

# Naïve Bayes Classification

# Nội dung

- Giới thiệu Naïve Bayes Classification (NBC)
- Mô hình toán
- Các dạng phân phối dùng trong NBC
- Các Ví dụ
- Bài Tập

# Giới Thiệu

- Naïve Bayes Classification (NBC) là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê.
- Thuộc vào nhóm supervised learning

# Giới Thiệu

- ❖ Thuật toán Naïve Bayes Classification được áp dụng vào các loại ứng dụng sau:
  - Real time Prediction: NBC chạy khá nhanh nên nó thích hợp áp dụng ứng dụng nhiều vào các ứng dụng chạy thời gian thực, như hệ thống cảnh báo, các hệ thống trading ...
  - **Text classification/ Spam Filtering/ Sentiment Analysis:** NBC cũng rất thích hợp cho các hệ thống phân loại văn bản hay ngôn ngữ tự nhiên vì tính chính xác của nó lớn hơn các thuật toán khác. Ngoài ra các hệ thống chống thư rác cũng rất ưu chuộng thuật toán này. Và các hệ thống phân tích tâm lý thị trường cũng áp dụng NBC để tiến hành phân tích tâm lý người dùng ưu chuộng hay không ưu chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.
  - ...

# Bayes's theorem

- Gọi A, B là hai sự kiện (event)

Với  $P(B) > 0$ :

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

Suy ra:

$$P(AB) = P(A | B)P(B) = P(B | A)P(A)$$

Công thức Bayes:

$$P(B | A) = \frac{P(AB)}{P(A)} = \frac{P(A | B)P(B)}{P(A)} = \frac{P(A | B)P(B)}{P(AB) + P(\overline{A}B)}$$

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

# Bayes's theorem

- Công thức Bayes tổng quát

Với  $P(A) > 0$  và  $\{B_1, B_2, \dots, B_n\}$  là một hệ đầy đủ các biến cố:

- ▣ Tổng xác suất của hệ bằng 1:

$$\sum_{k=1}^n P(B_k) = 1$$

- ▣ Từng đôi một xung khắc:

$$P(B_i \cap B_j) = 0$$

Khi đó ta có:

$$\begin{aligned} P(B_k | A) &= \frac{P(A | B_k) P(B_k)}{P(A)} \\ &= \frac{P(A | B_k) P(B_k)}{\sum_{i=1}^n P(A | B_i) P(B_i)} \end{aligned}$$

# Bayes's theorem

Công thức Bayes:

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

*Posterior ~ likelihood x prior*

Trong đó:

$P(A)$ : gọi là evidence (cố định, có thể xem như hằng số)

$P(B)$ : gọi là prior probability (xác suất tiên nghiệm) là phân phối xác suất trên A

$P(A|B)$  : gọi là likelihood, thể hiện độ phù hợp của A đối với những giá trị B khác nhau

$P(B|A)$ : gọi là posterior probability (xác suất hậu nghiệm) phản ánh sự ước lượng cho B khi đã biết A.

# Naïve Bayes Classification

## ❖ Mô hình:

- Giả sử có tập huấn luyện chứa các  $N$  mẫu  $\mathbf{x} = \{x_1, x_2, \dots, x_d\} \in \mathbb{R}^d$ .
- Giả sử có  $C$  classes:  $c \in \{1, 2, \dots, C\}$ .
- Hãy tính xác suất để điểm dữ liệu này rơi vào class  $c$ : Tính  $\mathbf{p(c|x)}$  ??? , nghĩa là tính xác suất để đầu ra là class  $c$  biết rằng đầu vào là vector  $\mathbf{x}$  (đây chính là posterior probability)
- Từ đó, có thể giúp xác định class của điểm dữ liệu  $\mathbf{x}$  đó bằng cách chọn ra class có xác suất cao nhất

$$c = \arg \max_{c \in \{1, \dots, C\}} p(c|\mathbf{x})$$

# Naïve Bayes Classification

- Dựa vào lý thuyết Bayes:

$$\begin{aligned}c &= \arg \max_c p(c|\mathbf{x}) \\&= \arg \max_c \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})} \\&= \arg \max_c p(\mathbf{x}|c)p(c)\end{aligned}$$

vì mẫu số  $p(\mathbf{x})$  (evidence) không phụ thuộc vào  $c$

Trong đó :

- $p(c|\mathbf{x})$  : posterior probability
- $p(c)$ : prior probability - ước lượng bằng  $|N_i|/|N|$ , trong đó  $N_i$  là tập các phần tử dữ liệu thuộc lớp  $c_i$ .
- $p(\mathbf{x}|c)$ : likelihood, tức phân phối của các điểm dữ liệu trong class  $c$ , thường rất khó tính toán vì  $\mathbf{x}$  là một biến ngẫu nhiên nhiều chiều, cần rất rất nhiều dữ liệu training để có thể xây dựng được phân phối đó

# Naïve Bayes Classification

- Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toán  $\mathbf{p}(\mathbf{x}|\mathbf{c})$  là rất lớn.
- Đó đó có thể giảm độ phức tạp của thuật toán Naïve Bayes giả thiết các thuộc tính **độc lập nhau**. Khi đó,

$$p(\mathbf{x}|c) = p(x_1, x_2, \dots, x_d|c) = \prod_{i=1}^d p(x_i|c)$$

# Naïve Bayes Classification

- Tại sao gọi là Naïve Bayes (ngây thơ)
  - các thuộc tính (biến) có độ quan trọng như nhau
  - các thuộc tính (biến) độc lập có điều kiện
- Nhận xét
  - giả thiết các thuộc tính độc lập không bao giờ đúng
  - nhưng trong thực tế, Naïve Bayes cho kết quả khá tốt
- Cách xác định class của dữ liệu dựa trên giả thiết này có tên là Naive Bayes Classifier (NBC).
- NBC có tốc độ training và test rất nhanh. Việc này giúp nó mang lại hiệu quả cao trong các bài toán large-scale.

# Naïve Bayes Classification

- Ở bước **training**, việc tính toán prior  $p(c)$  và likelihood  $p(x|c)$  sẽ dựa vào tập huấn luyện (training data).
  - Việc xác định các giá trị này có thể dựa vào **Maximum Likelihood Estimation** hoặc **Maximum A Posteriori**.
- \* **Tìm hiểu thêm về : Maximum Likelihood Estimation và Maximum A Posteriori.**

# Naïve Bayes Classification

- Ở bước testing, với một điểm dữ liệu mới  $x$ , class của nó sẽ được xác định bởi:

$$c = \arg \max_{c \in \{1, \dots, C\}} p(c) \prod_{i=1}^d p(x_i | c)$$

# Naïve Bayes Classification

- Việc tính toán  $p(\mathbf{x}_i|\mathbf{c})$  phụ thuộc vào loại dữ liệu (liên tục hay rời rạc)
- Có ba loại được sử dụng phổ biến là: Gaussian Naive Bayes, Multinomial Naive Bayes, và Bernoulli Naive .

# Các Phân Phối Thường Dùng Cho Likelihood

- ❖ Gaussian Naive Bayes (dùng cho dữ liệu liên tục – dạng số) : Mô hình này được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các **biến liên tục**.
- Với mỗi chiều dữ liệu  $i$  và một class  $c$ ,  $x_i$  tuân theo một **phân phối chuẩn** có kỳ vọng  $\mu_{ci}$  và phương sai  $\sigma_{ci}^2$ :

$$p(x_i|c) = p(x_i|\mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right)$$

- Trong đó, bộ tham số  $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$  được xác định bằng Maximum Likelihood

$$(\mu_{ci}, \sigma_{ci}^2) = \arg \max_{\mu_{ci}, \sigma_{ci}^2} \prod_{n=1}^N p(x_i^{(n)} | \mu_{ci}, \sigma_{ci}^2)$$

# Các Phân Phối Thường Dùng Cho Likelihood

- ❖ Multinomial Naive Bayes (dùng cho dữ liệu rời rạc)
- Khi đó,  $p(\mathbf{x}_i|\mathbf{c})$  tỉ lệ với tần suất thuộc tính thứ  $i$  (hay feature thứ  $i$  cho trường hợp tổng quát) xuất hiện trong các data của class  $c$ . Giá trị này có thể được tính bằng cách:

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

Trong đó:  $N_{ci}$  : là tổng số lần xuất hiện của thuộc tính  $i$  trong data của lớp  $c$

$N_c$ : là tổng số data của lớp  $c$

- Vấn đề: nếu có thuộc tính  $i$  không xuất hiện trong lớp  $c$  -> dẫn tới xác suất sẽ bằng 0

# Các Phân Phối Thường Dùng Cho Likelihood

- ❖ Multinomial Naive Bayes (dùng cho dữ liệu rời rạc)
- Khắc phục trường hợp xác suất = 0, dùng Laplace Smoothing bằng cách cộng thêm vào cả tử và mẫu để giá trị luôn khác 0.

$$\hat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{N_c + d\alpha}$$

Trong đó:

$\alpha$  thường là số dương, bằng 1.

$d\alpha$  được cộng vào mẫu để đảm bảo  $\sum_{i=1}^d P(x_i|y) = 1$

- Như vậy, mỗi class  $c$  sẽ được mô tả bởi bộ các số dương có tổng bằng 1:

$$\hat{\lambda}_c = \{\hat{\lambda}_{c1}, \dots, \hat{\lambda}_{cd}\}$$

# Các Phân Phối Thường Dùng Cho Likelihood

- ❖ Bernoulli Naive Bayes: Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary - bằng 0 hoặc 1.
- Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của 1 từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.
- Khi đó,  $p(\mathbf{x}_i|\mathbf{c})$  được tính bằng:

$$p(x_i|c) = p(i|c)^{x_i} (1 - p(i|c))^{1-x_i}$$

với  $p(i|c)$  có thể được hiểu là xác suất thuộc tính thứ  $i$  xuất hiện trong các data của class  $c$ .

# Ví dụ: Multinomial Naive Bayes

- Dữ liệu weather, dựa trên các thuộc tính (Outlook, Temp, Humidity, Windy), quyết định (play/no)

Dữ liệu rời rạc

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

# Ví dụ

- Tính prior và likelihood

Outlook			Temperature			Humidity			Windy			Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

$$P(\text{yes}) = 9/14$$

$$P(\text{no}) = 5/14$$

Prior

■ quyết định (play=yes/no)

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Tính xác suất hậu nghiệm

$$P(\text{yes} | X = \{\text{Sunny, Cool, High, True}\}) = ???$$

$$P(\text{no} | X = \{\text{Sunny, Cool, High, True}\}) = ???$$

# Ví dụ

$$P(\text{yes}) = 9/14$$

$$P(\text{no}) = 5/14$$

Outlook			Temperature			Humidity			Windy			Play	
Yes No			Yes No			Yes No			Yes No			Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

■ quyết định (play=yes/no)

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

- $P(\text{yes}|X=\{\text{Sunny,Cool,High,True}\}) \sim P(X=\{\text{Sunny,Cool,High,True}\}|\text{yes}) \times P(\text{yes}) = 0.0053$
- $P(\text{no}|X=\{\text{Sunny,Cool,High,True}\}) \sim P(X=\{\text{Sunny,Cool,High,True}\}|\text{no}) \times P(\text{no}) = 0.026$

$$\text{Likelihood}(\text{yes}) = P(X=\{\text{Sunny,Cool,High,True}\}|\text{yes})$$

$$= P(\text{Sunny}|\text{yes}) \times P(\text{Cool}|\text{yes}) \times P(\text{High}|\text{yes}) \times P(\text{True}|\text{yes})$$

$$= 2/9 \times 3/9 \times 3/9 \times 3/9$$

$$\text{Likelihood}(\text{no}) = P(\{\text{Sunny,Cool,High,True}\}|\text{no})$$

$$= P(\text{Sunny}|\text{no}) \times P(\text{Cool}|\text{no}) \times P(\text{High}|\text{no}) \times P(\text{True}|\text{no})$$

$$= 3/5 \times 1/5 \times 4/5 \times 3/5$$

Xác suất :

$$P(\text{yes}) = 0.0053 / (0.0053 + 0.0206) = 0.205$$

$$P(\text{no}) = 0.0206 / (0.0053 + 0.0206) = 0.795$$

Kết quả là : no

# Xác suất = 0

- Giá trị của thuộc tính không xuất hiện trong tất cả các lớp (“Outlook=Overcast” của lớp “no”)
  - Probability will be zero!
  - A posteriori probability will also be zero!
- Sử dụng Laplace estimator
- Xác suất không bao giờ có giá trị 0
- ❖ Laplace estimator

Ví dụ : thuộc tính outlook cho lớp yes

$$\frac{2 + \mu/3}{9 + \mu}$$

***Sunny***

$$\frac{4 + \mu/3}{9 + \mu}$$

***Overcast***

$$\frac{3 + \mu/3}{9 + \mu}$$

***Rainy***

# Giá trị thuộc tính nhiều/ thiếu thông tin

- Quá trình học: bỏ qua dữ liệu nhiều
- Quá trình phân lớp : bỏ qua các thuộc tính nhiều
- Ví dụ

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

$$\text{Likelihood}(\text{yes}) = 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$$

$$\text{Likelihood}(\text{no}) = 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$$

$$P(\text{yes}) = 0.0238 / (0.0238 + 0.0343) = 41$$

$$P(\text{no}) = 0.0343 / (0.0238 + 0.0343) = 59$$

# Ví dụ Gaussian Naive Bayes (Dữ liệu liên tục)

- Khi dữ liệu liên tục, cần có cách tính cho likelihood

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	78	false	yes
rain	70	96	false	yes
rain	68	80	false	yes
rain	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rain	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rain	71	80	true	no

# Ước Lượng Likelihood Dựa Vào Hàm Phân Phối Xác Suất

- ❖ Giả sử các thuộc tính có phân phối Gaussian
- Hàm mật độ xác suất được tính như sau

- mean

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

- standard deviation

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2$$

- hàm mật độ xác suất  $f(x) \sim \text{likelihood}$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

# Dữ liệu liên tục

- Ví dụ

Outlook			Temperature		Humidity		Windy		Play				
	yes	no		yes	no		yes	no		yes	no		
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std. dev.	6.2	7.9	std. dev.	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

$$\text{ví dụ : } f(\text{temperature} = 66 \mid \text{yes}) = \frac{1}{\sqrt{2\pi}6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

# Dữ liệu liên tục

phân lớp

Outlook	Temp.	Humidity	Windy	Play
Sunny	66	90	true	?

$$\text{Likelihood}(\text{yes}) = 2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$$

$$\text{Likelihood}(\text{no}) = 3/5 \times 0.0291 \times 0.0380 \times 3/5 \times 5/14 = 0.000136$$

$$P(\text{yes}) = 0.000036 / (0.000036 + 0.000136) = 20.9$$

$$P(\text{no}) = 0.000136 / (0.000036 + 0.000136) = 79.1$$

# Ví dụ: Phân loại email

- Bài toán phân loại mail Spam (S) và Not Spam (N).
- Ta có bộ training data gồm E1, E2, E3. Cần phân loại E4.
- Bảng từ vựng:  $[w_1, w_2, w_3, w_4, w_5, w_6, w_7]$ .
- Số lần xuất hiện của từng từ trong từng email tương ứng như bảng dưới.

	Email	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	Label
Training data	E1	1	2	1	0	1	0	0	N
	E2	0	2	0	0	1	1	1	N
	E3	1	0	1	1	0	2	0	S
Test data	E4	1	0	0	0	0	0	1	?

# Ví dụ: Phân loại email

- Tính được Prior probability

$$P(S) = \frac{1}{3}, P(N) = \frac{2}{3}$$

- Sử dụng Laplace Smoothing với  $\alpha=1$  ta tính được xác suất xuất hiện của từng từ trong văn bản như sau:

class = Spam (S)

	Email	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$
	E <sub>3</sub>	1	0	1	1	0	2	0
$P(w_i S)$	(trước Smoothing)	$1/5$	$0/5$	$1/5$	$1/5$	$0/5$	$2/5$	$0/5$
$P(w_i S)$	(sau Smoothing)	$2/12$	$1/12$	$2/12$	$2/12$	$1/12$	$3/12$	$1/12$

class = Not Spam (N)

	Email	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$
	E <sub>1</sub>	1	2	1	0	1	0	0
	E <sub>2</sub>	0	2	0	0	1	1	1
	Tổng	1	4	1	0	2	1	1
$P(w_i N)$	(trước Smoothing)	$1/10$	$4/10$	$1/10$	$0/10$	$2/10$	$1/10$	$1/10$
$P(w_i N)$	(sau Smoothing)	$2/17$	$5/17$	$2/17$	$1/17$	$3/17$	$2/17$	$2/17$

# Ví dụ: Phân loại email

- Vậy ta tính được:

Test data	E4	1	0	0	0	0	0	1	?
-----------	----	---	---	---	---	---	---	---	---

$$\begin{aligned}P(S|E4) &\propto P(S) \prod_{i=1}^7 P(w_i|S) \\&\propto \frac{1}{3} \times \left( \frac{2}{12} \times \frac{1}{12} \right) \\&\propto 0.0046\end{aligned}$$

$$\begin{aligned}P(N|E4) &\propto P(N) \prod_{i=1}^7 P(w_i|N) \\&\propto \frac{2}{3} \times \left( \frac{2}{17} \times \frac{2}{17} \right) \\&\propto 0.0092\end{aligned}$$

Vậy xác suất tương ứng sẽ là:

$$P(S|E4) = \frac{0.0046}{0.0046 + 0.0092} \approx 0.334$$

$$P(N|E4) = \frac{0.0092}{0.0046 + 0.0092} \approx 0.666$$

Do đó ta phân loại E4 là Not Spam (N).

# Ví dụ: python code cho ví dụ trên

```
from sklearn.naive_bayes import MultinomialNB
import numpy as np
# train data
e1 = [1, 2, 1, 0, 1, 0, 0]
e2 = [0, 2, 0, 0, 1, 1, 1]
e3 = [1, 0, 1, 1, 0, 2, 0]
train_data = np.array([e1, e2, e3])
label = np.array(['N', 'N', 'S'])
# test data
e4 = np.array([[1, 0, 0, 0, 0, 0, 1]])
clf1 = MultinomialNB(alpha=1)
# training
clf1.fit(train_data, label)
# test
print('Probability of e4 in each class:', clf1.predict_proba(e4))
print('Predicting class of e4:', str(clf1.predict(e4)[0]))
```

```
Probability of e4 in each class: [[0.66589595 0.33410405]]
Predicting class of e4: N
```

# Kết luận

- **Naïve Bayes Classification** cho kết quả tốt trong thực tế mặc dù chịu những giả thiết về tính độc lập có điều kiện (khi được cho nhãn/lớp) của các thuộc tính
  - dễ cài đặt, thời gian train và test nhanh
  - sử dụng trong phân loại text, spam, etc
  - có thể hoạt động với các feature vector mà một phần là liên tục (sử dụng Gaussian Naive Bayes), phần còn lại ở dạng rời rạc (sử dụng Multinomial hoặc Bernoulli).
- Hạn chế:
  - giả định độc lập (ưu điểm cũng chính là nhược điểm) hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau
  - khi dữ liệu có nhiều thuộc tính dư thừa thì Naïve Bayes không còn hiệu quả
  - dữ liệu liên tục có thể không tuân theo phân phối chuẩn

# Tìm hiểu thêm

- Probabilistic Graphical Model (Mô hình xác suất dạng đồ thị)
- Bayesian network (BN) (Mạng Bayesian)

# Bài Tập

1) Toy example: Phân loại giới tính Male/Female dựa vào thông tin chiều cao và cân nặng. Cài đặt chương trình demo thuật toán Naïve Bayes

Tham khảo: <https://alphacoder.xyz/naive-bayes/>

height (ft)	weight (kg)	sex
6.3	50.2	Male
5.9	79.7	Female
5.1	61.4	Female
5.6	47.1	Male
5.1	59.8	Female

# Bài Tập

## 2) Nhận dạng ký tự dùng thuật toán Naïve Bayes

- The Database: UCI Letter-Recognition Data in a text file.
  - 26 classes: A to Z
  - 16-D feature vectors
  - 20,000 samples

<https://archive.ics.uci.edu/ml/datasets/letter+recognition>