan Molankula

5. Experimental results:

	Accuracy	Precision	Recall	F1-score
Multinomial NB	99.06%	99.62%	99.15%	99.39%
Gaussian NB	95.79%	95.63%	99.00%	97.29%
Bernoulli NB	96.64%	96.43%	99.31%	97.85%

Confusion Matrix:

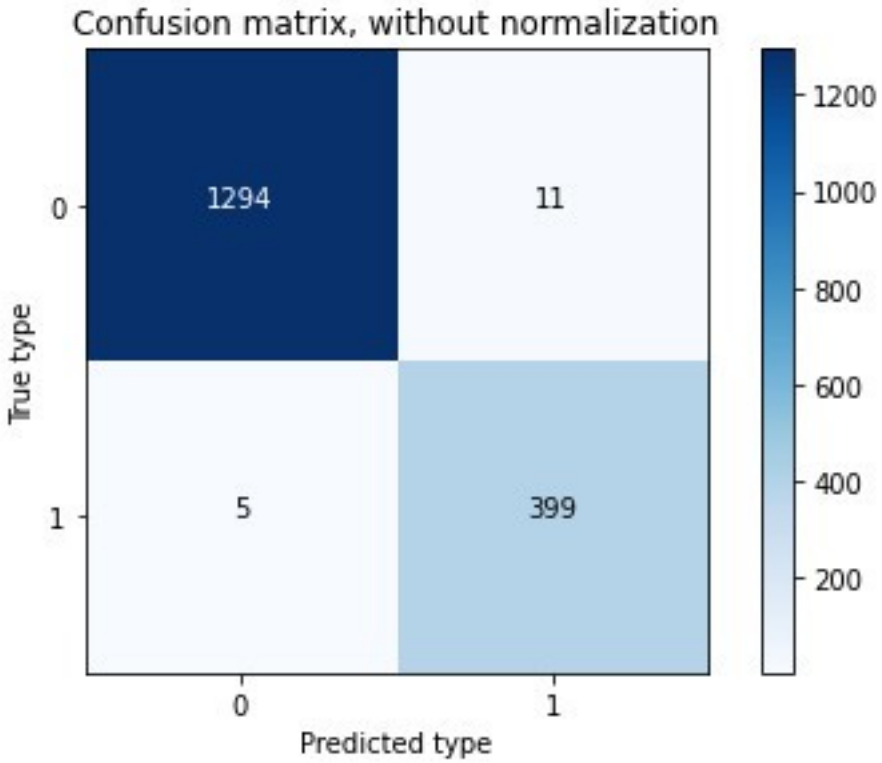

Figure 1: Multinomial Naive Bayes

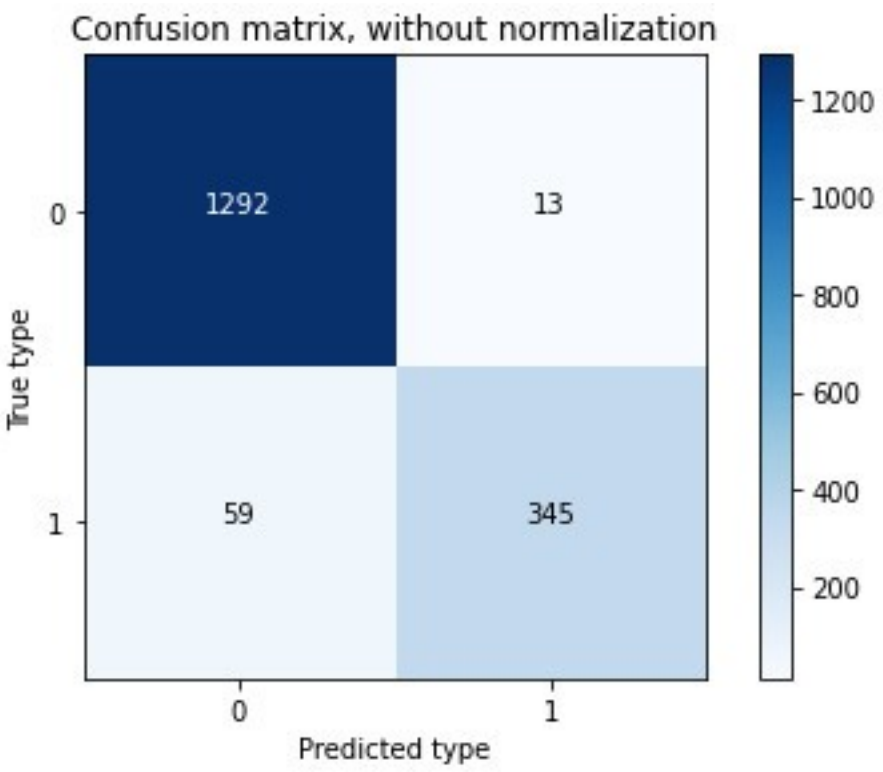

Figure 2: Gaussian Naive Bayes

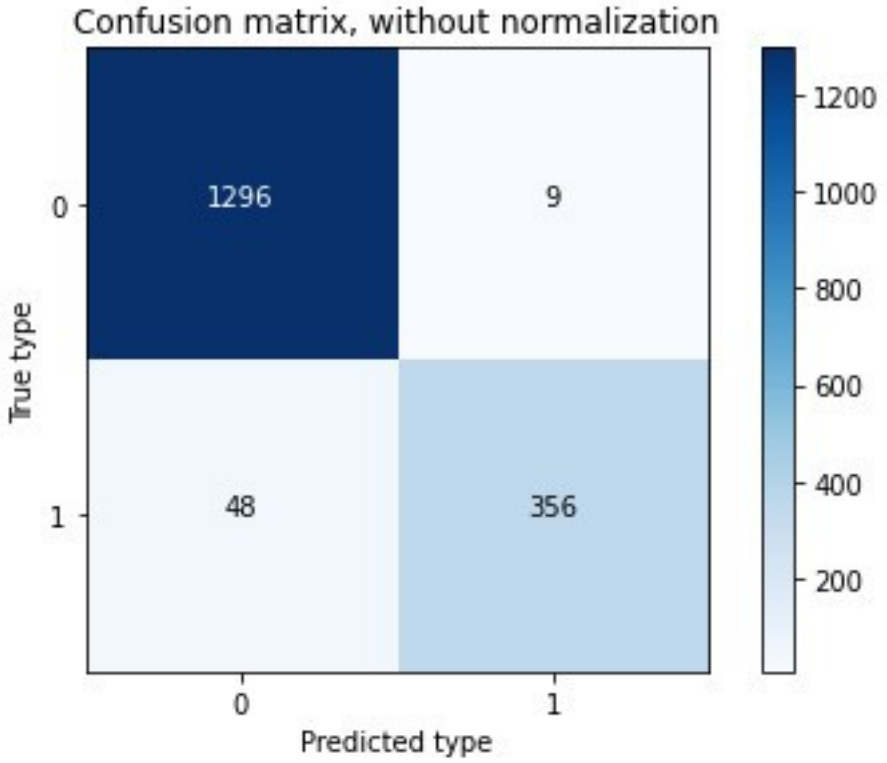

Figure 3: Bernoulli Naive Bayes

6. System's main function:

- Firstly, we use 'read_csv()' to read file data set.

- Function 'info()' is used to show out information of the imported data set, such as: type, name of columns, non-null elements, …

- In the pre-processing data part, we use these functions (imported from 'gensim' library):

  + strip_non_alphanum() is to remove words which are not alphanumeric.

  + strip() is to remove head and tail of mail (blank space).

  + lower() is to set all word in email to lowercase.

  + split_alphanum() is to split words combining characters and digits.

(like: ‘gh56’ → ‘gh’ & 45).

+ strip_short(minsize=2) is to remove single-character words.

+ strip_numeric() is to remove numbers.

+ word_tokenize() is to tokenize email into tokens.

- Function ‘WordNetLemmatize()’ of NLTK library is used to convert the suffix stripped words to their base forms.

- After finishing pre-processing data part, we use CountVectorizer(stop_words='english') function in order to vectorize email. At the same time, ‘stop_words’ is that we remove stop-word in English such as “a”, “the”, “is”, “are” (eliminate words that are so commonly used that they carry very little useful information).

-Train_test_split() function is used to separate data after vectorizing into train set and test set.

- Accuracy_score(), precision_score(), recall_score(), f1_score, confusion_matrix() are evaluating function in NB.

7. Difficulties:

- Preprocessing data:

- How to remove stop word, not important characters in email.
- How to tokenize.
- Process lemmatization
- Handle missing data

- Solution:

We had referred different source on the Internet then. After that each part was processed by a particular way that was the most understandable. Therefore, in the preprocessing procedure, we combined a lot of ways together and worked with many libraries (nltk, genism,….)

8. Conclusion:

- From the experimental results, it can be seen that all three model give good output. However, Multinomial Naïve Bayes has the best result with accuracy, precision, recall and F1-score are both above 99%. To sum up, we should use Multinomial Naïve Bayes in the Email Spam Filtering problem to have the best productivity.
- Improvement proposals:

In this dataset, there were no missing values so we did not have any solution to handle them. As a result, the application will perform poorly in the future if it works with a different dataset. Hence, we need to upgrade the system in processing missing data for the next version of the application.