FPT QUY NHON UNIVERSITY
ARTIFICIAL INTELLIGENCE

MAI391

COMPUTER PROJECT

Topic:

SPAM EMAIL DETECTION

Class:	AI18C	
Students:	Nguyen Tan Thang	QE180019
	Vo Quang Trieu	QE180113
	Nguyen Tran Truong Chinh	QE180191
	Nguyen Quoc Nhut	SE181753
	Pham Thi Kim Oanh	QE180218
	Do Thi Quynh Nga	QE180219

Quy Nhon, 7-2024

Contents

1 Introduction **2**
- 1.1 Problem Statement . . . 2
- 1.2 Overview of the Project . . . 2

2 Dataset **2**
- 2.1 Data Source . . . 2
- 2.2 Data Preprocessing . . . 2
- 2.3 Exploratory Data Analysis . . . 3
  - 2.3.1 Data Visualization . . . 3
- 2.4 Spam vs Non-Spam Email Characteristics . . . 4

3 Build Model **5**
- 3.1 Model Selection . . . 5
  - 3.1.1 Introduction to Naive Bayes . . . 5
  - 3.1.2 Multinomial Naive Bayes . . . 5
  - 3.1.3 Why Choose Multinomial Naive Bayes . . . 6
- 3.2 Text Preprocessing . . . 6
- 3.3 Text Vectorization . . . 6
- 3.4 Building and Training Multinomial Navie Bayes Model . . . 7

4 Evaluating Model Performance **8**
- 4.1 Method . . . 8
- 4.2 Evaluating . . . 8

5 Model Optimization **9**

6 Creating UI **10**

7 Conclusion **10**

8 References **10**

1. Introduction

1.1 Problem Statement

_ Spam emails cause significant disruption, security risks, and annoyance to users. This project aims to develop an effective spam email classification model using machine learning to improve the accuracy of detection and minimize the impact of spam on users.

1.2 Overview of the Project

_ The project aims to classify and identify spam emails using machine learning techniques. By analyzing and processing email content, the goal is to build model that accurately distinguish between spam (unwanted) and ham (legitimate) emails. Key steps include data preprocessing, feature extraction, model selection, and deployment for practical use. The objective is to empower users to identify and handle spam emails effectively.

2. Dataset

2.1 Data Source

_ Data link: https://www.kaggle.com/datasets/satyajeetbedi/email-hamspam-dataset

_ Kaggle provides a rich source of resources with thousands of datasets across various fields. We have searched and selected a dataset on this platform.

2.2 Data Preprocessing

_ Data set overview: The dataset has 5572 rows and 4 colums.

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

_ Clean data:

- Remove columns with all NaN values.
- Rename columns v1, v2 to label and message.
- Drop rows with null values.
- Drop duplicate values in the message column, keeping the first occurrence.
- Encode labels in the label column: 1 for spam and 0 for normal.

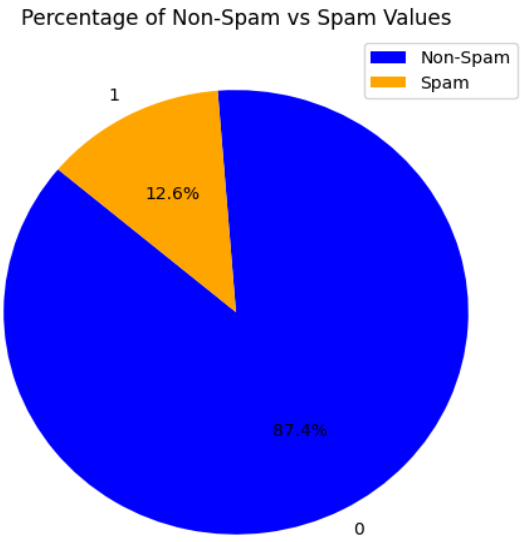
_ Cleaned dataset overview: The cleaned dataset has 5169 rows and 2 colums.

	label	message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	1	This is the 2nd time we have tried 2 contact u...
5568	0	Will Ì_ b going to esplanade fr home?
5569	0	Pity, * was in mood for that. So...any other s...
5570	0	The guy did some bitching but I acted like i'd...
5571	0	Rofl. Its true to its name

5169 rows × 2 columns

2.3 Exploratory Data Analysis

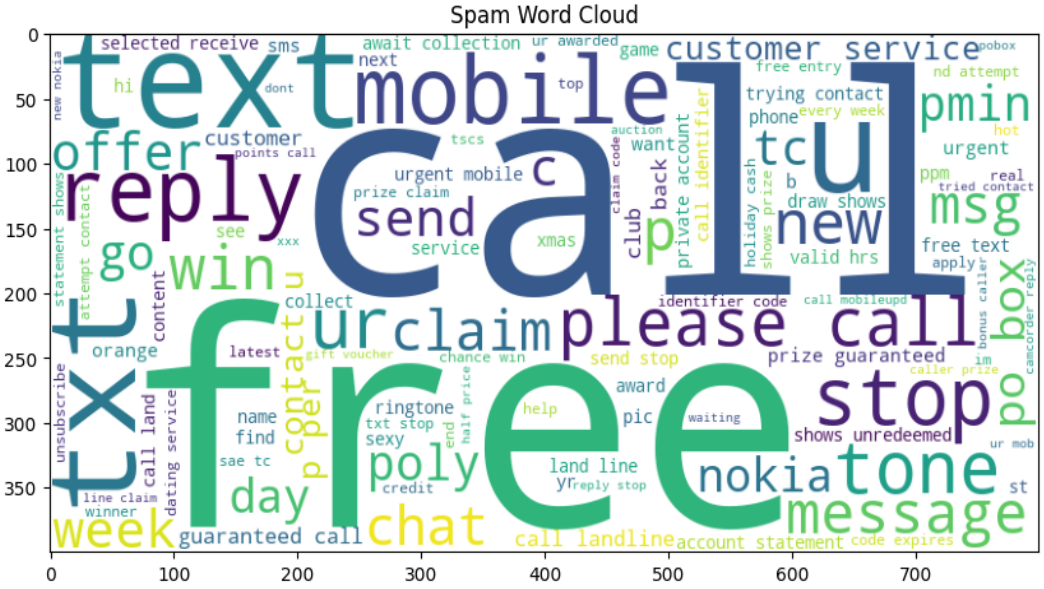
2.3.1 Data Visualization



_ From this pie chart, we can see that the percentage of spam emails in this dataset is 12.6% and the percentage of non-spam emails is 87.4%.

2.4 Spam vs Non-Spam Email Characteristics

_ Below is a chart showing the frequency of meaningful words appearing in spam and non-spam emails. Words with larger sizes indicate a higher frequency of occurrence, while words with smaller sizes indicate a lower frequency.



_ From the chart, we can see that in spam emails, words like 'call', 'free', 'reply', 'please', and 'stop' appear frequently, while in non-spam emails, words like 'im', 'u', 'ok', 'go', 'got', and 'time' appear frequently. This indicates a difference in text between spam and non-spam emails.

3. Build Model

3.1 Model Selection

3.1.1 Introduction to Naive Bayes

_ Naive Bayes is a classification algorithm based on Bayes' theorem with the "naive" assumption that features in the data are independent of each other. Multinomial Naive Bayes is a variant specifically designed to handle discrete data.

_ Naive Bayes' Formula:

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

$P(y|X)$(posterior probability): the probability of the target y given the feature X.

$P(X|y)$(likelihood): the probability of the feature X given by the target y.

$P(y)$: prior probability of the target y.

$P(X)$: probability of the feature X.

3.1.2 Multinomial Naive Bayes

_ Here, X represents the feature vector, which can be written as:

$$X = (x_1, x_2, x_3, ..., x_n)$$

_ In that case, Bayes' theorem becomes:

$$P(y|X) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

_ The probability of word occurrence in the text $P(x_i|y) is as flow$:

$$P(x_i|y) = \frac{N_i}{N_c}$$

N_i is the total number of times the word x_i appear in text.

N_c is the total number of times all words $x_1, x_2, ..., x_n$ appear in the text.

_ The formula above has a limitation: when the word x_i does not appear in the text at all, we have $N_i = 0$. This make $P(x_i|y) = 0$.

_ To overcome this issue, we use a technique called Laplace Smoothing by adding a constant to both the numerator and the denominator so that the value is always non-zero.

$$P(x_i|y) = \frac{N_i + \alpha}{N_c + d\alpha}$$

α (smoothing parameter) is typically a positive number, often equal to 1.

$d\alpha$ is added to the denominator to ensure $\sum_{i=1}^{d} P(x_i|y) = 1$, ussually is total numbers of words.

_ In this project, we compare the probabilities with each other, therefore we have:

$$P(y|X) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

3.1.3 Why Choose Multinomial Naive Bayes

_ **Efficiency**: It is computationally efficient and works well with large feature spaces.

_ **Suitability**: Handles text classification tasks effectively where features are typically word frequencies or counts.

_ **Simplicity**: Requires relatively few parameters to estimate.

3.2 Text Preprocessing

_ Remove URLs: Remove URLs within the text.

_ Lowercasing: Convert all text to lowercase to ensure consistency.

_ Removing special characters: Eliminate non-alphabetic and non-numeric characters.

_ Removing whitespace: Trim excess spaces from the text.

_ Removing stopwords: Filter out common words (e.g., "and", "the") that do not contribute to the meaning of the text.

_ Removing digits: Exclude numeric characters from the text.

→ Noise reduction and reducing data size make it easier to build models.

3.3 Text Vectorization

_ CountVectorizer is a feature extraction technique in Natural Language Processing (NLP) used to transform a collection of text documents into a numerical feature matrix. It counts the frequency of each word (or token) in the text data and represents this count as numerical values.

_ Example:

- Sample dataset:

	sentences
0	The quick brown fox jumps over the lazy dog
1	She sells seashells by the seashore
2	All work and no play makes Jack a dull boy
3	I love the smell smell smell of fresh flowers ...

- Sample dataset after being vectorized:

	boy	brown	dog	dull	flower	fox	fresh	jack	jump	lazy	love	make	play	quick	seashells	seashore	sell	smell	spring	work
0	0	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
2	1	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1
3	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	3	1	0

→ Apply this text vectorization technique to the entire dataset.

3.4 Building and Training Multinomial Navie Bayes Model

_ Based on the mathematical formulas, we can construct the Multinomial Naive Bayes model, but applying only those formulas could lead to underflow errors.

- **Underflow**: Because $P(X|y)$ is often the product of many probabilities $P(x_i|y)$, as the number of features x_i increases, the product of small probabilities can become extremely small, leading to numerical underflow issues on computers.

_ Solving underflow issue: using log transform.

- **Log transform** helps mitigate underflow issues in numerical computations by applying the logarithm function to values, transforming small values into larger ones for more stable calculations.

→ We got:

$$logP(X|y) = logP(x_1|y) + logP(x_2|y) + ... + logP(x_n|y)$$

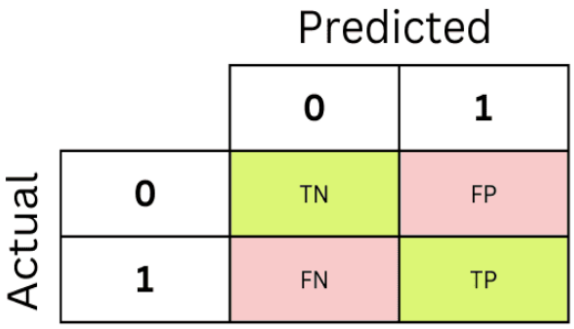
(In machine learning log mean logarit nepe)

→ The Multinomial Naive Bayes model has been built in the file **project.ipynb**.

4. Evaluating Model Performance

4.1 Method

_ **Confusion matrix**: Display the number of correct and incorrect predictions.



- TP (True Positive): The number of data points that actually belong to a class (actual class) and the model correctly predicts them to be in that class.

- TN (True Negative): The number of data points that do not belong to a class, and the model correctly predicts them not to belong to that class.

- FP (False Positive): The number of data points that do not belong to a class, but the model incorrectly predicts them to belong to that class (Type I error).

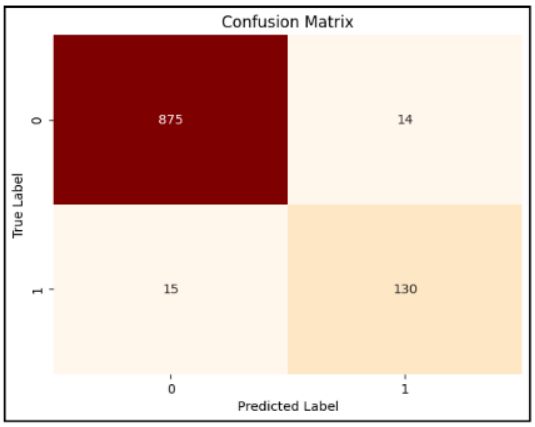
- FN (False Negative): The number of data points that belong to a class, but the model incorrectly predicts them not to belong to that class (Type II error).

_ **Accuracy score:** The ratio of the number of correct predictions to the total number of samples. The closer accuracy is to 1, the more accurate the model is.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

4.2 Evaluating

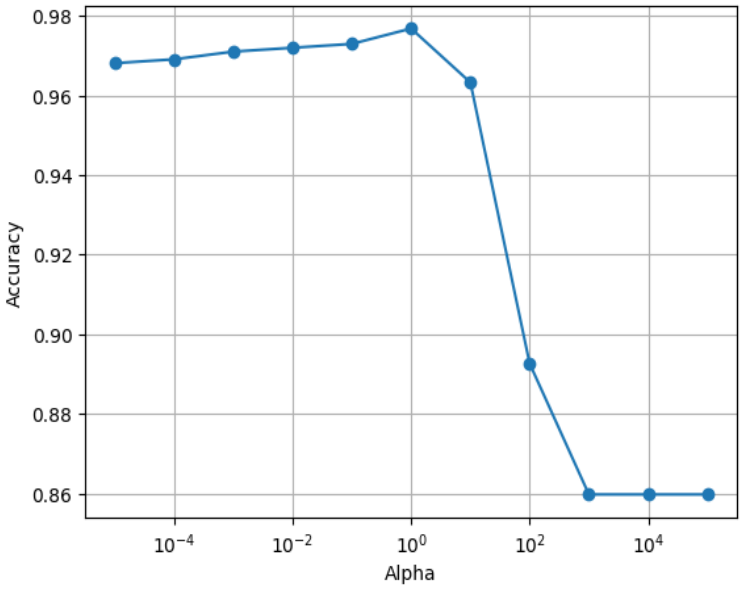
_ Confusion matrix and accurency of the model.



Accurency = 0.971

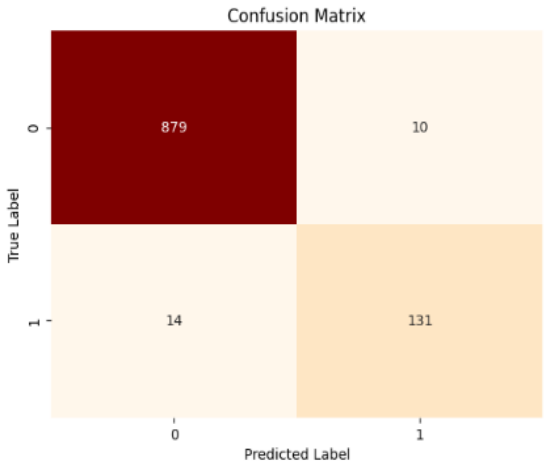
5. Model Optimization

_ Try smoothing parameter from 10^{-5} to 10^{5}.



→ "From the chart, we can see that when the smoothing parameter = 1, the accuracy is highest.

_ Confusion matrix after optimization.

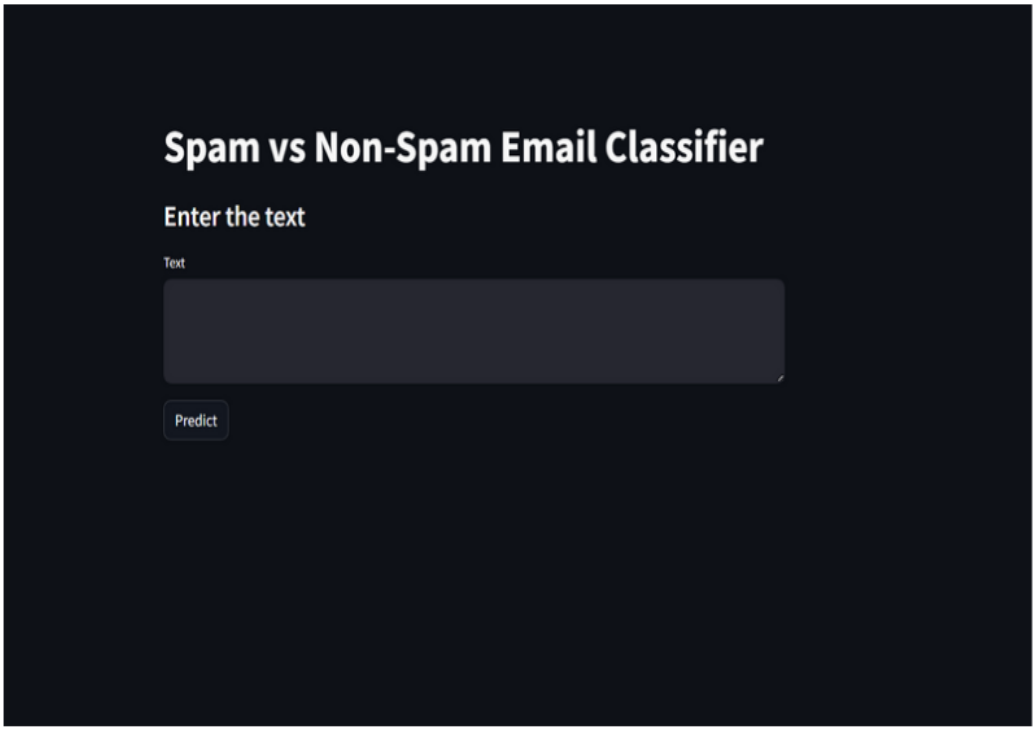


Accurency = 0.976

6. Creating UI

_ Use streamlit in python to create a simple web app.

- Input: Body of the email.
- Output: Spam or Non-Spam.



_ This web app is built in the file **webapp.py**.

7. Conclusion

_ Through this project, we have:

- Researched and understood the Naive Bayes algorithm.
- Text preprocessing before building the model.
- Building a Multinomial Naive Bayes model.
- Creating an interface to apply the model.

8. References

_ Multinomial Naive Bayes:

https://viblo.asia/p/mo-hinh-phan-lop-naive-bayes-vyDZO0A7lwj

https://viblo.asia/p/phan-lop-voi-navie-bayes-classification-mo-hinh-va-ung-dung-WAyK8PRkKxX