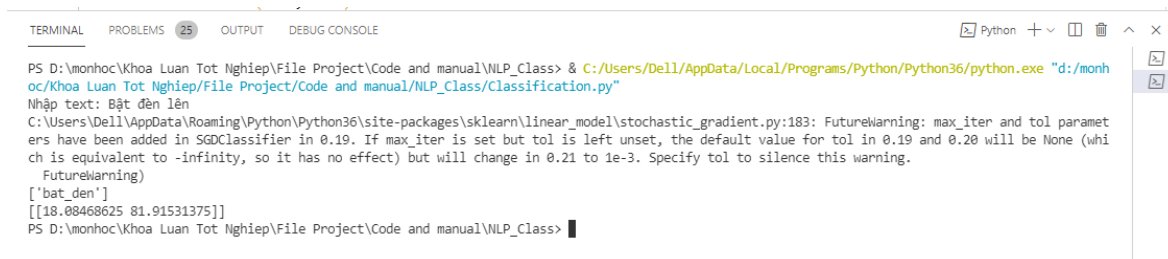


**Để mô tả cho quá trình huấn luyện và phân loại của model, tôi sẽ thực hành thí nghiệm test model với việc cho 2 nhãn (2 đầu ra) là ‘bat\_den’ và ‘bat\_bom’ như sau:**

Với văn bản thí nghiệm là “Bật đèn lên”, khi qua model Text Classification thì cho ra 2 nhãn (2 ngõ ra của model), model sẽ lựa chọn xác suất của ngõ ra nào là lớn nhất để đưa ra kết quả cuối cùng.



```
PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\WLP_Class> & C:/Users/Dell/AppData/Local/Programs/Python/Python36/python.exe "d:/monhoc/Khoa Luan Tot Nghiep/File Project\Code and manual\NLP_Class\Classification.py"
Nhập text: Bật đèn lên
C:\Users\Dell\AppData\Roaming\Python\Python36\site-packages\sklearn\linear_model\stochastic_gradient.py:183: FutureWarning: max_iter and tol parameters have been added in SGDClassifier in 0.19. If max_iter is set but tol is left unset, the default value for tol in 0.19 and 0.20 will be None (which is equivalent to -infinity, so it has no effect) but will change in 0.21 to 1e-3. Specify tol to silence this warning.
FutureWarning)
['bat_den']
[[18.08468625 81.91531375]]
PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\WLP_Class>
```

Hình 1 Kết quả phân loại văn bản cho thí nghiệm “Bật đèn lên”.

Và ở đây ta thấy model đã xác định được nhãn có xác suất cao nhất là nhãn “bat\_den” với tỉ lệ dự đoán chính xác lên đến 81,91%. (Hình 1)

Bây giờ hãy cùng tính lại xác suất để có đầu ra là nhãn “bat\_den” theo công thức Naive Bayes :

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (1)$$

Trong đó:

- $P(y|X)$  Xác suất của mục tiêu  $y$  với điều kiện có đặc trưng  $X$
- $P(X|y)$  Xác suất của đặc trưng  $X$  khi đã biết mục tiêu  $y$
- $P(y)$  Xác suất của mục tiêu  $y$
- $P(X)$  Xác suất của đặc trưng  $X$

Ở đây,  $X$  là vector các đặc trưng, có thể viết dưới dạng:

$$X = (x_1, x_2, x_3, \dots, x_n) \quad (2)$$

Baysian được xây dựng và dựa vào công thức dưới đây để tính rồi so sánh các kết quả lại với nhau để biết xác suất nào là cao hơn.

$$y = \operatorname{argmax} * P(x_1, x_2, \dots, x_n | c_j) P(c_j) \quad (3)$$

$P(c_j)$  ở đây được tính là tần suất xuất hiện của nhãn trên toàn bộ tập dữ liệu  $P(c_j)$  ở đây được tính là tần suất xuất hiện của nhãn trên toàn bộ tập dữ liệu.

$P(x_1, x_2, \dots, x_n | c_j)$  - Xác suất xảy ra đồng thời các điều kiện  $x_1, x_2, \dots, x_n$  khi nhãn  $c_j$  xảy ra. Thông thường việc tính này là bất khả thi (nhất là khi số lượng điều kiện  $n$  là lớn). Vậy nên thông thường, chúng ta sẽ coi các xác suất  $x_1, x_2, \dots, x_n$  là độc lập với nhau, từ đó:

$$P(x_1, x_2, \dots, x_n | c_j) = P(x_1 | c_j) * P(x_2 | c_j) * \dots * P(x_n | c_j) \quad (4)$$

$P(x_1 | c_j)$  được tính dựa trên tập dữ liệu có trước đó bằng số lần  $x_1$  xuất hiện cùng với  $c_j$  chia cho tổng số lần  $x_1$  xuất hiện.

Tóm lại bộ phân loại Bayes đơn giản sẽ được viết tổng quát lại như sau (viết lại biểu thức (3)):

$$y = \operatorname{argmax} * P(c_j) \prod_{i=1}^n P(x_i | c_j) \quad (5)$$

Bảng 1 Tập văn bản huấn luyện cho thí nghiệm test model

Ký hiệu	Văn bản huấn luyện	Nhãn
d1	Mở đèn lên	Bat_den
d2	Bật đèn đi	Bat_den
d3	Bật đèn nhanh đi	Bat_den
d4	mở đèn	Bat_den
d5	Bật máy bơm	Bat_bom
d6	Bật máy bơm đi	Bat_bom

d7	Bật bơm	Bat_bom
d8	Khởi động máy bơm	Bat_bom
d9	Bơm nước	Bat_bom

Để tính xác suất dựa theo công thức naive bayes classifier thì chúng ta cần thực hiện đúng các bước sau:

**Bước 1:** Tiền xử lý văn bản tệp dữ liệu huấn luyện.

- Từ tập dữ liệu huấn luyện ở bảng 1 ta thực hiện tách như mô tả ở bảng 2.

Bảng 2 Tách từ với tệp dữ liệu huấn luyện

Nhãn	Từ được tách trong tệp văn bản huấn luyện	Số câu đã huấn luyện
Bat_den	['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'bật_đèn', 'nhANH', 'đi', 'mở', 'đèn']	4
Bat_bom	['bật', 'máy_bơm', 'bật', 'máy_bơm', 'đi', 'bật_bơm', 'khởi_động', 'máy_bơm', 'bơm', 'nước']	5

- Lập từ điển với tập dữ liệu đã được tách từ (đảm bảo mỗi từ chỉ xuất hiện 1 lần):  
['mở', 'đèn', 'lên', 'bật\_đèn', 'đi', 'nhANH', 'bật', 'máy\_bơm', 'bật\_bơm', 'khởi\_động', 'bơm', 'nước']
- Từ điển của chúng ta có 12 từ, với tổng số câu huấn luyện là 9 câu.

Các kết quả tách từ và lập từ điển trên được tính toán bằng máy tính (Hình 2). Sử dụng thư viện ‘pyvi’ của python.



```
TERMINAL PROBLEMS 121 OUTPUT DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\NLP_Class> & C:/Users/Dell/AppData/Local/Programs/Python/Python36/python.exe "d:/monhoc/Khoa Luan Tot Nghiep/File Project/Code and manual/NLP_Class/tach_tu_pyvi.py"
Tách từ:
[['mở', 'đèn', 'lên'], ['bật_đèn', 'đi'], ['bật_đèn', 'nhANH', 'đi'], ['mở', 'đèn'], ['bật', 'máy_bom'], ['bật', 'máy_bom', 'đi'], ['bật_bom'], ['khởi_động', 'máy_bom'], ['bom', 'nước']]

Danh sách các từ được tách của tệp huấn luyện:
['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'bật_đèn', 'nhANH', 'đi', 'mở', 'đèn', 'bật', 'máy_bom', 'bật', 'máy_bom', 'đi', 'bật_bom', 'khởi_động', 'máy_bom', 'bom', 'nước']

Từ Điển của tệp huấn luyện:
['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'nhANH', 'bật', 'máy_bom', 'bật_bom', 'khởi_động', 'bom', 'nước']

Số từ có trong Từ Điển: 12

Tách từ cho văn bản test: Bật_đèn lên
PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\NLP_Class>
```

Hình 2 Kết quả tách từ và lập từ điển bằng máy tính.

**Bước 2:** Lập từ điển cho văn bản huấn luyện và tính tần suất xuất hiện của nhãn  $P(c_j)$  trên toàn bộ tập dữ liệu.

- Nhận thấy rằng có 2 class (nhãn) lần lượt là: ‘bat\_den’, ‘bat\_bom’.
- Dựa vào dữ liệu ở bảng 2 và kết quả tính toán bằng máy tính ở hình 2 ta tính được xác suất của các class trên:

*$P(class) = \text{tổng số câu thuộc class trong tệp huấn luyện} / \text{tổng số câu trong tệp huấn luyện}.$*

**$P(bat\_den)=4/9$ ,  $P(bat\_bom)=5/9$**

Từ điển  $V=\{ 'mở', 'đèn', 'lên', 'bật\_đèn', 'đi', 'nhANH', 'bật', 'máy\_bom', 'bật\_bom', 'khởi\_động', 'bom', 'nước' \}$

- Tổng số phần tử có trong từ điển là  $|V| = 12$

Nguyên lý của Bag of words là: đầu tiên xây dựng từ điển và tạo vector số cho các văn bản theo phương pháp túi đựng từ. Tất các từ trong văn bản cần được chuyển thành dạng biểu diễn số. Cách đơn giản nhất là xây dựng một bộ từ điển, sau đó thay thế từ đó bằng thứ tự xuất hiện trong từ điển. Mỗi vector có độ dài chính bằng số từ trong từ điển.

Khi này mỗi văn bản được biểu diễn bởi một vectors có độ dài  $\mathbf{d}$ , chính là giá trị thành phần thứ  $i$  xuất hiện trong văn bản đó.

Khi đó,  $P(x_i|c)$  tỉ lệ với tần suất từ thứ  $i$  (hay feature thứ  $i$  cho trường hợp tổng quát) xuất hiện trong các văn bản của class  $c$ . Giá trị này có thể tính bằng cách:

$$\text{Đặt } \lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c} \quad (6)$$

Trong đó :

- $N_{ci}$  là tổng số lần thứ  $i$  xuất hiện trong các văn bản của class  $c$ , nó được tính là tổng của tất cả các thành phần thứ  $i$  của feature vectors ứng với class  $c$ .
- $N_c$  là tổng số từ (kể cả lặp) xuất hiện trong class  $c$ . Nói cách khác nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào class  $c$ .

Tuy nhiên cách tính này có hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong class  $c$  thì biểu thức (6) sẽ bằng 0 sẽ dẫn đến biểu thức số (1) bằng 0. Để giải quyết vấn đề này, tôi đã áp dụng một kỹ thuật có tên là smoothing:

$$\widehat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{N_c + d\alpha} \quad (7)$$

Với  $\alpha$  là một số dương, thường là 1, để tránh trường hợp tử số bằng 0. Mẫu số được cộng với  $d\alpha$  để đảm bảo tổng xác suất  $\sum_{i=1}^d \widehat{\lambda}_{ci} = 1$ .

**Bước 3:** Huấn luyện và phân loại.

- ✚ Minh họa quá trình huấn luyện và dự đoán khi sử dụng Multinomial Naive Bayes theo bảng 3 và bảng 4, trong đó có sử dụng Laplace smoothing với  $\alpha = 1$ . Áp dụng thuật toán Bag of words (Trích xuất đặc trưng) ở biểu thức số (7)

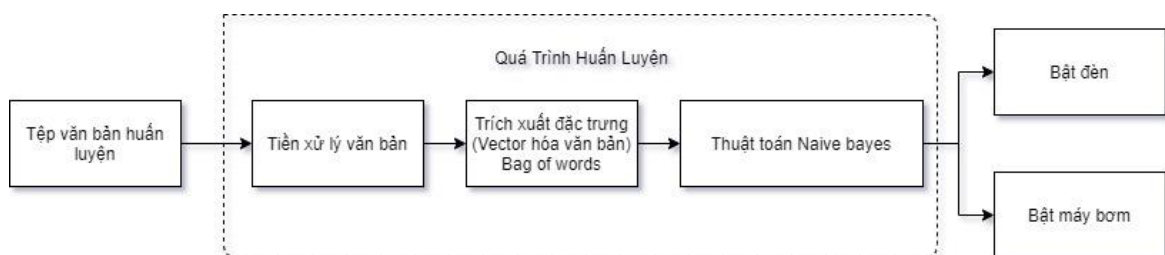
Bảng 3 Minh họa quá trình huấn luyện khi sử dụng Multinomial Naive Bayes cho nhãn “bat\_den” thí nghiệm test model.

STT	Từ trong từ điển	d1: $x_1$	d2: $x_2$	d3: $x_3$	d4: $x_4$	Total	$\Rightarrow \lambda_{bd}$
1	Mở	1	0	0	1	2	3/22
2	đèn	1	0	0	1	2	3/22
3	Lên	1	0	0	0	1	2/22
4	Bật_đèn	0	1	1	0	2	3/22
5	Đi	0	1	1	0	2	3/22
6	Nhanh	0	0	1	0	1	2/22
7	Bật	0	0	0	0	0	1/22
8	Máy_bơm	0	0	0	0	0	1/22
9	Bật_bơm	0	0	0	0	0	1/22
10	Khởi_động	0	0	0	0	0	1/22
11	Bơm	0	0	0	0	0	1/22
12	Nước	0	0	0	0	0	1/22
$ V $ =12						$N_{bd}$ = 10	$N_{bd} +  V $ = 22

Bảng 4 Minh họa quá trình huấn luyện khi sử dụng Multinomial Naive Bayes cho nhãn “bat\_bom” thí nghiệm test model.

STT	Từ trong từ điển	$d5:x_5$	$d6:x_6$	$d7:x_7$	$d8:x_8$	$d9:x_9$	Total	$\Rightarrow \lambda_{bb}$
1	Mở	0	0	0	0	0	0	1/22
2	đèn	0	0	0	0	0	0	1/22
3	Lên	0	0	0	0	0	0	1/22
4	Bật_đèn	0	0	0	0	0	0	1/22
5	Đi	0	1	0	0	0	1	2/22
6	Nhanh	0	0	0	0	0	0	1/22
7	Bật	1	1	0	0	0	2	3/22
8	Máy_bom	1	1	0	1	0	3	4/22
9	Bật_bom	0	0	1	1	0	2	3/22
10	Khởi_động	0	0	0	0	0	0	1/22
11	Bom	0	0	0	0	1	1	2/22
12	Nước	0	0	0	0	1	1	2/22
$ V $ =12							$N_{bd}$ = 10	$N_{bb} +  V $ = 22

Mô tả quá trình huấn luyện với 2 lớp đầu ra ‘bat\_den’ và ‘bat\_bom’ tại hình 3



Hình 3 Mô tả quá trình huấn luyện với 2 lớp đầu ra.

## 🚦 Quá trình dự đoán văn bản:

Văn bản cần dự đoán của chúng ta là : **“Bật đèn lên”** gọi là d10, sau khi tách từ cho câu này ta được 2 từ: **‘Bật\_đèn’** và **‘lên’**. (Dựa vào kết quả của máy tính ở hình 2)

⇒ Vector đặc trưng của văn bản d10:  $x_{10} = [0,0,1,1,0,0,0,0,0,0,0,0]$ , có số chiều là 12 được biểu diễn theo 12 từ trong từ điển của model được tính ở bước 2. Mỗi phần tử đại diện cho số từ tương ứng xuất hiện trong văn bản d10.

Dựa theo biểu thức tổng quát (5) và kết quả ở bảng 3, bảng 4 ta tính xác suất cho các nhãn:

- Tính xác suất cho đầu ra nhãn ‘bat\_den’:

$$P(\text{bat\_den} | d10) \propto P(\text{bat\_den}) \prod_{i=1}^d P(x_i | \text{bat\_den}) = \\ P(\text{bat\_den}) * P(\text{lên} | \text{bat\_den}) * P(\text{Bật\_đèn} | \text{bat\_den}) = 4/9 * 2/22 * 3/22 = 2/363$$

- Tính xác suất cho đầu ra nhãn ‘bat\_bom’:

$$P(\text{bat\_bom} | d10) \propto P(\text{bat\_bom}) \prod_{i=1}^d P(x_i | \text{bat\_bom}) = \\ P(\text{bat\_bom}) * P(\text{lên} | \text{bat\_bom}) * P(\text{Bật\_đèn} | \text{bat\_bom}) = 5/9 * 1/22 * 1/22 \\ = 5/3872$$

Ta thấy:  $P(\text{bat\_den} | d10) > P(\text{bat\_bom} | d10) \Rightarrow d10 \in \text{class}(\text{bat\_den})$

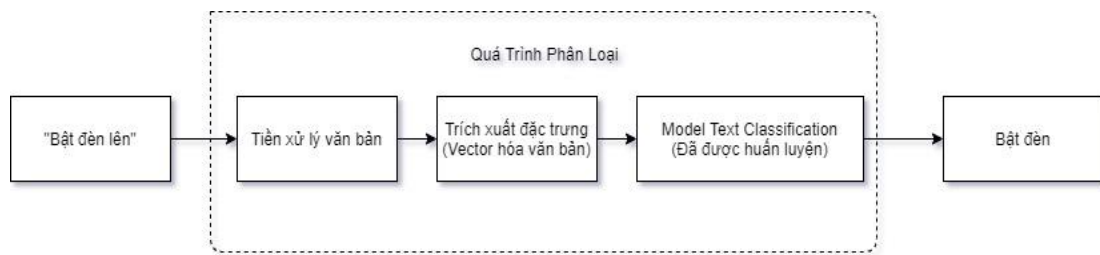
**Xác suất cần tìm:**

$$P(\text{bat\_den} | d10) = \frac{\frac{2}{363}}{\left(\frac{2}{363} + \frac{5}{3872}\right)} = 0,8101 \\ P(\text{bat\_bom} | d10) = \frac{\frac{5}{3872}}{\left(\frac{2}{363} + \frac{5}{3872}\right)} = 0,1899$$

Mô tả quá trình phân loại văn bản với đầu vào là các từ trong văn bản đó.

[‘Bật\_đèn’, ‘lên’] (2 input) của văn bản test **“Bật đèn lên”**: (Hình 4)





Hình 4 Mô tả quá trình phân loại với đầu vào là vector của văn bản test

Vậy với câu “Bật đèn đi” được đưa vào model sẽ có kết quả dự đoán cho nhãn Bật đèn (bat\_den) là : 81,01% , và cho nhãn bật bom (bat\_bom) là: 18,99%. Kết quả tính tay đúng với kết quả dự đoán của máy tính ở Hình 1.