

Naïve Bayes

Classification

ไม่มีความต่อเนื่องทางเวลา
หรือเป็นทศนิยมไม่ได้
แยก Class ได้ชิ้นๆ เชน
วัน,เดือน,ปี

Categorical Data
(Discrete Data)

Numerical Data
(Continuous Data)

มีความต่อเนื่องทางเวลา
หรือเป็นทศนิยมได้
เช่น ราคาหุ้น, อุณหภูมิ

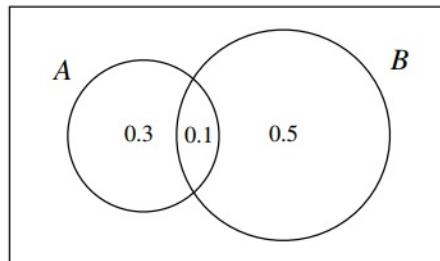
Bayes' Theorem

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

$$P(A) = 0.3$$

$$P(B) = 0.5$$

$$P(A \cap B) = 0.1 = P(A, B)$$



$$P(A | B) = \frac{P(A, B)}{P(B)} \quad \text{ความน่าจะเป็นของเหตุการณ์ A เมื่อเรารู้ B}$$

$$P(A | B) = \frac{P(A, B)}{P(B)} = \frac{0.1}{0.5} = \frac{1}{5}$$

$$P(B | A) = \frac{P(B, A)}{P(A)} = \frac{0.1}{0.3} = \frac{1}{3}$$

ความน่าจะเป็นของเหตุการณ์ B เมื่อเรารู้ A

ข้อสังเกต

$$P(A, B) = P(B, A)$$

Proof

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

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

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

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

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

$$\swarrow \qquad \searrow$$

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

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

Naïve Bayes

ใช้หาความน่าจะเป็นสูงสุด โดยดูจากผลคุณของความน่าจะเป็น^{พิสูจน์แล้วนี่กัน}
(ผนรวมที่ได้ไม่จำเป็นต้องเท่ากับ 1 เสมอไป)

$$C = \max_{k \in K} \left(P(C_k | F_1, F_2, \dots, F_D) \right) = \max_{k \in K} \left(P(C_k) \prod_{d=1}^D P(F_d | C_k) \right)$$

ความน่าจะเป็น Naive Bayes โดยที่ทุก Feature อิสระต่อกัน

Feature อิสระต่อกัน เพื่อกันโอกาสความน่าจะเป็นศูนย์
ที่เกิดมาจากการ Feature ใน Test set ไม่มีอยู่ใน Traing set

ต้องการพิสูจน์ว่า

ค่าสูงสุดของ Prob

Feature จากข้อมูลที่มี

$$C = \max_{k \in K} \left(P(C_k | F_1, F_2, \dots, F_D) \right) = \max_{k \in K} \left(P(C_k) \prod_{d=1}^D P(F_d | C_k) \right)$$

Ck คือ Class

จาก Bayes' Theorem

เรารู้ว่า

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

$$\begin{aligned} \max_{k \in K} \left(P(C_k | F_1, F_2, \dots, F_D) \right) &= \max_{k \in K} \left(\frac{P(F_1, F_2, \dots, F_D | C_k)P(C_k)}{P(F_1, F_2, \dots, F_D)} \right) \\ &= \max_{k \in K} \left(\frac{P(C_k)P(F_1, F_2, \dots, F_D | C_k)}{P(F_1, F_2, \dots, F_D)} \right) \end{aligned}$$

เมื่อจากเราพิจารณาค่า $\max_{k \in K}$ และ $P(F_1, F_2, \dots, F_D)$ เป็นค่าคงที่
ดังนั้น

เมื่อพิจารณาค่าสูงสุด
เราสามารถตัวตัวส่วนออกໄປได้
 เพราะส่วนมีค่าเท่ากันหมด เช่น
(1/8, 4/8, 10/8) จะได้ (1, 4, 10)

$$\max_{k \in K} \left(P(C_k | F_1, F_2, \dots, F_D) \right) = \max_{k \in K} \left(P(C_k)P(F_1, F_2, \dots, F_D | C_k) \right)$$

$$= P(C_k)P(F_1, F_2, \dots, F_D | C_k)$$

พิจารณาตัวนี้กัน

$$P(C_k)P(F_1, F_2, \dots, F_D | C_k)$$

จาก *Product Rule*

เรารู้ว่า $P(A, B) = P(A | B)P(B)$

ซึ่งสามารถแสดงในรูปของ *Bayes* ได้ดังนี้

$$P(A, B | Z) = P(A | B, Z)P(B | Z)$$

ตั้งขึ้น

$$\begin{array}{lcl} A & B & Z \\ P(F_1, F_2, \dots, F_D | C_k) & = & P(F_1 | F_2, \dots, F_D, C_k)P(F_2, \dots, F_D | C_k) \end{array}$$

**ไม่ใช่เรื่อยๆ
(Chain rule)**

$$\begin{array}{lcl} P(F_2, \dots, F_D | C_k) & = & P(F_2 | F_3, \dots, F_D, C_k)P(F_3, \dots, F_D | C_k) \\ P(F_3, \dots, F_D | C_k) & = & P(F_3 | F_4, \dots, F_D, C_k)P(F_4, \dots, F_D | C_k) \\ \vdots & & \\ P(F_{D-1}, F_D | C_k) & = & P(F_{D-1} | F_D, C_k)P(F_D | C_k) \end{array}$$

ผนั่นคือ

$$P(F_1, F_2, \dots, F_D | C_k) = P(F_1 | F_2, \dots, F_D, C_k) P(F_2 | F_3, \dots, F_D, C_k) P(F_3 | F_4, \dots, F_D, C_k) P(F_{D-1} | F_D, C_k) P(F_D | C_k)$$

จาก *Conditional Independence*

เรารู้ว่า

$$\left. \begin{array}{l} P(X | Y) = P(X) \\ P(X | Y, Z) = P(X | Y) \end{array} \right\} \quad X \text{ is independent from } Y$$

We have the assumption that F_1, F_2, \dots, F_D are all independent.

เรามีสมมติฐานว่า F_1, F_2, \dots, F_D เป็นอิสระต่อกัน

$$P(F_1 | F_2, \dots, F_D, C_k) = P(F_1 | C_k)$$

$$P(F_2 | F_3, \dots, F_D, C_k) = P(F_2 | C_k)$$

$$P(F_3 | F_4, \dots, F_D, C_k) = P(F_3 | C_k)$$

\vdots

$$P(F_{D-1} | F_D, C_k) = P(F_{D-1} | C_k)$$

$$P(F_D | C_k) = P(F_D | C_k)$$

$$P(F_1, F_2, \dots, F_D | C_k) = P(F_1 | C_k)P(F_2 | C_k)\dots P(F_D | C_k)$$

$$= \prod_{d=1}^D P(F_d | C_k)$$

คืนค่า C_k
ในตอนต้น

$$P(C_k)P(F_1, F_2, \dots, F_D | C_k) = P(C_k)P(F_1 | C_k)P(F_2 | C_k)\dots P(F_D | C_k)$$

$$= P(C_k) \prod_{d=1}^D P(F_d | C_k)$$

$$P(C_k) \prod_{d=1}^D P(F_d | C_k)$$

Data =

X1	X2	...	XD	Y
x_1^1	x_1^2	...	x_1^D	y_1
x_2^1	x_2^2	...	x_2^D	y_2
x_3^1	x_3^2	...	x_3^D	y_3
⋮	⋮	⋮	⋮	⋮
x_N^1	x_N^2	...	x_N^D	y_N

X คือ ตัวแปรต้น (**Feature**)

Y คือ ตัวแปรตาม (**Target**)

N คือ จำนวนตัวอย่างทั้งหมด

D คือ จำนวน **Feature** ทั้งหมด

State Data (Discrete Data)

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^D \\ x_2^1 & x_2^2 & \cdots & x_2^D \\ x_3^1 & x_3^2 & \cdots & x_3^D \\ \vdots & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^D \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}$$

x_3^D คือ Feature ที่ D ของ Example ที่ 3

y_2 คือ Target ของ Example ที่ 2

Ex.

Data =

การท่องงาน	การศึกษา	ฐานะ
ขยัน	เรียน	ด้วย
ขยัน	ไม่เรียน	ด้วย
ไม่เกียจ	เรียน	ด้วย
ไม่เกียจ	ไม่เรียน	จน
ขยัน	ไม่เรียน	จน
ไม่เกียจ	ไม่เรียน	จน

$$X = \begin{bmatrix} \text{ขยัน} & \text{เรียน} \\ \text{ขยัน} & \text{ไม่เรียน} \\ \text{ไม่เกียจ} & \text{เรียน} \\ \text{ไม่เกียจ} & \text{ไม่เรียน} \\ \text{ขยัน} & \text{ไม่เรียน} \\ \text{ไม่เกียจ} & \text{ไม่เรียน} \end{bmatrix}, Y = \begin{bmatrix} \text{ด้วย} \\ \text{ด้วย} \\ \text{จน} \\ \text{จน} \\ \text{จน} \\ \text{จน} \end{bmatrix}$$

มี 2 Class
จน, ด้วย

$$P(C_k)P(F_1, F_2, \dots, F_D | C_k) = P(C_k) \prod_{d=1}^D P(F_d | C_k)$$

$$C = \max_{k \in K} \left(P(C_k | F_1, F_2, \dots, F_D) \right) = \max_{k \in K} \left(P(C_k) \prod_{d=1}^D P(F_d | C_k) \right)$$

$$P(\text{ฐานะ} = \text{จน} | \text{การทำงาน} = \text{ขยัน}, \text{การศึกษา} = \text{เรียน}) = P(\text{จน} | \text{ขยัน}, \text{เรียน})$$

$$P(\text{ฐานะ} = \text{ด้วย} | \text{การทำงาน} = \text{ขยัน}, \text{การศึกษา} = \text{เรียน}) = P(\text{ด้วย} | \text{ขยัน}, \text{เรียน})$$

$$P(\text{จน} | \text{ขยัน}, \text{เรียน}) = P(\text{จน}) P(\text{ขยัน} | \text{จน}) P(\text{เรียน} | \text{จน})$$

$$P(\text{ด้วย} | \text{ขยัน}, \text{เรียน}) = P(\text{ด้วย}) P(\text{ขยัน} | \text{ด้วย}) P(\text{เรียน} | \text{ด้วย})$$

$$P(\text{จน} | \text{ขยัน}, \text{เรียน}) = P(\text{จน}) P(\text{ขยัน} | \text{จน}) P(\text{เรียน} | \text{จน})$$

$$P(\text{ราย} | \text{ขยัน}, \text{เรียน}) = P(\text{ราย}) P(\text{ขยัน} | \text{ราย}) P(\text{เรียน} | \text{ราย})$$

การทำงาน	การศึกษา	ฐานะ
ขยัน	เรียน	ราย
ขยัน	ไม่เรียน	ราย
ชี้เกี้ยจ	เรียน	ราย
ชี้เกี้ยจ	ไม่เรียน	จน
ขยัน	ไม่เรียน	จน
ชี้เกี้ยจ	ไม่เรียน	จน

$$P(\text{ราย}) = \frac{3}{6}$$

$$P(\text{จน}) = \frac{3}{6}$$

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

$$P(\text{ขยัน} | \text{ราย}) = \frac{2}{3}$$

$$P(\text{ชี้เกี้ยจ} | \text{ราย}) = \frac{1}{3}$$

$$P(\text{ขยัน} | \text{จน}) = \frac{1}{3}$$

$$P(\text{ชี้เกี้ยจ} | \text{จน}) = \frac{2}{3}$$

$$P(\text{เรียน} | \text{ราย}) = \frac{2}{3}$$

$$P(\text{ไม่เรียน} | \text{ราย}) = \frac{1}{3}$$

$$P(\text{เรียน} | \text{จน}) = 0$$

$$P(\text{ไม่เรียน} | \text{จน}) = 1$$

$$\begin{aligned} P(\text{จน} | \text{ขยัน}, \text{เรียน}) &= P(\text{จน}) P(\text{ขยัน} | \text{จน}) P(\text{เรียน} | \text{จน}) \\ &= \frac{3}{6} \times \frac{1}{3} \times 0 = \boxed{0} \end{aligned}$$

$$\begin{aligned} P(\text{ราย} | \text{ขยัน}, \text{เรียน}) &= P(\text{ราย}) P(\text{ขยัน} | \text{ราย}) P(\text{เรียน} | \text{ราย}) \\ &= \frac{3}{6} \times \frac{2}{3} \times \frac{2}{3} = \boxed{\frac{4}{18}} \end{aligned}$$

ท่านนาย

การทำงาน	การศึกษา	ฐานะ
ขยัน	เรียน	?

$$\max(P(\text{จน} | \text{ขยัน}, \text{เรียน}), P(\text{ราย} | \text{ขยัน}, \text{เรียน}))$$

$$\text{MAX}(0, 4/18) \Rightarrow \text{ราย}$$

Code Naive Bayes สำหรับ Discrete Data

- I. สร้าง **Dictionary** สำหรับหัวข้อที่มีใน Training Set และ เก็บความน่าจะเป็น
- II. นับจำนวนที่มีใน Training Set เพื่อใส่ใน **Count_Matrix**
- III. คำนวณความน่าจะเป็นจาก **Count_Matrix** เพื่อใส่ใน **Prob_Matrix**
- IV. เรียนรู้
- V. พยากรณ์

การทำงาน	การศึกษา	ฐานะ
ขยัน	เรียน	ราย
ขยัน	ไม่เรียน	ราย
ชี้เกียจ	เรียน	ราย
ชี้เกียจ	ไม่เรียน	จน
ขยัน	ไม่เรียน	จน
ชี้เกียจ	ไม่เรียน	จน

Feature_Name = [การทำงาน, การศึกษา],

Feature_Value = [[ขยัน, ชี้เกียจ], [เรียน, ไม่เรียน]],

All_Class = [จน, ราย]

```
Feature_Name = np.array(Data.columns[:-1])
All_Class, Count_All_Class = np.unique(Y_Train[:, :], return_counts = True)
print(All_Class)
print(Count_All_Class)
```

I. สร้าง Dictionary สำหรับนับจำนวนสมาชิก และ เก็บความน่าจะเป็น

```
def NB_create_DictStorage(Feature_Name, Feature_Value, All_Class):
    DictStorage = {}
    n_class = len(All_Class)
    n_columns = n_class + 1
    D = len(Feature_Name)
    for d in range(D):
        fn = Feature_Name[d]
        n_value = len(Feature_Value[d])
        n_rows = n_value + 1
        MatrixStorage = np.zeros([n_rows, n_columns], dtype='object')
        for c in range(n_class):
            MatrixStorage[0, c+1] = All_Class[c]
        for r in range(n_value):
            MatrixStorage[r+1, 0] = Feature_Value[d][r]
        DictStorage[fn] = MatrixStorage
    return DictStorage
```

MatrixStorage = $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

```
DictStorage = {}
### All_Class = [จน, ราย]
n_class = 2 = len(All_Class) = len([จน, ราย])
n_columns = 3 = 2 + 1 = n_class + 1
### Feature_Name = [การทำงาน, การศึกษา]
D = 2 = len(Feature_Name) = len([การทำงาน, การศึกษา])
```

$$\text{MatrixStorage} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{matrix} C=0 \\ C=1 \end{matrix}$$

$$\begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0 & 0 \\ \text{ชีเกียจ} & 0 & 0 \end{bmatrix} \quad \begin{matrix} r=0 \\ r=1 \end{matrix}$$

$$DictStorage = \{\text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0 & 0 \\ \text{ชีเกียจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}\}$$

d = 0 d = 1

$$X_{Train} = \begin{bmatrix} \text{ขยัน} & \text{เรียน} \\ \text{ขยัน} & \text{ไม่เรียน} \\ \text{ชี้เกียจ} & \text{เรียน} \\ \text{ชี้เกียจ} & \text{เรียน} \\ \text{ขยัน} & \text{ไม่เรียน} \\ \text{ชี้เกียจ} & \text{ไม่เรียน} \end{bmatrix} \quad Feature_Name = [\text{การทำงาน}, \text{การศึกษา}]$$

```
def NB_find_Feature_Value(X_Train, Feature_Name):
    Feature_Value = []
    D = len(Feature_Name)
    for d in range(D):
        Value = np.unique(X_Train[:, d])
        Feature_Value.append(Value)
    return Feature_Value
```

d=0 $\text{np.unique}(\begin{bmatrix} \text{ขยัน} \\ \text{ขยัน} \\ \text{ชี้เกียจ} \\ \text{ชี้เกียจ} \\ \text{ขยัน} \\ \text{ชี้เกียจ} \end{bmatrix})$ = [[**ขยัน, ชี้เกียจ**]]

d=1 $\text{np.unique}(\begin{bmatrix} \text{เรียน} \\ \text{ไม่เรียน} \\ \text{เรียน} \\ \text{เรียน} \\ \text{ไม่เรียน} \\ \text{ไม่เรียน} \end{bmatrix})$

Feature_Value = [[**ขยัน, ชี้เกียจ**], [**เรียน, ไม่เรียน**]]

II. พัฒนาสมการจาก Training Set เพื่อใส่ใน Count_Matrix

$$X_Train = \begin{bmatrix} \text{ชั้น} & \text{เรียน} \\ \text{ชั้น} & \text{ไม่เรียน} \\ \text{ชีวิจ} & \text{เรียน} \\ \text{ชีวิจ} & \text{ไม่เรียน} \\ \text{ชั้น} & \text{ไม่เรียน} \\ \text{ชีวิจ} & \text{ไม่เรียน} \end{bmatrix}, \quad Y_Train = \begin{bmatrix} \text{ราย} \\ \text{ราย} \\ \text{ราย} \\ \text{จน} \\ \text{จน} \\ \text{จน} \end{bmatrix}$$

$D = 2 = \text{len(Feature_Name)} = \text{len}([\text{การทำงาน}, \text{การศึกษา}])$

$n_rows = 3$
 $n_columns = 3$

```
def NB_count_element(X_Train, Y_Train, Feature_Name, empty_Count):
    D = len(Feature_Name)
    for d in range(D):
        fn = Feature_Name[d]
        n_rows, n_columns = empty_Count[fn].shape
        for c in range(1, n_columns):
            filter_class = (Y_Train[:, 0] == empty_Count[fn][0, c])
            for r in range(1, n_rows):
                filter_feature = (X_Train[:, d] == empty_Count[fn][r, 0])
                empty_Count[fn][r, c] = (filter_feature & filter_class).sum()
    return empty_Count
```

$Feature_Name = [\text{การทำงาน}, \text{การศึกษา}]$

$$\text{empty_Count} = \{\text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชั้น} & 0 & 0 \\ \text{ชีวิจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}\}$$

$d = 0$

$$\begin{aligned} \text{filter_feature} = \begin{bmatrix} \text{True} \\ \text{True} \\ \text{False} \\ \text{False} \\ \text{True} \\ \text{False} \end{bmatrix} &= (X_Train[:, 0] == \text{ชั้น}) \\ &= (X_Train[:, d] == empty_Count[fn][r, 0]) \\ &= \begin{bmatrix} \text{ชั้น} \\ \text{ชั้น} \\ \text{ชีวิจ} \\ \text{ชีวิจ} \\ \text{ชั้น} \\ \text{ชีวิจ} \end{bmatrix} == \text{ชั้น} \end{aligned}$$

$\text{empty_Count} = \{\text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชั้น} & 0 & 0 \\ \text{ชีวิจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}\}$

$\text{filter_class} = \begin{bmatrix} \text{False} \\ \text{False} \\ \text{False} \\ \text{True} \\ \text{True} \\ \text{True} \end{bmatrix} = (Y_Train[:, 0] == empty_Count[fn][0, c]) = (Y_Train[:, 0] == empty_Count[fn][0, 0]) = \begin{bmatrix} \text{ราย} \\ \text{ราย} \\ \text{ราย} \\ \text{จน} \\ \text{จน} \\ \text{จน} \end{bmatrix} == \text{จน}$

$\text{empty_Count}[fn][r, c] = 1 = (\begin{bmatrix} \text{True} \\ \text{True} \\ \text{False} \\ \text{False} \\ \text{False} \\ \text{True} \end{bmatrix} \& \begin{bmatrix} \text{False} \\ \text{False} \\ \text{False} \\ \text{True} \\ \text{True} \\ \text{True} \end{bmatrix}) = \begin{bmatrix} \text{False} \\ \text{False} \\ \text{False} \\ \text{True} \\ \text{True} \\ \text{False} \end{bmatrix} . \text{sum}()$

$\text{empty_Count} = \{\text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชั้น} & 1 & 0 \\ \text{ชีวิจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}\}$

$$empty_Count = \{ \text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ขยัน} & 1 & 0 \\ \text{ชีกิจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix} \}$$

Return

$$empty_Count = \{ \text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ขยัน} & 1 & 2 \\ \text{ชีกิจ} & 2 & 1 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 2 \\ \text{ไม่เรียน} & 3 & 1 \end{bmatrix} \}$$



เปลี่ยนชื่อตัวแปรเปรี้ยว
จะได้ไม่ซ้ำกัน

$$Count_Matrix = \{ \text{การทำงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ขยัน} & 1 & 2 \\ \text{ชีกิจ} & 2 & 1 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 2 \\ \text{ไม่เรียน} & 3 & 1 \end{bmatrix} \}$$

III. คำนวณความน่าจะเป็นจาก Count_Matrix เพื่อใส่ใน Prob_Matrix

```
def NB_calculate_prob(empty_Prob, Count_Matrix, Count_All_Class, Feature_Name):
    D = len(Feature_Name)
    for d in range(D):
        fn = Feature_Name[d]
        empty_Prob[fn][1:, 1:] = Count_Matrix[fn][1:, 1:]/Count_All_Class
    return empty_Prob
```

empty_Prob = {การทำงาน : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{ขยัน} & 0 & 0 \\ \text{ซึ้งกิจ} & 0 & 0 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}$ }

Count_Matrix = {การทำงาน : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{ขยัน} & 1 & 2 \\ \text{ซึ้งกิจ} & 2 & 1 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{เรียน} & 0 & 2 \\ \text{ไม่เรียน} & 3 & 1 \end{bmatrix}$ }

Count_All_Class = [3,3]

$$\text{empty_Prob}[fn][1:, 1:] = \frac{\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 3 \end{bmatrix}}{[3 \ 3]} = \text{Count_Matrix}[fn][1:, 1:]/\text{Count_All_Class}$$

empty_Prob = {การทำงาน : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{ขยัน} & 0.33 & 0.66 \\ \text{ซึ้งกิจ} & 0.66 & 0.33 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}$ }

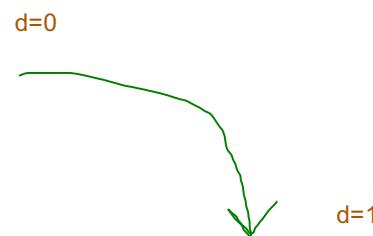
empty_Prob = {การทำงาน : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{ขยัน} & 0 & 0 \\ \text{ซึ้งกิจ} & 0 & 0 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}$ }

Count_Matrix = {การทำงาน : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{ขยัน} & 1 & 2 \\ \text{ซึ้งกิจ} & 2 & 1 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{เรียน} & 0 & 2 \\ \text{ไม่เรียน} & 3 & 1 \end{bmatrix}$ }

Count_All_Class = [3,3]

Feature_Name = [การทำงาน, การศึกษา]

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



$$\text{empty_Prob}[fn][1:, 1:] = \frac{\begin{bmatrix} 0 & 2 \\ 3 & 1 \\ 3 & 3 \end{bmatrix}}{[3 \ 3]} = \text{Count_Matrix}[fn][1:, 1:]/\text{Count_All_Class}$$

empty_Prob = {การทำงาน : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{ขยัน} & 0.33 & 0.66 \\ \text{ซึ้งกิจ} & 0.66 & 0.33 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{รวย} \\ \text{เรียน} & 0 & 0.66 \\ \text{ไม่เรียน} & 1 & 0.33 \end{bmatrix}$ }

IV. เรียนรู้

$$X_Train = \begin{bmatrix} \text{ชยัน} & \text{เรียน} \\ \text{ชยัน} & \text{ไม่เรียน} \\ \text{ชีกิจ} & \text{เรียน} \\ \text{ชีกิจ} & \text{ไม่เรียน} \\ \text{ชยัน} & \text{ไม่เรียน} \\ \text{ชีกิจ} & \text{ไม่เรียน} \end{bmatrix}, Y_Train = \begin{bmatrix} \text{ราย} \\ \text{ราย} \\ \text{ราย} \\ \text{จน} \\ \text{จน} \\ \text{จน} \end{bmatrix}$$

$Feature_Name = [\text{การทำางาน}, \text{การศึกษา}]$
 $Feature_Value = [[\text{ชยัน}, \text{ชีกิจ}], [\text{เรียน}, \text{ไม่เรียน}]]$,
 $All_Class = [\text{จน}, \text{ราย}]$
 $Count_All_Class = [3,3]$

Return

$$Count_Matrix = \{\text{การทำางาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 1 & 2 \\ \text{ชีกิจ} & 2 & 1 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 2 \\ \text{ไม่เรียน} & 3 & 1 \end{bmatrix}\}$$

$$Prob_Matrix = \{\text{การทำางาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0.33 & 0.66 \\ \text{ชีกิจ} & 0.66 & 0.33 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0.66 \\ \text{ไม่เรียน} & 1 & 0.33 \end{bmatrix}\}$$

```

def NBDD_fit(X_Train, Y_Train, Feature_Name, Feature_Value, All_Class, Count_All_Class):
    (1) empty_Count = NB_create_DictStorage(Feature_Name, Feature_Value, All_Class)
    (2) Count_Matrix = NB_count_element(X_Train, Y_Train, Feature_Name, empty_Count)
    (1) empty_Prob = NB_create_DictStorage(Feature_Name, Feature_Value, All_Class)      สร้าง dic เก็บผลลัพธ์ค่า Prob
    (3) Prob_Matrix = NB_calculate_prob(empty_Prob, Count_Matrix, Count_All_Class, Feature_Name)
    return Count_Matrix, Prob_Matrix

```

$$empty_Count = \{\text{การทำางาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0 & 0 \\ \text{ชีกิจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}\}$$

$$Count_Matrix = \{\text{การทำางาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 1 & 2 \\ \text{ชีกิจ} & 2 & 1 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 2 \\ \text{ไม่เรียน} & 3 & 1 \end{bmatrix}\}$$

$$empty_Prob = \{\text{การทำางาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0 & 0 \\ \text{ชีกิจ} & 0 & 0 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0 \\ \text{ไม่เรียน} & 0 & 0 \end{bmatrix}\}$$

$$Prob_Matrix = \{\text{การทำางาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชยัน} & 0.33 & 0.66 \\ \text{ชีกิจ} & 0.66 & 0.33 \end{bmatrix}, \text{การศึกษา} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0.66 \\ \text{ไม่เรียน} & 1 & 0.33 \end{bmatrix}\}$$

V. พยากรณ์

$Prob_Matrix = \text{การท่องงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชั้น} & 0.33 & 0.66 \\ \text{นักเรียน} & 0.66 & 0.33 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0.66 \\ \text{ไม่เรียน} & 1 & 0.33 \end{bmatrix}$

$Feature_Name = [\text{การท่องงาน}, \text{การศึกษา}],$

$Feature_Value = [[\text{ชั้น}, \text{นักเรียน}], [\text{เรียน}, \text{ไม่เรียน}]]$,

$All_Class = [\text{จน}, \text{ราย}]$

$X_Test = [\text{ชั้น เรียน}]$

$Count_All_Class = [3,3]$

```
def NBDD_predict(X_Test, Prob_Matrix, Feature_Name, Feature_Value, All_Class, Count_All_Class, top = 1):
    prob_class = Count_All_Class/(Count_All_Class.sum())
    Yhat = []
    D = len(Feature_Name)
    C = len(All_Class)
    for x_test in X_Test:
        prob_fgc = np.zeros([D, C])
        for d in range(D):
            fn = Feature_Name[d]
            r = np.argwhere(Feature_Value[d] == x_test[d]).ravel()[0]
            prob_fgc[d, :] = Prob_Matrix[fn][r+1, 1:]
        prob_afgc = prob_fgc.prod(axis=0)
        prob_nb = prob_class*prob_afgc
        sorted_class = All_Class[prob_nb.argsort()[:-1]]
        Yhat.append(sorted_class)
    return np.array(Yhat)[:, :top]
```

$$\text{prob_class} = [0.5 \quad 0.5] = \left[\frac{3}{6} \quad \frac{3}{6} \right] = \frac{[3 \quad 3]}{6} = \text{Count_All_Class}/(\text{Count_All_Class.sum()})$$

$D = 2 = \text{len}(\text{Feature_Name}) = \text{len}([\text{การท่องงาน}, \text{การศึกษา}])$

$C = 2 = \text{len}(\text{All_Class}) = \text{len}([\text{จน}, \text{ราย}])$

$$r = 0 = \text{np.argwhere}([\text{ชั้น}, \text{นักเรียน}] == [\text{ชั้น}]).ravel()[0] \quad 0, 1 \quad d = 0$$

$$r = 0 = \text{np.argwhere}([\text{เรียน}, \text{ไม่เรียน}] == [\text{เรียน}]).ravel()[0] \quad 0, 1 \quad d = 1$$

$Prob_Matrix = \text{การท่องงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชั้น} & 0.33 & 0.66 \\ \text{nักเรียน} & 0.66 & 0.33 \end{bmatrix}$, การศึกษา : $\begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{เรียน} & 0 & 0.66 \\ \text{ไม่เรียน} & 1 & 0.33 \end{bmatrix}$

d=0

$$\text{prob_fgc} = [0, 0] = \text{การท่องงาน} : \begin{bmatrix} 0 & \text{จน} & \text{ราย} \\ \text{ชั้น} & 0.33 & 0.66 \\ \text{nักเรียน} & 0.66 & 0.33 \end{bmatrix} = \begin{bmatrix} 0.33 & 0 \\ 0 & 0 \end{bmatrix}$$

$$d=1 \quad \text{prob_fgc} = \begin{bmatrix} 0.33 & 0.66 \\ 0 & 0.66 \end{bmatrix}$$

$$\text{prob_afgc} = [0 \quad 0.44] = \text{prob_fgc}.prod(\text{axis}=0) = [0.33 \times 0 \quad 0.66 \times 0.66]$$

$\text{prob_class} = [0.5 \quad 0.5]$

index 0 1

$$\text{prob_nb} = [0 \quad 0.44] = \text{prob_class} * \text{prob_afgc} = \text{prob_nb} = [0 \times 0.5 \quad 0.44 \times 0.5]$$

จน ราย

$\text{sorted_class} = [\text{จน} \text{ ราย}] = \text{All_Class}[\text{prob_nb.argsort()[:-1]}] = \text{All_Class}[[0, 1][:-1]]$



จน ราย

$\text{Yhat} = []$

$\text{Yhat.append}(\text{sorted_class})$

$\text{Yhat} = [[\text{จน}, \text{ราย}]]$

นักเรียนมากสุด

Numerical Data (Continuous Data)

มีความต่อเนื่องทางเวลา
หรือเป็นทศนิยมได้
 เช่น ราคาหุ้น, อุณหภูมิ

$$P(C_k | F_1, F_2, \dots, F_D) = \max_{k \in K} \left(P(C_k) \prod_{d=1}^D P(F_d | C_k) \right)$$

Ex.

Data =

ชั่วโมงเรียน	ชั่วโมงทบทวน	ผลสอบ
3.4	13.7	ผ่าน
2.1	10.8	ผ่าน
2.3	6.8	ผ่าน
7.3	0	ไม่ผ่าน
4.7	1	ไม่ผ่าน
1.5	4.8	ไม่ผ่าน

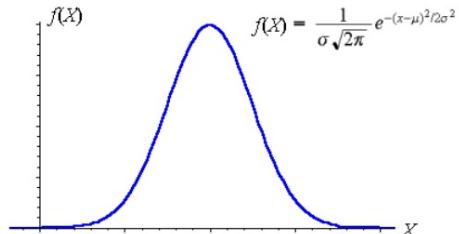
ใช้ Test (พยากรณ์)

ชั่วโมงเรียน	ชั่วโมงทบทวน	ผลสอบ
3.5	6	?

$$\max(P(\text{ผ่าน}|3.5, 6), P(\text{ไม่ผ่าน}|3.5, 6))$$

คำนวณความน่าจะเป็น

Normal Distribution



(pdf) ใช้วัดความหนาแน่น ของความน่าจะเป็น

ทฤษฎีของเกาส์ : ข้อมูลตัวอย่างที่มากพอ จะมีการกระจายตัว แบบ Normal เสมอ (โดยส่วนใหญ่) -> ถึงแม้ข้อมูลนั้นอาจไม่ใช่ Normal กราฟเบี้ยงซ้าย/ขวา) เราก็สามารถดึงค่าเฉลี่ย Mean ให้อยู่ตรงกลางได้ (Cox-Box)

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

Standard Deviation

$$\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{n-1}}$$

$$\bar{X} = \frac{\text{Sum}}{n} = \mu$$

76	84	69	92	58
89	73	97	85	77

Ex.

Data =

ชั่วโมงเรียน	ชั่วโมงทบทวน	ผลสอบ
3.4	13.7	ผ่าน
2.1	10.8	ผ่าน
2.3	6.8	ผ่าน
7.3	0	ไม่ผ่าน
4.7	1	ไม่ผ่าน
1.5	4.8	ไม่ผ่าน

$$\mu(\text{ชั่วโมงเรียน} | \text{ผ่าน}) = 2.6$$

$$\mu(\text{ชั่วโมงเรียน} | \text{ไม่ผ่าน}) = 4.5$$

หาค่าเฉลี่ย

$$\mu(\text{ชั่วโมงทบทวน} | \text{ผ่าน}) = 10.43$$

$$\mu(\text{ชั่วโมงทบทวน} | \text{ไม่ผ่าน}) = 1.93$$

$$\sigma(\text{ชั่วโมงเรียน} | \text{ผ่าน}) = 0.57$$

$$\sigma(\text{ชั่วโมงเรียน} | \text{ไม่ผ่าน}) = 2.37$$

$$\sigma(\text{ชั่วโมงทบทวน} | \text{ผ่าน}) = 2.83$$

$$\sigma(\text{ชั่วโมงทบทวน} | \text{ไม่ผ่าน}) = 2.07$$

$$P(\text{ผลสอบ} = \text{ผ่าน} \mid \text{ชั่วโมงเรียน} = 3.5, \text{ชั่วโมงทบทวน} = 6) = P(\text{ผ่าน}) P(3.5 | \text{ผ่าน}) P(6 | \text{ผ่าน})$$

$$P(3.5 | \text{ผ่าน}) = \frac{1}{0.57\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{3.5-2.6}{0.57}\right)^2} = 0.20 \quad \text{ชั่วโมงเรียน}$$

$$P(\text{ผ่าน}) = \frac{3}{6}$$

$$P(6 | \text{ผ่าน}) = \frac{1}{2.83\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{6-10.43}{2.83}\right)^2} = 0.04 \quad \text{ชั่วทบทวน}$$

$$P(\text{ผลสอบ} = \text{ไม่ผ่าน} \mid \text{ชั่วโมงเรียน} = 3.5, \text{ชั่วโมงทบทวน} = 6) = P(\text{ไม่ผ่าน}) P(3.5 | \text{ไม่ผ่าน}) P(6 | \text{ไม่ผ่าน})$$

$$P(3.5 | \text{ไม่ผ่าน}) = \frac{1}{2.57\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{3.5-4.5}{2.37}\right)^2} = 0.15 \quad \text{ชั่วโมงเรียน}$$

$$P(\text{ไม่ผ่าน}) = \frac{3}{6}$$

$$P(6 | \text{ไม่ผ่าน}) = \frac{1}{2.07\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{6-1.93}{2.07}\right)^2} = 0.03 \quad \text{ชั่วทบทวน}$$

ชั่วโมงเรียน	ชั่วโมงทบทวน	ผลสอบ
3.5	6	?

$$\max(P(\text{ผ่าน} | 3.5, 6), P(\text{ไม่ผ่าน} | 3.5, 6))$$

$$P(\text{ผลสอบ = ผ่าน} \mid \text{ชั่วโมงเรียน} = 3.5, \text{ชั่วโมงทบทวน} = 6)$$

$$\begin{aligned} P(\text{ผ่าน} | 3.5, 6) &= P(\text{ผ่าน}) P(3.5 | \text{ผ่าน}) P(6 | \text{ผ่าน}) \\ &= \frac{3}{6} \times 0.2 \times 0.04 = \boxed{0.004} \end{aligned}$$

ผ่าน ไม่ผ่าน

$$\max(0.004, 0.002) = 0.004 \Rightarrow \underline{\text{ผ่าน}}$$

$$P(\text{ผลสอบ = ไม่ผ่าน} \mid \text{ชั่วโมงเรียน} = 3.5, \text{ชั่วโมงทบทวน} = 6)$$

$$\begin{aligned} P(\text{ไม่ผ่าน} | 3.5, 6) &= P(\text{ไม่ผ่าน}) P(3.5 | \text{ไม่ผ่าน}) P(6 | \text{ไม่ผ่าน}) \\ &= \frac{3}{6} \times 0.15 \times 0.03 = \boxed{0.002} \end{aligned}$$

Code Naive Bayes สำหรับ Continuous Data

- I. สร้าง Matrix สำหรับเก็บค่าเฉลี่ย และ ส่วนเบี่ยงเบนมาตรฐาน
- II. เรียนรู้
- III. พยากรณ์

ชั่วโมงเรียน	ชั่วโมงทบทวน	ผลสอบ
3.4	13.7	ผ่าน
2.1	10.8	ผ่าน
2.3	6.8	ผ่าน
7.3	0	ไม่ผ่าน
4.7	1	ไม่ผ่าน
1.5	4.8	ไม่ผ่าน

Training Set

ชั่วโมงเรียน	ชั่วโมงทบทวน	ผลสอบ
3.5	6	?

Test Set

Feature_Name = [ชั่วโมงเรียน, ชั่วโมงทบทวน]

All_Class = [ผ่าน, ไม่ผ่าน]

Count_All_Class = [3, 3]

I. สร้าง Matrix สำหรับเก็บค่าเฉลี่ย และ ส่วนเบี่ยงเบนมาตรฐาน

Feature_Name = [ชั่วโมงเรียน, ชั่วโมงทบทวน]

All_Class = [ผ่าน, ไม่ผ่าน]

```
def NB_create_MeanStd_Storage(Feature_Name, All_Class):
    n_feature = len(Feature_Name)
    n_rows = n_feature + 1
    n_class = len(All_Class)
    n_columns = n_class + 1
    mean = np.zeros([n_rows, n_columns])
    std = np.zeros([n_rows, n_columns])
    for c in range(n_class):
        mean[0, c+1] = All_Class[c]
        std[0, c+1] = All_Class[c]
    for r in range(n_feature):
        mean[r+1, 0] = Feature_Name[r]
        std[r+1, 0] = Feature_Name[r]
    return mean, std
```

$$mean = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$std = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & ผ่าน & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & ผ่าน & ไม่ผ่าน \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & ผ่าน & ไม่ผ่าน \\ ชั่วโมงเรียน & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & ผ่าน & ไม่ผ่าน \\ ชั่วโมงเรียน & 0 & 0 \\ ชั่วโมงทบทวน & 0 & 0 \end{bmatrix}$$

Return

$$mean = \begin{bmatrix} 0 & ผ่าน & ไม่ผ่าน \\ ชั่วโมงเรียน & 0 & 0 \\ ชั่วโมงทบทวน & 0 & 0 \end{bmatrix},$$

$$std = \begin{bmatrix} 0 & ผ่าน & ไม่ผ่าน \\ ชั่วโมงเรียน & 0 & 0 \\ ชั่วโมงทบทวน & 0 & 0 \end{bmatrix}$$

II. เรียนรู้

$$X_{Train} = \begin{bmatrix} 3.4 & 13.7 \\ 2.1 & 10.8 \\ 2.3 & 6.8 \\ 7.3 & 0 \\ 4.7 & 1 \\ 1.5 & 4.8 \end{bmatrix}, \quad Y_{Train} = \begin{bmatrix} \text{ผ่าน} \\ \text{ผ่าน} \\ \text{ผ่าน} \\ \text{ไม่ผ่าน} \\ \text{ไม่ผ่าน} \\ \text{ไม่ผ่าน} \end{bmatrix}$$

Feature_Name = [ชั้วโมงเรียน, ชั้วโมงทบทวน]

All_Class = [ผ่าน, ไม่ผ่าน]

```
def NBCD_fit(X_Train, Y_Train, Feature_Name, All_Class):
    mean, std = NB_create_MeanStd_Storage(Feature_Name, All_Class)
    n_feature = len(Feature_Name)
    n_class = len(All_Class)
    for c, _class in enumerate(All_Class):
        filter_class = np.argwhere(Y_Train[:, 0] == _class).ravel()
        X_Class = X_Train[filter_class]
        for r in range(n_feature):
            mean[r+1, c+1] = X_Class[:, r].mean()
            std[r+1, c+1] = X_Class[:, r].std()
    return mean, std
```

filter_class = np.argwhere($\begin{bmatrix} \text{ผ่าน} \\ \text{ผ่าน} \\ \text{ผ่าน} \\ \text{ไม่ผ่าน} \\ \text{ไม่ผ่าน} \\ \text{ไม่ผ่าน} \end{bmatrix} == \text{ผ่าน})$.ravel() = [0,1,2]

$X_{Train} = \begin{bmatrix} 3.4 & 13.7 \\ 2.1 & 10.8 \\ 2.3 & 6.8 \\ 7.3 & 0 \\ 4.7 & 1 \\ 1.5 & 4.8 \end{bmatrix}$

$X_{Class} = \begin{bmatrix} 3.4 & 13.7 \\ 2.1 & 10.8 \\ 2.3 & 6.8 \end{bmatrix}$

$mean[r+1, c+1] = 2.6 = X_{Class}[:, r].mean() = X_{Class}[:, 0].mean() = \begin{bmatrix} 3.4 \\ 2.1 \\ 2.3 \end{bmatrix}.mean()$

$std[r+1, c+1] = 0.57 = X_{Class}[:, r].std() = X_{Class}[:, 0].std() = \begin{bmatrix} 3.4 \\ 2.1 \\ 2.3 \end{bmatrix}.std()$

Return

$$mean = \begin{bmatrix} 0 & \text{ผ่าน} & \text{ไม่ผ่าน} \\ \text{ชั้วโมงเรียน} & 2.6 & 4.5 \\ \text{ชั้วโมงทบทวน} & 10.43 & 1.93 \end{bmatrix},$$

$$std = \begin{bmatrix} 0 & \text{ผ่าน} & \text{ไม่ผ่าน} \\ \text{ชั้วโมงเรียน} & 0.57 & 2.37 \\ \text{ชั้วโมงทบทวน} & 2.83 & 2.07 \end{bmatrix}$$

III. พยากรณ์

เรียน ทบทวน
 $X_{Test} = [3.5 \ 6]$

$All_Class = [\text{ผ่าน}, \text{ไม่ผ่าน}]$
 $Count_All_Class = [3, 3]$

$$mean = \begin{bmatrix} 0 & \text{ผ่าน} & \text{ไม่ผ่าน} \\ \frac{1}{3} \cdot 3 & 2.6 & 4.5 \\ \frac{1}{3} \cdot 3 & 10.43 & 1.93 \end{bmatrix}, \quad std = \begin{bmatrix} 0 & \text{ผ่าน} & \text{ไม่ผ่าน} \\ \sqrt{\frac{1}{3}} \cdot 3 & 0.57 & 2.37 \\ \sqrt{\frac{1}{3}} \cdot 3 & 2.83 & 2.07 \end{bmatrix}$$

```
def NB_CD_predict(X_Test, mean, std, All_Class, Count_All_Class, top = 1):
    prob_class = Count_All_Class / (Count_All_Class.sum())
    n_feature = X_Test.shape[1]
    n_class = len(All_Class)
    Yhat = []
    for x_test in X_Test:
        prob_fgc = np.zeros([n_feature, n_class])
        for d in range(n_feature):
            for c in range(n_class):
                prob_fgc[d, c] = normal_pdf(x_test[d], mean[d+1, c+1], std[d+1, c+1])
        prob_afgc = prob_fgc.prod(axis=0)
        prob_nb = prob_class * prob_afgc
        sorted_class = All_Class[prob_nb.argsort()[:-1]]
        Yhat.append(sorted_class)
    return np.array(Yhat)[:, :top]
```

$$\text{Count_All_Class} = [3, 3] \\ \text{prob_class} = [0.5 \ 0.5] = \begin{bmatrix} \frac{1}{3} & 1 \\ 1 & \frac{1}{3} \end{bmatrix} = \frac{[3 \ 3]}{6} = \text{Count_All_Class}/(\text{Count_All_Class.sum()}) \\ \text{Feature_Name} = [\text{ชั่วโมงเรียน}, \text{ชั่วโมงพักผ่อน}] \\ n_feature = 2 = \text{len}(\text{Feature_Name}) = \text{len}([\text{ชั่วโมงเรียน}, \text{ชั่วโมงพักผ่อน}]) \\ All_Class = [\text{ผ่าน}, \text{ไม่ผ่าน}] \\ n_class = 2 = \text{len}(All_Class) = \text{len}([\text{ผ่าน}, \text{ไม่ผ่าน}])$$

$$\text{prob_fgc}[d, c] = 0.202 \\ = \text{normal_pdf}(x_{\text{test}}[d], \text{mean}[d+1, c+1], \text{std}[d+1, c+1]) \\ = \text{normal_pdf}(x_{\text{test}}[0], \text{mean}[1, 1], \text{std}[1, 1]) \\ = \text{normal_pdf}(3.5, 2.6, 0.57)$$

$$x_{\text{test}}[d=0] = 3.5; \text{mean}[d=0] = 2.6; \text{std}[d=0] = 0.57 \\ prob_fgc = \begin{bmatrix} 0.202 & 0 \\ 0 & 0 \end{bmatrix} \quad prob_fgc = \begin{bmatrix} 0.202 & 0.154 \\ 0 & 0 \end{bmatrix}$$

$$prob_fgc = \begin{bmatrix} 0.202 & 0.154 \\ 0.041 & 0.028 \end{bmatrix} \quad \text{อุณหภูมิที่ร้อน}$$

$$prob_afgc = prob_fgc.prod(axis=0) = [0.202 \times 0.041 \ 0.154 \times 0.028] = [0.008 \ 0.004]$$

$$\text{prob_class} = [0.5 \ 0.5] \quad \text{ผ่าน} \ \text{ไม่ผ่าน}$$

$$prob_nb = prob_class * prob_afgc = [0.5 \times 0.008 \ 0.5 \times 0.004] = [0.004 \ 0.002]$$

Return

$$\text{All_Class} = [\text{ผ่าน}, \text{ไม่ผ่าน}]$$

$$\text{sorted_class} = All_Class[prob_nb.argsort()[:-1]] = All_Class[[1, 0][:-1]] = All[0, 1]$$

$$Yhat = []$$

$$Yhat.append(sorted_class)$$

$$Yhat = [[\text{ผ่าน}, \text{ไม่ผ่าน}]]$$

$$Yhat_Test = [\text{ผ่าน}]$$

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

```
def normal_pdf(x, mean, std):
    degree = ((x - mean)/std)**2
    fraction = np.e**(-degree/2)
    denominator = std*np.sqrt(2*np.pi)
    pdf = fraction/denominator
    return pdf
```

$\left(\frac{x-\mu}{\sigma}\right)^2$
 $e^{-\text{degree}/2}$
 \downarrow
 $\text{std } \sqrt{2\pi}$
 $\frac{e^{-\text{degree}}}{\text{std } \sqrt{2\pi}}$