

CRITERION AND MAX_LEAF_NODES

กลุ่ม กลุ่ม X หอยหลอดด



กลุ่ม กลุ่ม X หอยหลอดด

- 643020501-6 นายตะวัน เป้าหล่อเพชร
- 643021260-7 นางสาวกิตติลักษณ์ ลาดโภม
- 643021261-5 นางสาวจารุพร การร้อย
- 643021263-1 นางสาวชนเมษนก อั้งคุระเจ
- 643021265-7 นายธนาธิป อินทรคีรี
- 643021266-5 นางสาวธิตพร ใจเอื้อ
- 643021268-1 นายพุทธิพงศ์ ย่างนอก
- 643021273-8 นายศตวรรษ มูลสันเทียะ

■ CRITERION

ค่าวัดในการ Split โดยตัดสินว่าข้อมูลตัวอย่างนั้นจะถูกจัดประเภทไปยังกลุ่มใดโดยค่าไหนมีให้เลือกหลากหลาย เช่น gini entropy logloss

■ MAX_LEAFT_NODES

การกำหนดจำนวนสูงสุดของโหนดในต้นไม้ตัดสินใจ โดยกำหนดให้การ Splitting หยุดลงเมื่อจำนวนโหนดในถิ่นขึ้นต่อไปไม่สามารถตัดสินใจได้

CRITERION



GINI

$$\square gini(D) = 1 - \sum_{j=1}^n p_j^2$$

Gini Impurity เป็นตัววัดความ "ไม่บริสุทธิ์" ของข้อมูลในโหมด Decision Tree หมายถึงจำนวนข้อมูลที่มี class label ผสมกันอยู่มากน้อยเพียงใด คะแนน Gini ก็ต่อменноแสดงถึงกลุ่มข้อมูลที่ "บริสุทธิ์" มากกว่า และเป็นการแบ่งกลุ่มที่ดีกว่า

GINI

$$\square gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

เป็นค่า gini ของ feature ยิ่งน้อยยิ่งดี

$$\square \Delta gini(A) = gini(D) - gini_A(D)$$

Reduction in Impurity คำนวณจาก ความแตกต่าง ของ ค่า ความไม่บริสุทธิ์ ของ node หลัก (parent node) กับ ค่าความไม่บริสุทธิ์ ของ node ย่อย (child nodes)

Entropy

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

Entropy ของข้อมูล เป็นตัววัด ความไม่แน่นอน ของข้อมูล หรือพูดอีกนัยหนึ่งคือ เป็นตัววัด ความสุ่ม ของข้อมูล

ค่า Entropy สูง หมายถึง ข้อมูลมีความไม่แน่นอนสูง หรือมีความสุ่มสูง

ค่า Entropy ต่ำ หมายถึง ข้อมูลมีความแน่นอนสูง หรือ มีความสุ่มต่ำ

Entropy

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

เป็นค่า Entropy ของ feature ยึงน้อยยึงดี

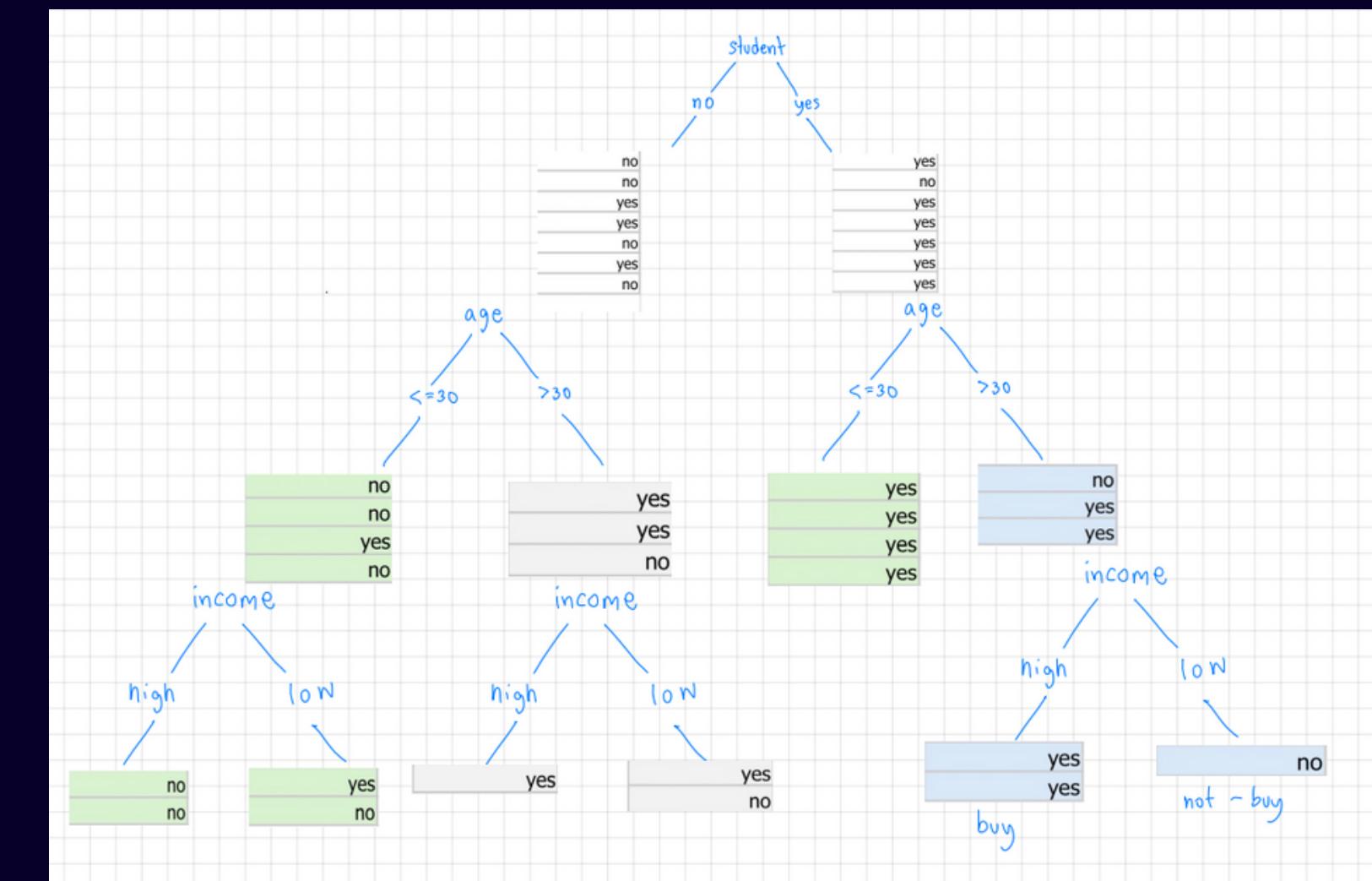
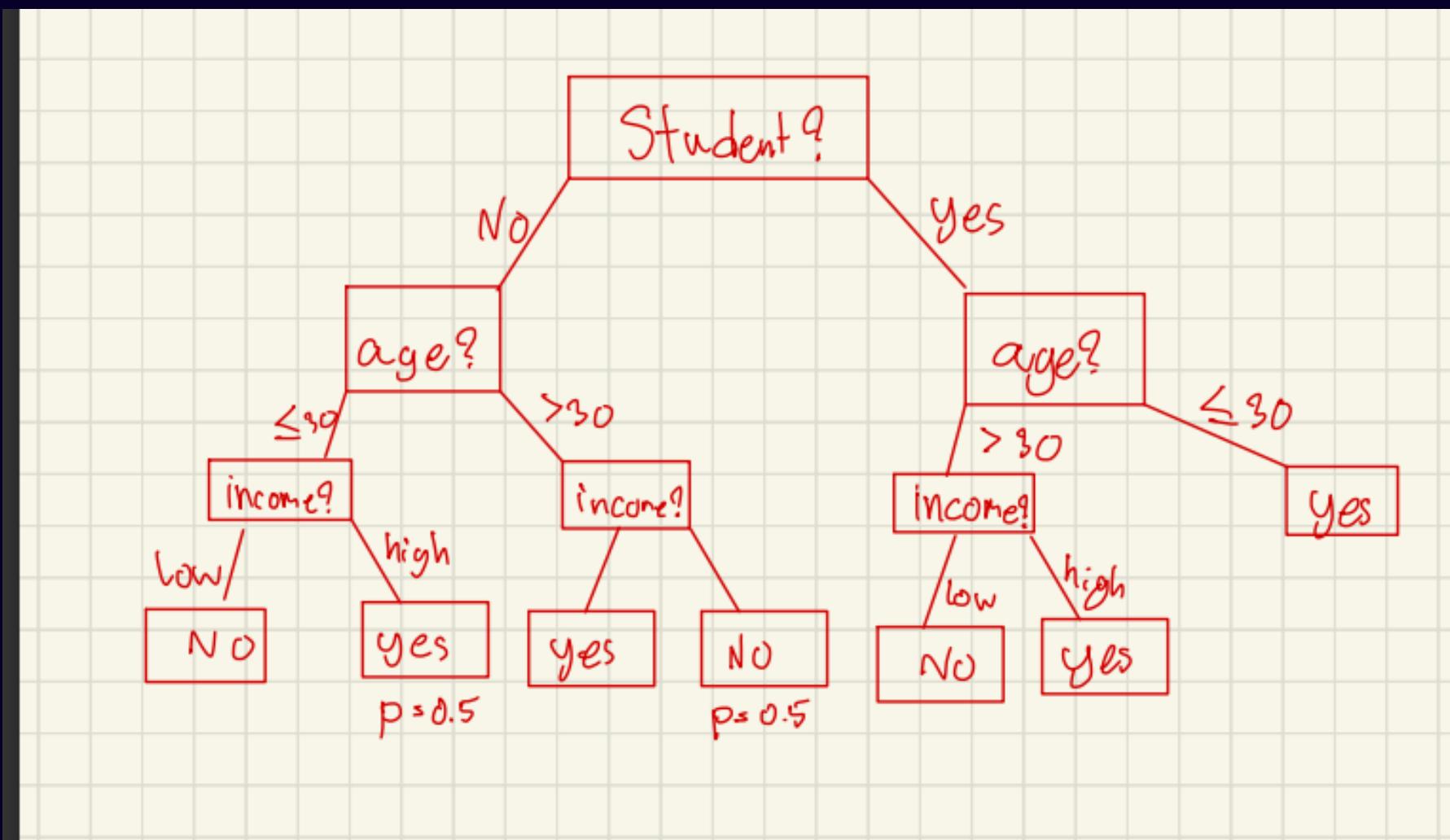
$$Gain(A) = Info(D) - Info_A(D)$$

Gain คำนวณจาก ความแตกต่าง ของ ค่าความไม่แน่นอน ของ node หลัก (parent node) กับ ค่าความไม่แน่นอน ของ node ย่อย (child nodes)

TRAINING DATA SET: WHO BUYS COMPUTER?

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
>30	high	no	fair	yes
<=30	low	no	fair	yes
<=30	high	yes	fair	yes
>30	low	yes	excellent	no
>30	high	yes	excellent	yes
<=30	low	no	fair	no
<=30	low	yes	fair	yes
<=30	low	yes	fair	yes
<=30	low	yes	excellent	yes
>30	low	no	excellent	yes
>30	high	yes	fair	yes
>30	low	no	excellent	no

Full Growth tree



$$gini(D) = 0.459$$

$$gini_{age}(D) = 0.457 ; \Delta gini_{age}(D) = 0.02$$

$$gini_{income}(D) = 0.457 ; \Delta gini_{income}(D) = 0.02$$

$$gini_{credit}(D) = 0.428 ; \Delta gini_{credit}(D) = 0.031$$

$$gini_{student}(D) = 0.366 ; \Delta gini_{student}(D) = 0.093$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.938 = 0.002$$

$$Gain(income) = Info(D) - Info_{income}(D) = 0.940 - 0.938 = 0.002$$

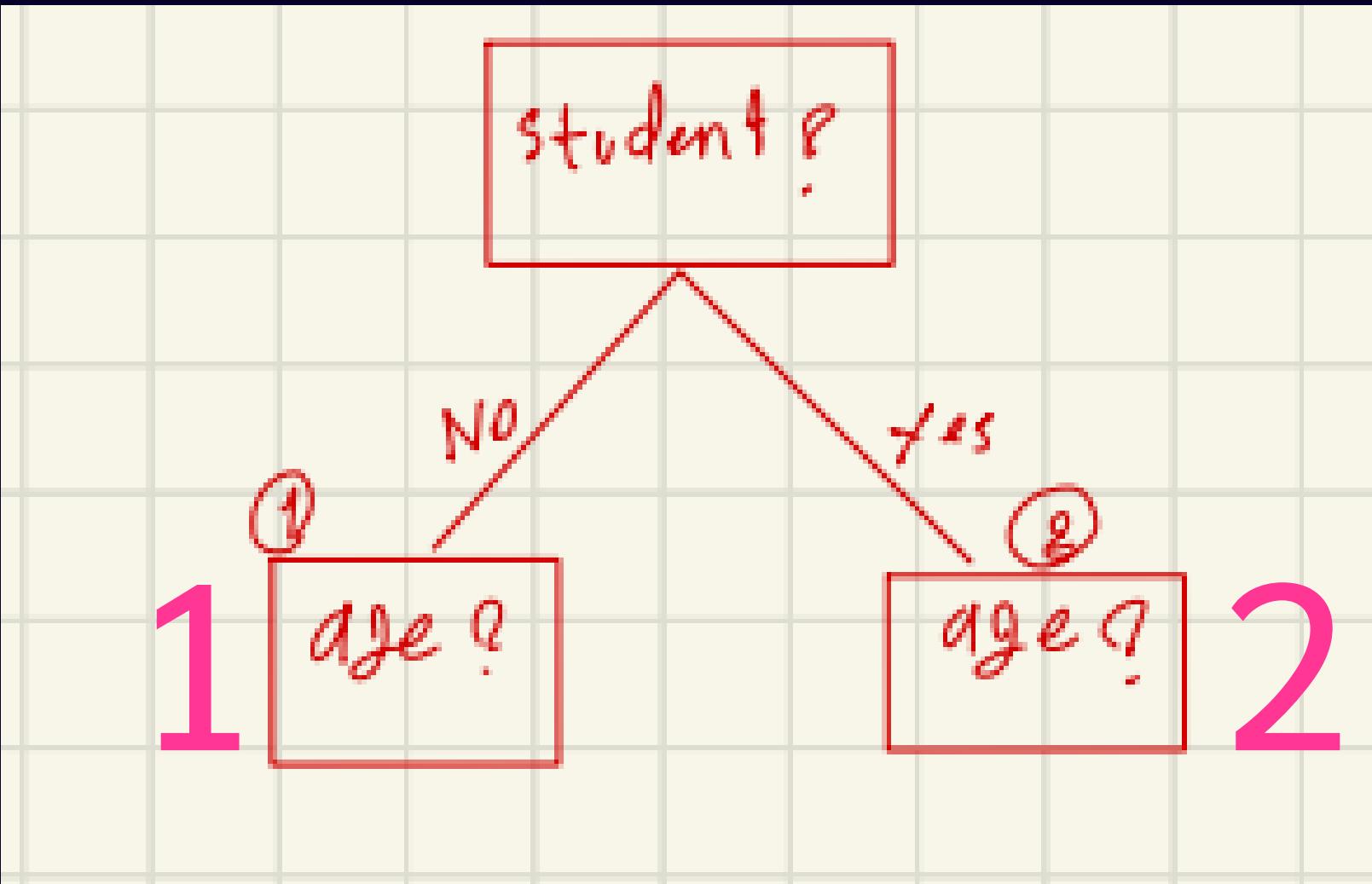
$$Gain(student) = Info(D) - Info_{student}(D) = 0.940 - 0.788 = 0.152$$

$$\begin{aligned} Gain(credit_rating) &= Info(D) - Info_{credit_rating}(D) \\ &= 0.940 - 0.892 = 0.048 \end{aligned}$$

∴ Root Node is student

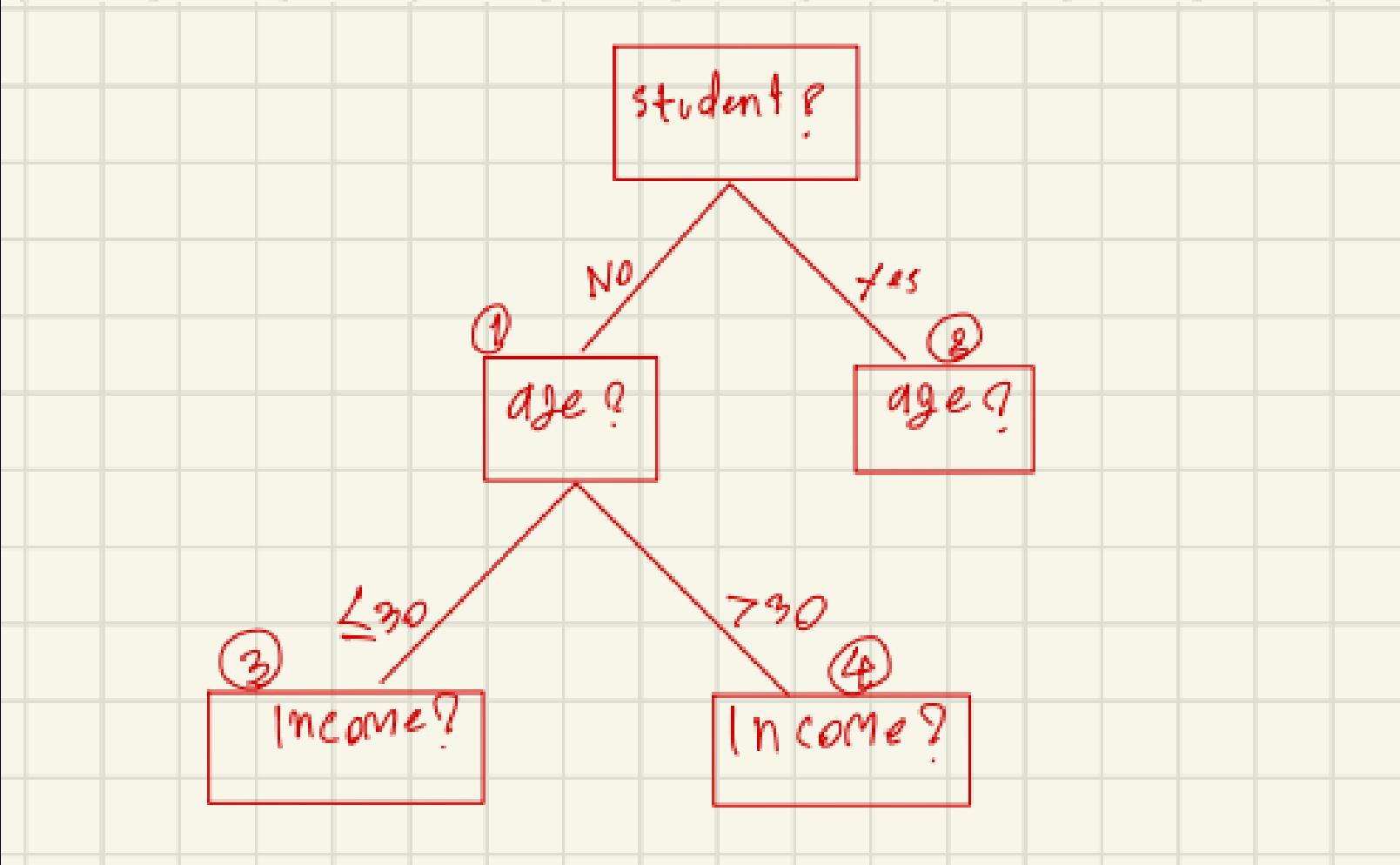
การตัดของต้นไม้

Entropy vs gini



node 1 $\Delta\text{gini}(\text{Student}=\text{no}, \text{age}) = 0.0854$

node 2 $\Delta\text{gini}(\text{Student}=\text{yes}, \text{age}) = 0.0547$



แบ่ง node 1 (ทางซ้าย) ก่อน

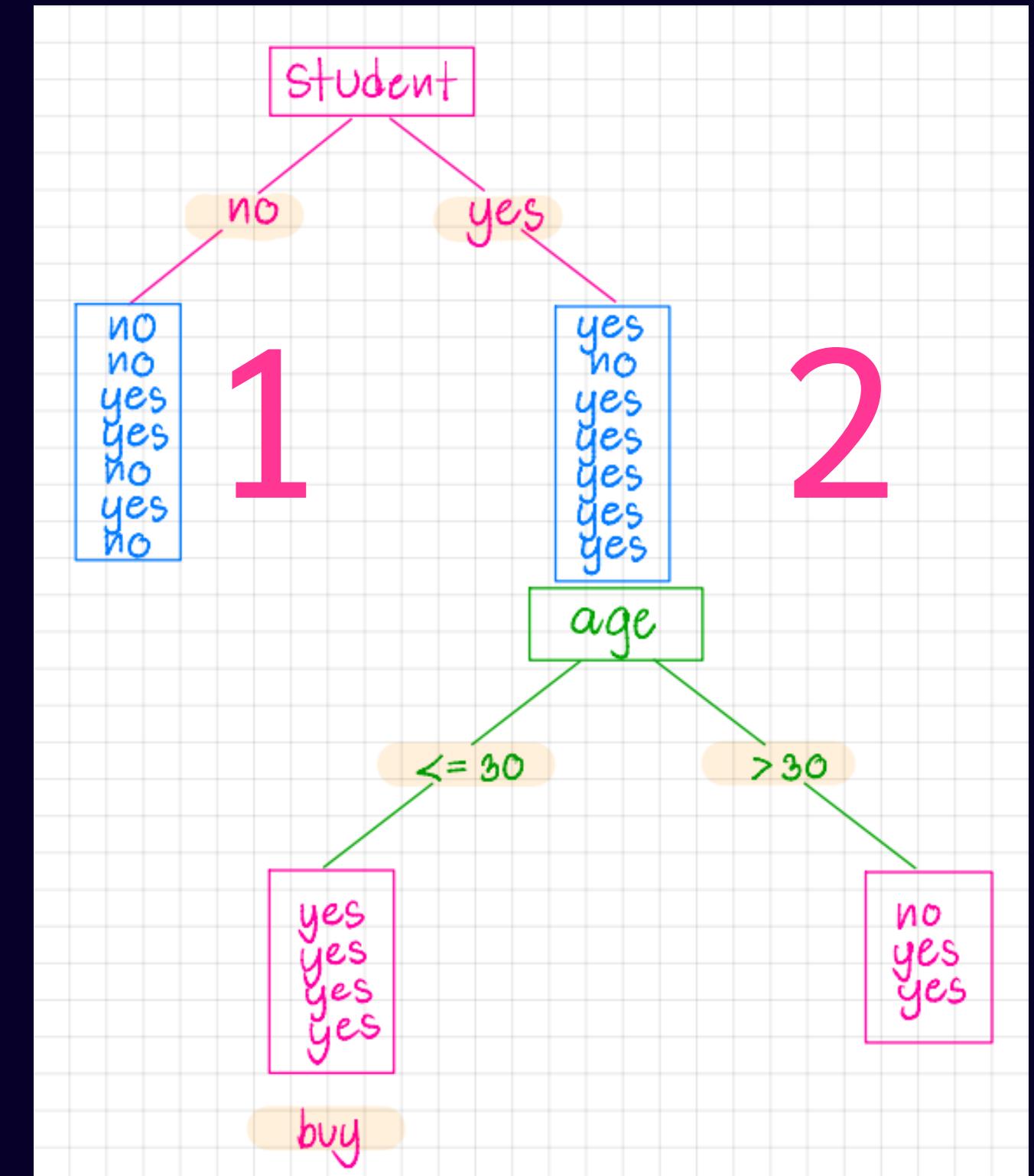
การตัดของต้นไม้

Entropy vs gini

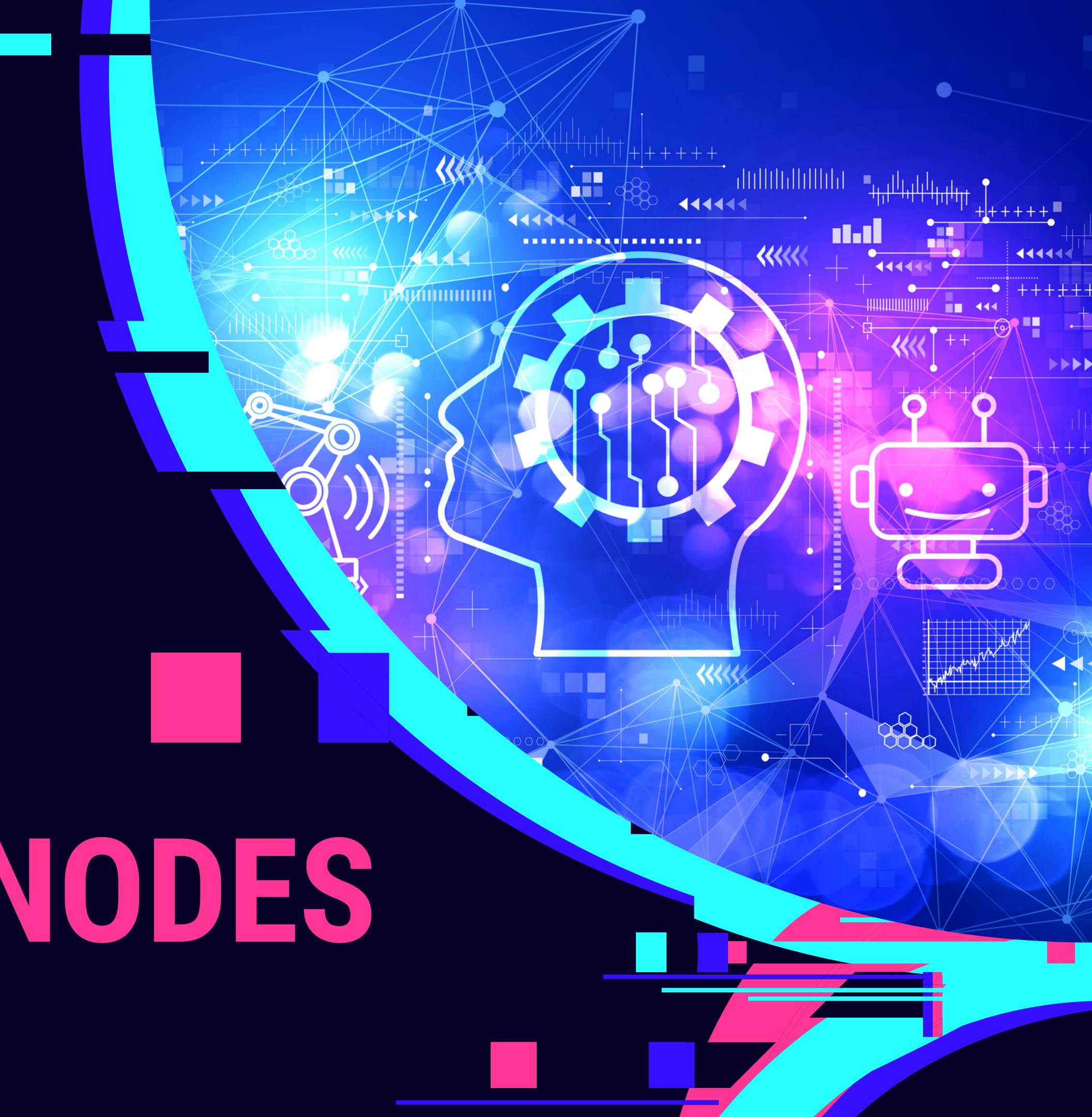
node 1 gain(Student=no, age) = 0.1282

node 2 gain(Student=yes, age) = 0.1986

แบ่ง node 2 (ทางขวา) ก่อน



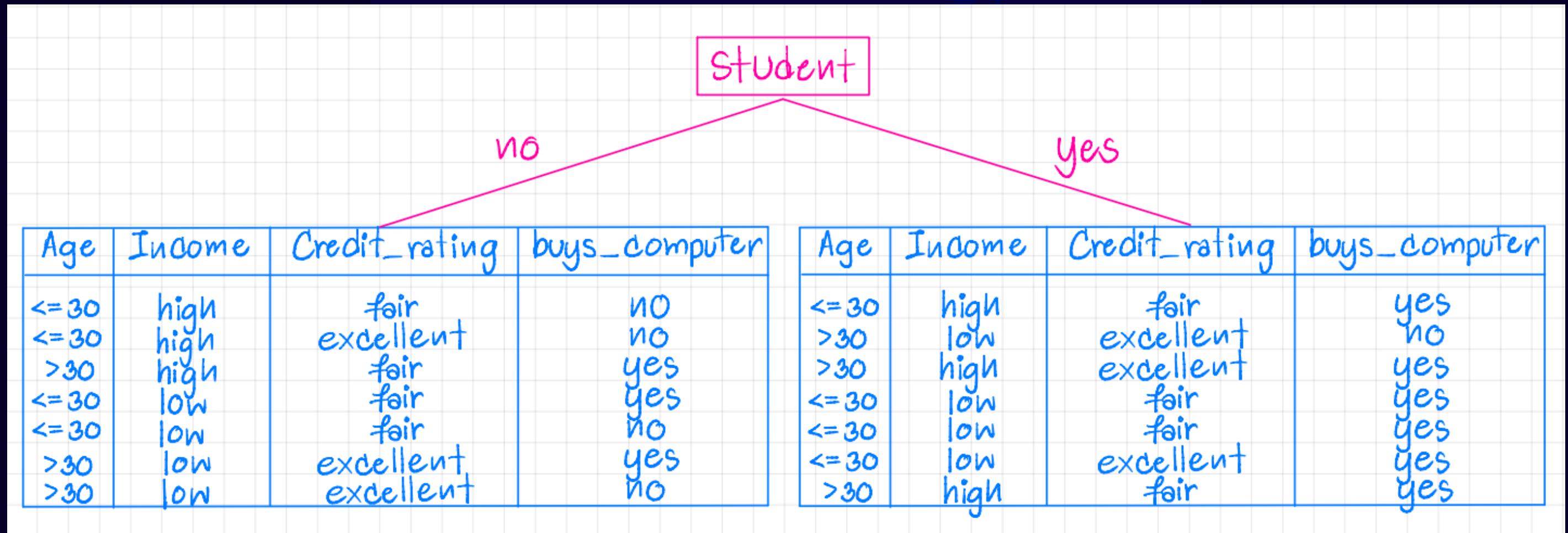
MAX.LEAF.NODES



TRAINING DATA SET: WHO BUYS COMPUTER?

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
>30	high	no	fair	yes
<=30	low	no	fair	yes
<=30	high	yes	fair	yes
>30	low	yes	excellent	no
>30	high	yes	excellent	yes
<=30	low	no	fair	no
<=30	low	yes	fair	yes
<=30	low	yes	fair	yes
<=30	low	yes	excellent	yes
>30	low	no	excellent	yes
>30	high	yes	fair	yes
>30	low	no	excellent	no

MAX_LEAF_NODES = 2



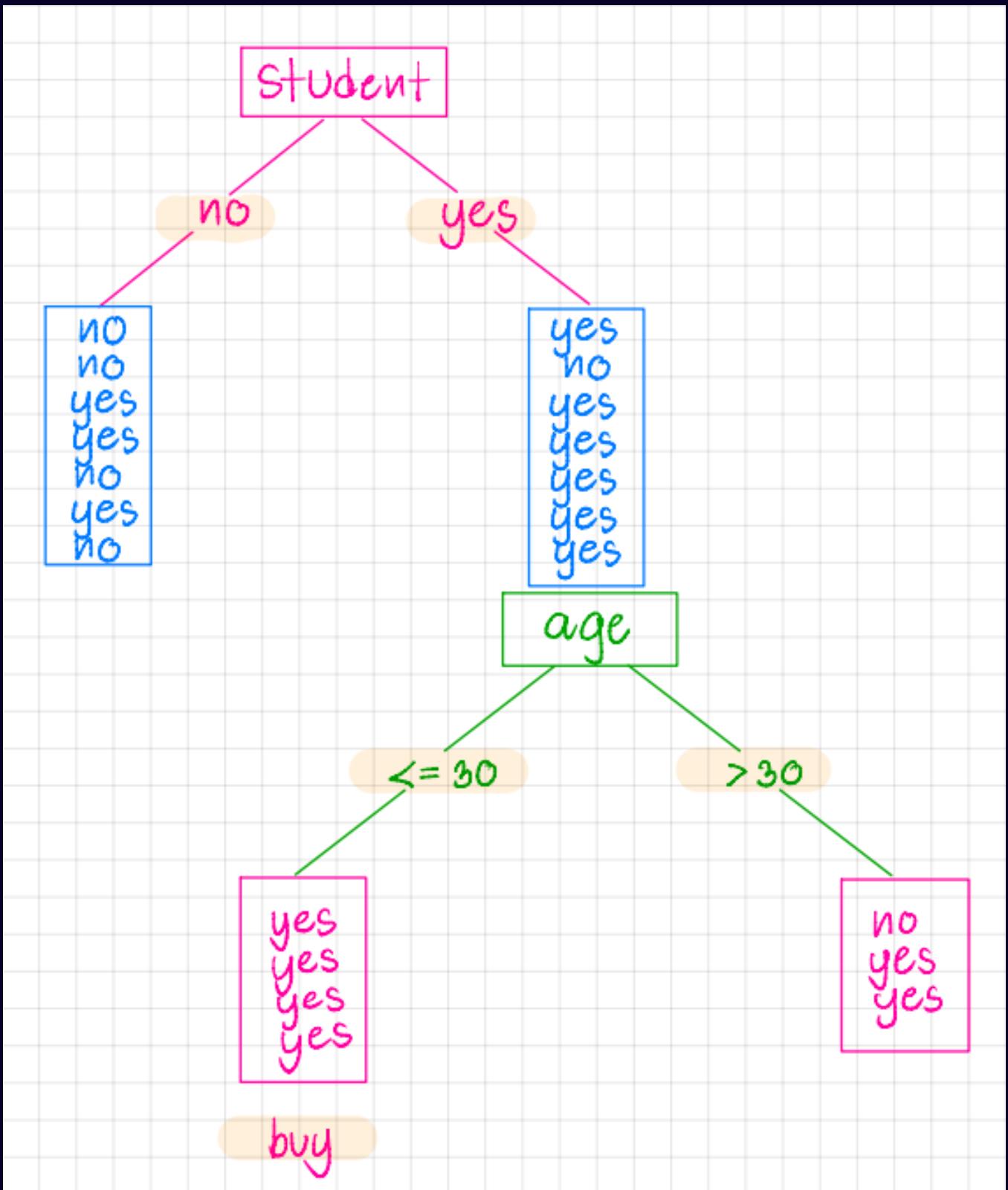
INFORMATION GAINED

- ↗ Class P: buys_computer = "yes" → 9
- ↗ Class N: buys_computer = "no" → 5

$$\text{Info}(D) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) \\ = 0.940$$

$$\begin{aligned}\text{Gain}(\text{age}) &= \text{Info}(D) - \text{Info}_{\text{age}}(D) = 0.940 - 0.938 = 0.002 \\ \text{Gain}(\text{income}) &= \text{Info}(D) - \text{Info}_{\text{income}}(D) = 0.940 - 0.938 = 0.002 \\ \text{Gain}(\text{student}) &= \text{Info}(D) - \text{Info}_{\text{student}}(D) = 0.940 - 0.788 = 0.152 \\ \text{Gain}(\text{credit_rating}) &= \text{Info}(D) - \text{Info}_{\text{credit_rating}}(D) \\ &= 0.940 - 0.892 = 0.048\end{aligned}$$

MAX_LEAF_NODES = 3



$$\text{Gain}(\text{age}) = \text{Info}_{\text{student:yes}}(D) - \text{Info}_{\text{age}}(D) = 0.592 - 0.393 = 0.199 \times$$

$$\text{Gain}(\text{income}) = \text{Info}_{\text{student:yes}}(D) - \text{Info}_{\text{income}}(D) = 0.592 - 0.463 = 0.129$$

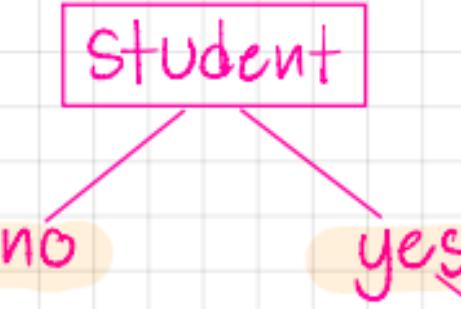
$$\begin{aligned}\text{Gain}(\text{credit_rating}) &= \text{Info}_{\text{student:yes}}(D) - \text{Info}_{\text{credit_rating}}(D) \\ &= 0.592 - 0.393 = 0.199 \times\end{aligned}$$

$$\begin{aligned}\text{Gain}(\text{student: no, age}) &= \text{Info}_{\text{student: no}}(D) - \text{Info}_{\text{age}}(D) \\ &= 0.985 - 0.128 = 0.128 \times\end{aligned}$$

$$\begin{aligned}\text{Gain}(\text{student: no, income}) &= \text{Info}_{\text{student: no}}(D) - \text{Info}_{\text{income}}(D) \\ &= 0.985 - 0.965 = 0.02\end{aligned}$$

$$\begin{aligned}\text{Gain}(\text{student: no, credit_rating}) &= \text{Info}_{\text{student: no}}(D) - \text{Info}_{\text{credit_rating}}(D) \\ &= 0.985 - 0.965 = 0.02\end{aligned}$$

Gain(student: yes)	0.199
Gain(student: no)	0.128



$$\text{Gain}(\text{age}) = \text{Info}_{\text{student:yes}}(D) - \text{Info}_{\text{age}}(D) = 0.592 - 0.393 = 0.199 \times$$

$$\text{Gain}(\text{income}) = \text{Info}_{\text{student:yes}}(D) - \text{Info}_{\text{income}}(D) = 0.592 - 0.463 = 0.129$$

$$\begin{aligned}\text{Gain}(\text{credit_rating}) &= \text{Info}_{\text{student:yes}}(D) - \text{Info}_{\text{credit_rating}}(D) \\ &= 0.592 - 0.393 = 0.199 \times\end{aligned}$$

$\text{Gain}(\text{student:yes}, \text{age} > 30, \text{income})$

$$= \text{Info}_{\text{student:yes}, \text{age} > 30}^{(D)} - \text{Info}_{\text{student:yes}, \text{age} > 30, \text{income}}^{(D)}$$
$$= 0.918 - 0 = 0.918$$

$\text{Gain}(\text{student:yes}, \text{age} > 30, \text{credit rating})$

$$= \text{Info}_{\text{student:yes}, \text{age} > 30}^{(D)} - \text{Info}_{\text{student:yes}, \text{age} > 30, \text{credit_rating}}^{(D)}$$
$$= 0.918 - 0.667 = 0.251$$

$\text{Gain}(s:\text{yes}, c = \text{excellent}, \text{age})$

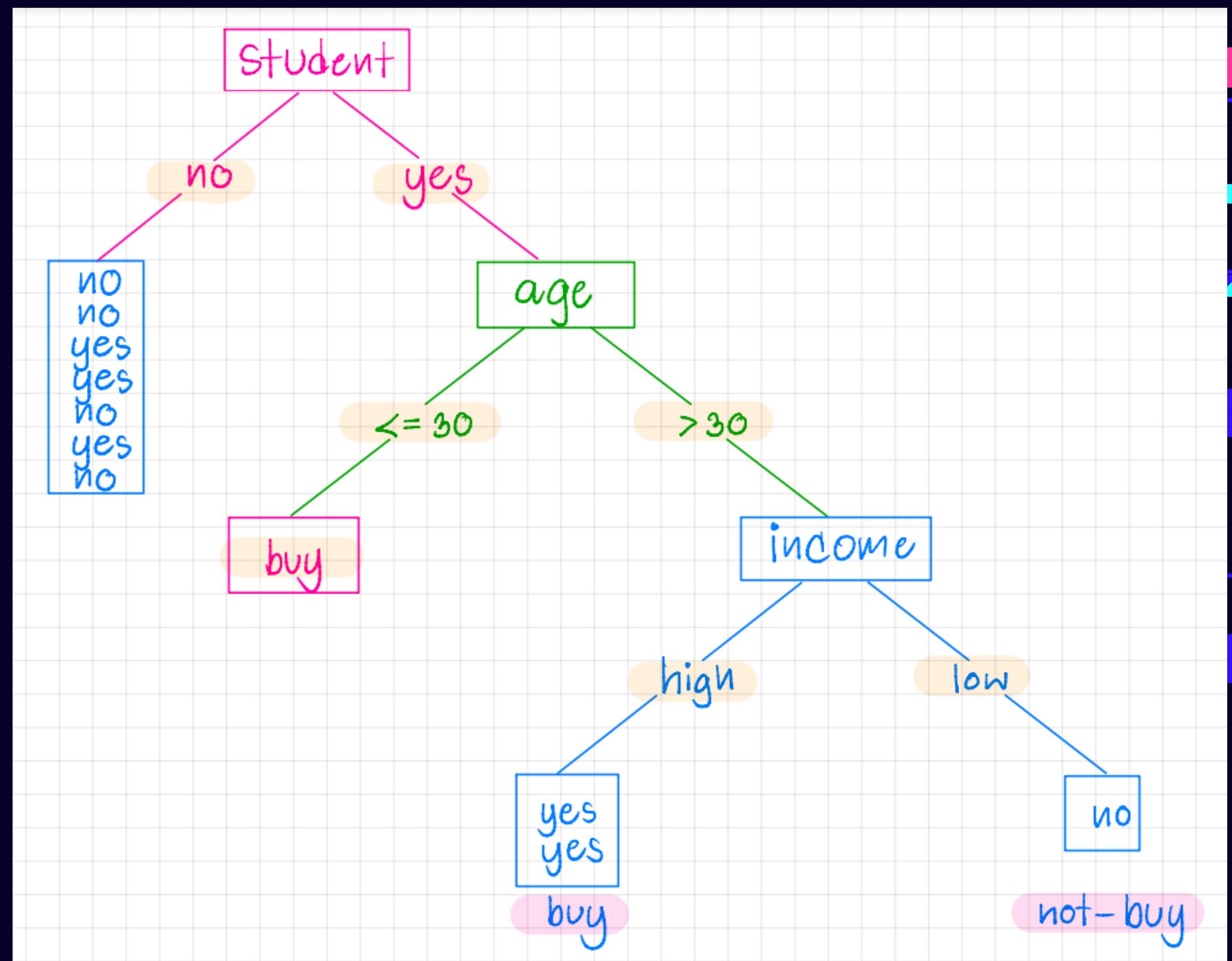
$$= \text{Info}_{s:\text{yes}, c = \text{excellent}}^{(D)} - \text{Info}_{s:\text{yes}, c = \text{excellent}, \text{age}}^{(D)}$$
$$= 0.918 - 0.667 = 0.251$$

$\text{Gain}(s:\text{yes}, c = \text{excellent}, \text{income})$

$$= \text{Info}_{s:\text{yes}, c = \text{excellent}}^{(D)} - \text{Info}_{s:\text{yes}, c = \text{excellent}, \text{income}}^{(D)}$$
$$= 0.918 - 0.667 = 0.251$$

$\text{Gain}(\text{student:yes}, \text{age} > 30, \text{income})$	0.918
$\text{Gain}(\text{student:yes}, \text{age} > 30, \text{credit rating})$	0.251
$\text{Gain}(s:\text{yes}, c = \text{excellent}, \text{age})$	0.251
$\text{Gain}(s:\text{yes}, c = \text{excellent}, \text{income})$	0.251

MAX_LEAF_NODES = 4

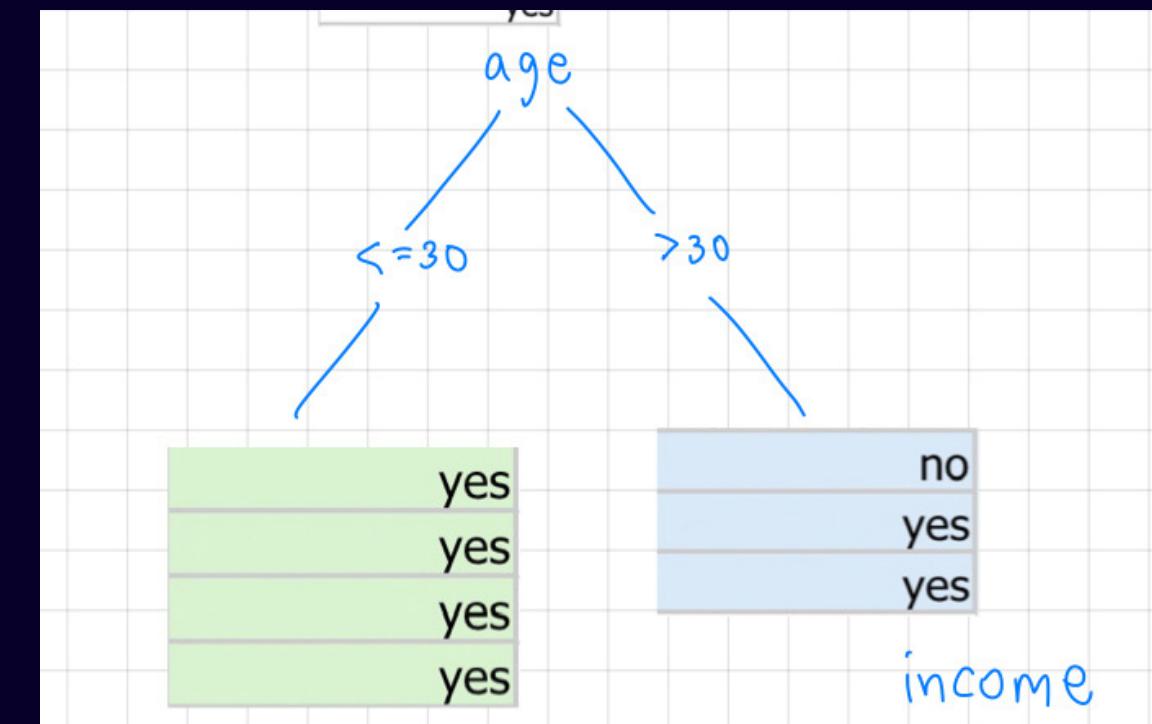


Step 1 : an expected info gain
classify them

→ Class P: buys_computer = "yes" → 4

→ Class N: buys_computer = "no" → 0

$$\begin{aligned} \text{Info}_{\text{student}}: \text{yes}, \text{age} \leq 30 & \stackrel{(D)}{=} \\ &= -\frac{4}{4} \log_2 \left(\frac{4}{4} \right) - \frac{0}{4} \log_2 \left(\frac{0}{4} \right) \\ &= 0 \end{aligned}$$



AGE <= 30

Gain(student: yes, age <= 30)

$$\begin{aligned} &= \text{Info}_{\text{student}: \text{yes}, \text{age} \leq 30}^{(D)} - \text{Info}_{\text{student}: \text{yes}, \text{age} \leq 30, \text{income}}^{(D)} \\ &= 0.918 - 0 = 0.918 \end{aligned}$$

Gain(student: yes, age > 30)

$$\begin{aligned} &= \text{Info}_{\text{student}: \text{yes}, \text{age} > 30}^{(D)} - \text{Info}_{\text{student}: \text{yes}, \text{age} > 30, \text{credit_rating}}^{(D)} \\ &= 0.918 - 0.667 = 0.251 \end{aligned}$$

AGE > 30

Gain(student: yes, age > 30, income)

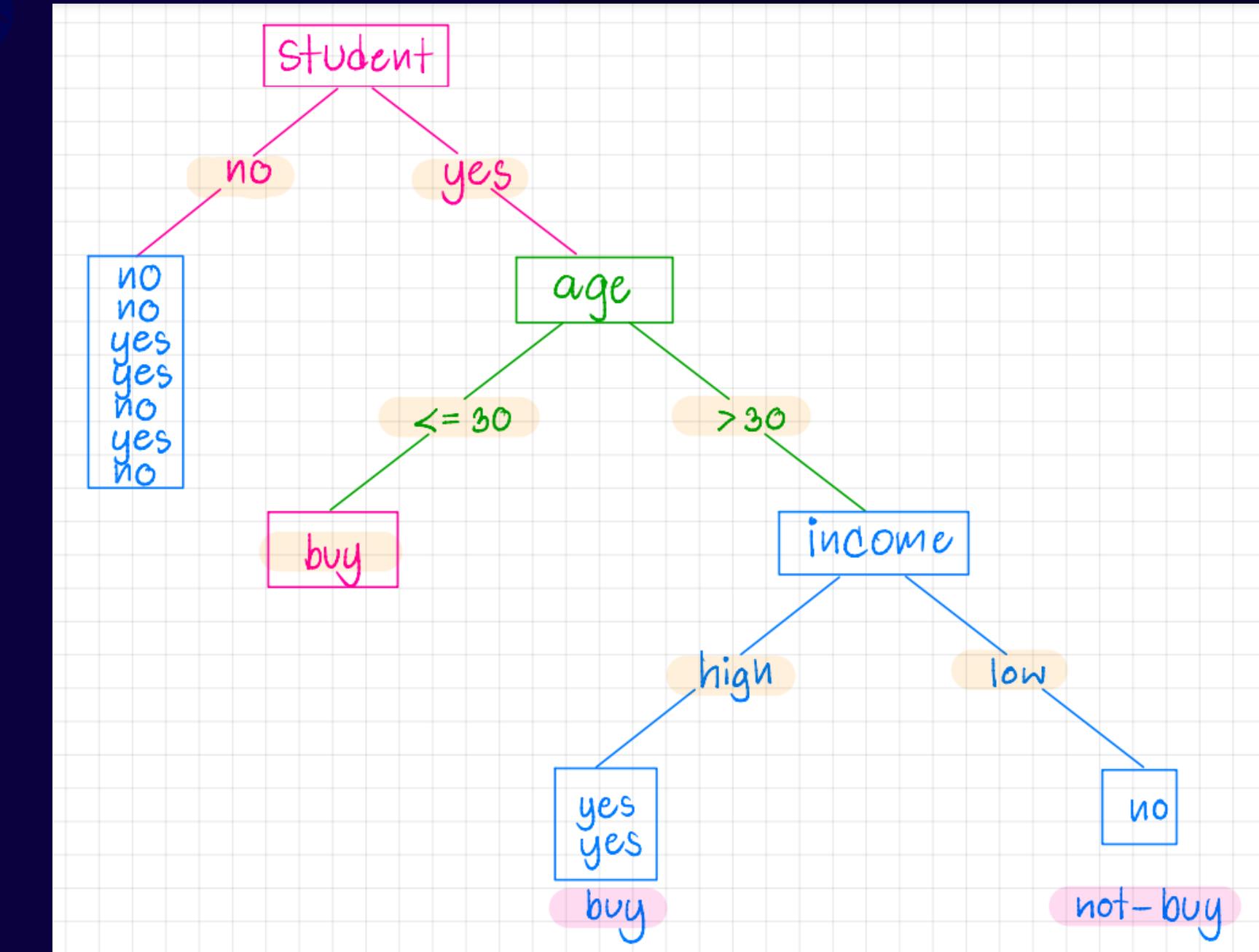
0.918

Gain(student: yes, age > 30, credit rating)

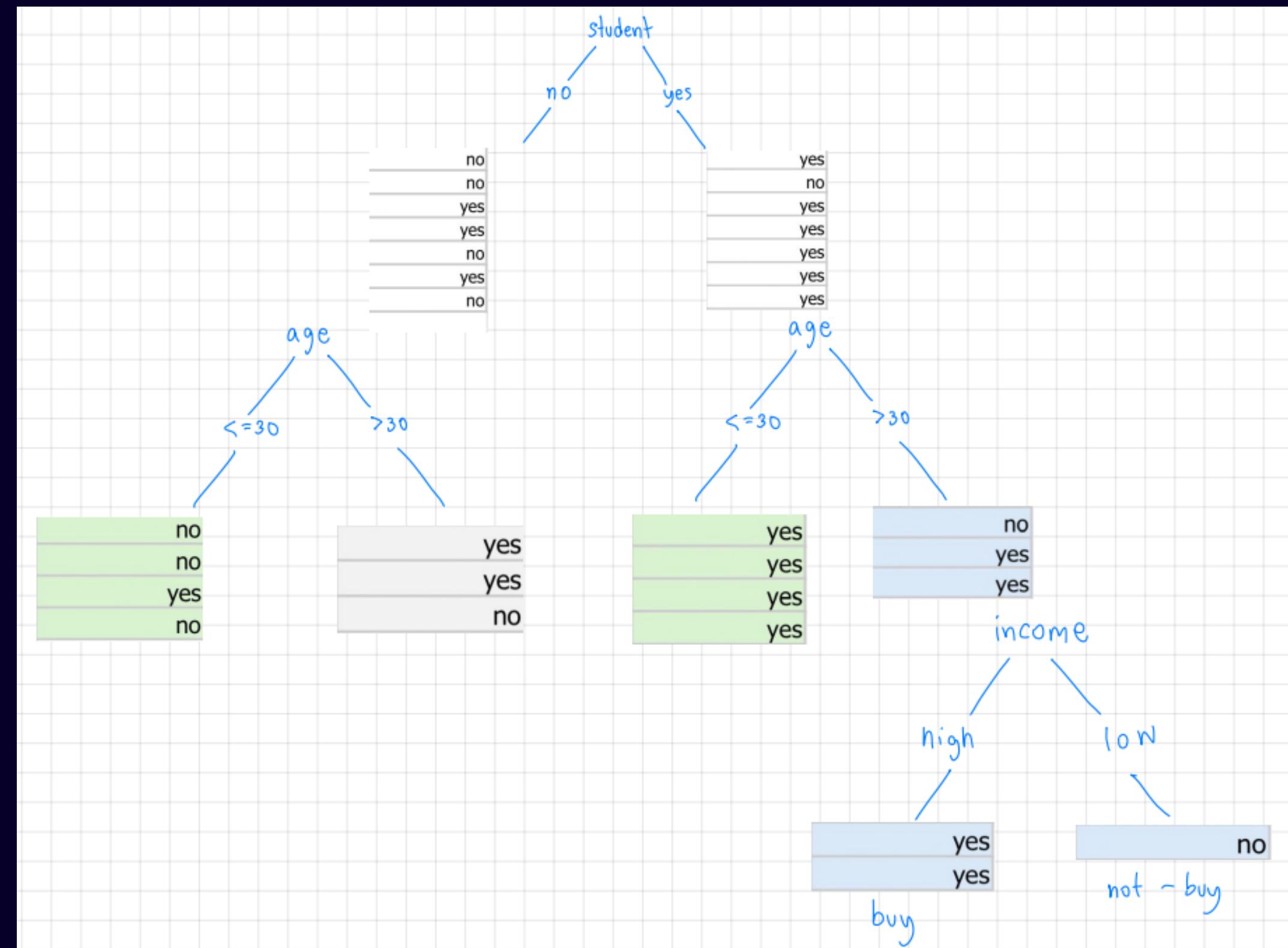
0.251

student	yes	age	>30	income = low
age	income	student	credit_rating	buys_computer
>30	low	yes	excellent	no

student	yes	age	>30	income = high
age	income	student	credit_rating	buys_computer
>30	high	yes	excellent	yes
>30	high	yes	fair	yes

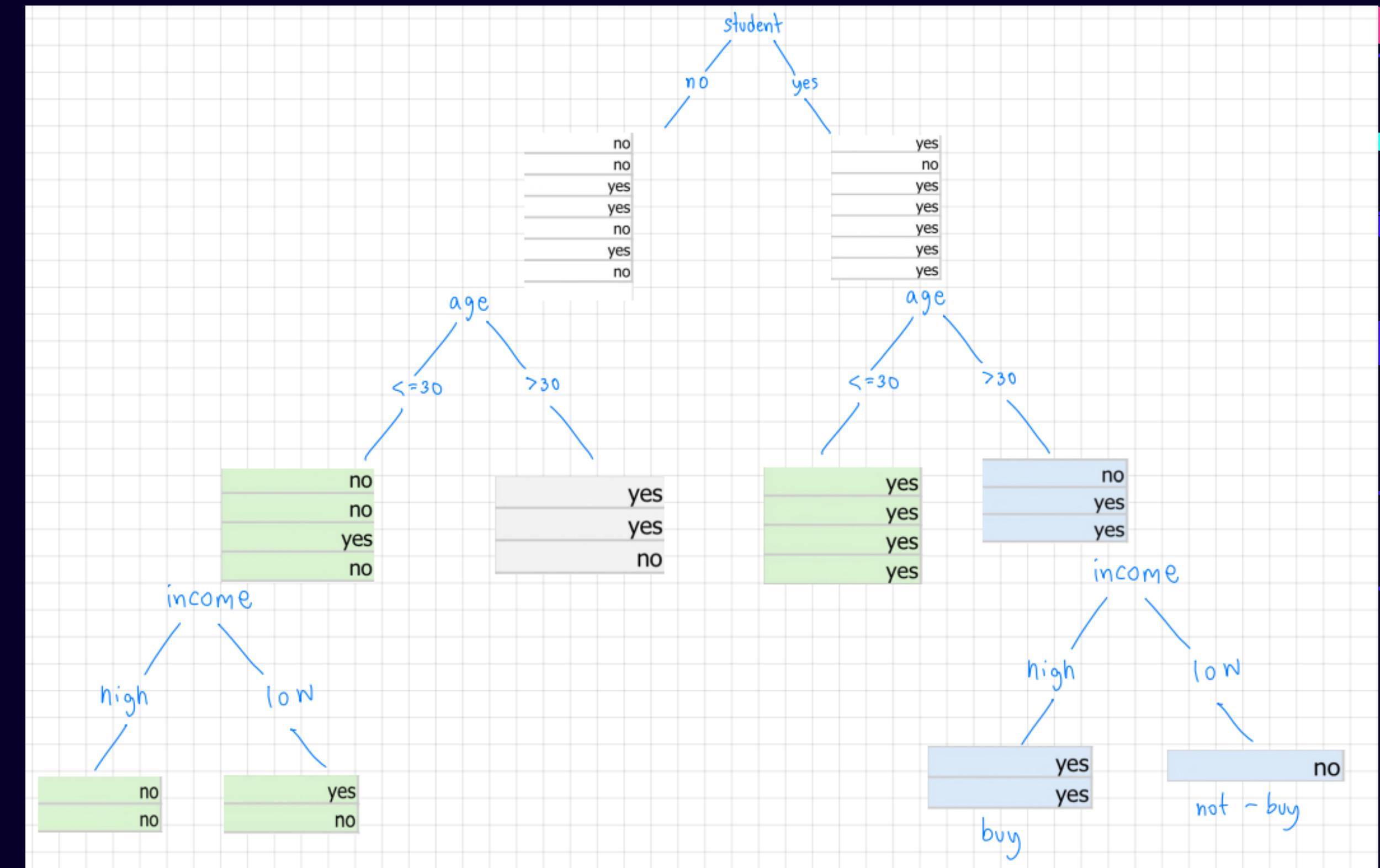


MAX_LEAF_NODES = 5



$$\text{Gain}(\text{age}) = \text{Info}_{\text{Student: no}(D)} - \text{Info}_{\text{age}(D)} = 0.9852 - 0.8571 = 0.1281 \quad \checkmark$$
$$\text{Gain}(\text{income}) = \text{Info}_{\text{Student: no}(D)} - \text{Info}_{\text{income}(D)} = 0.9852 - 0.9649 = 0.0203$$
$$\text{Gain}(\text{credit_rating}) = \text{Info}_{\text{Student: no}(D)} - \text{Info}_{\text{credit_rating}(D)} = 0.9852 - 0.9649 = 0.0203$$

MAX_LEAF_NODES = 6



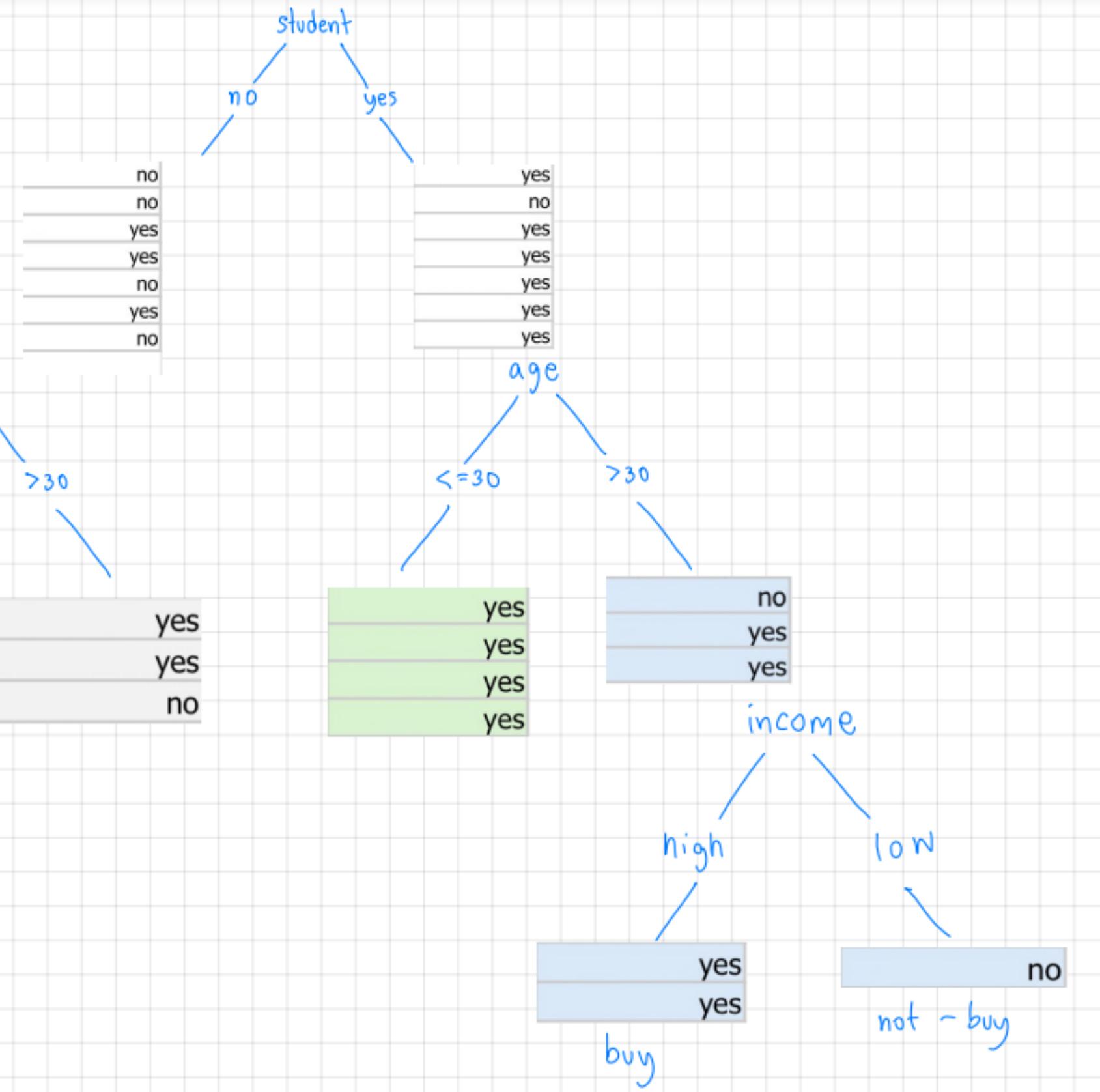
$$\text{Gain}(\text{student: no}, \text{age} \leq 30, \text{income}) = \text{Info}_{\text{student: no}, \text{age} \leq 30: \text{no}(D)} - \text{Info}_{\text{income}}(D) = 0.8113 - 0.5 = 0.3113 \quad \checkmark$$

$$\text{Gain}(\text{student: no}, \text{age} \leq 30, \text{credit_rating}) = \text{Info}_{\text{student: no}, \text{age} \leq 30: \text{no}(D)} - \text{Info}_{\text{credit_rating}}(D) = 0.8113 - 0.6887 = 0.1226$$

$$\text{Gain}(\text{student: no}, \text{age} > 30, \text{income}) = \text{Info}_{\text{student: no}, \text{age} > 30: \text{no}(D)} - \text{Info}_{\text{income}}(D) = 0.918 - 0.667 = 0.251$$

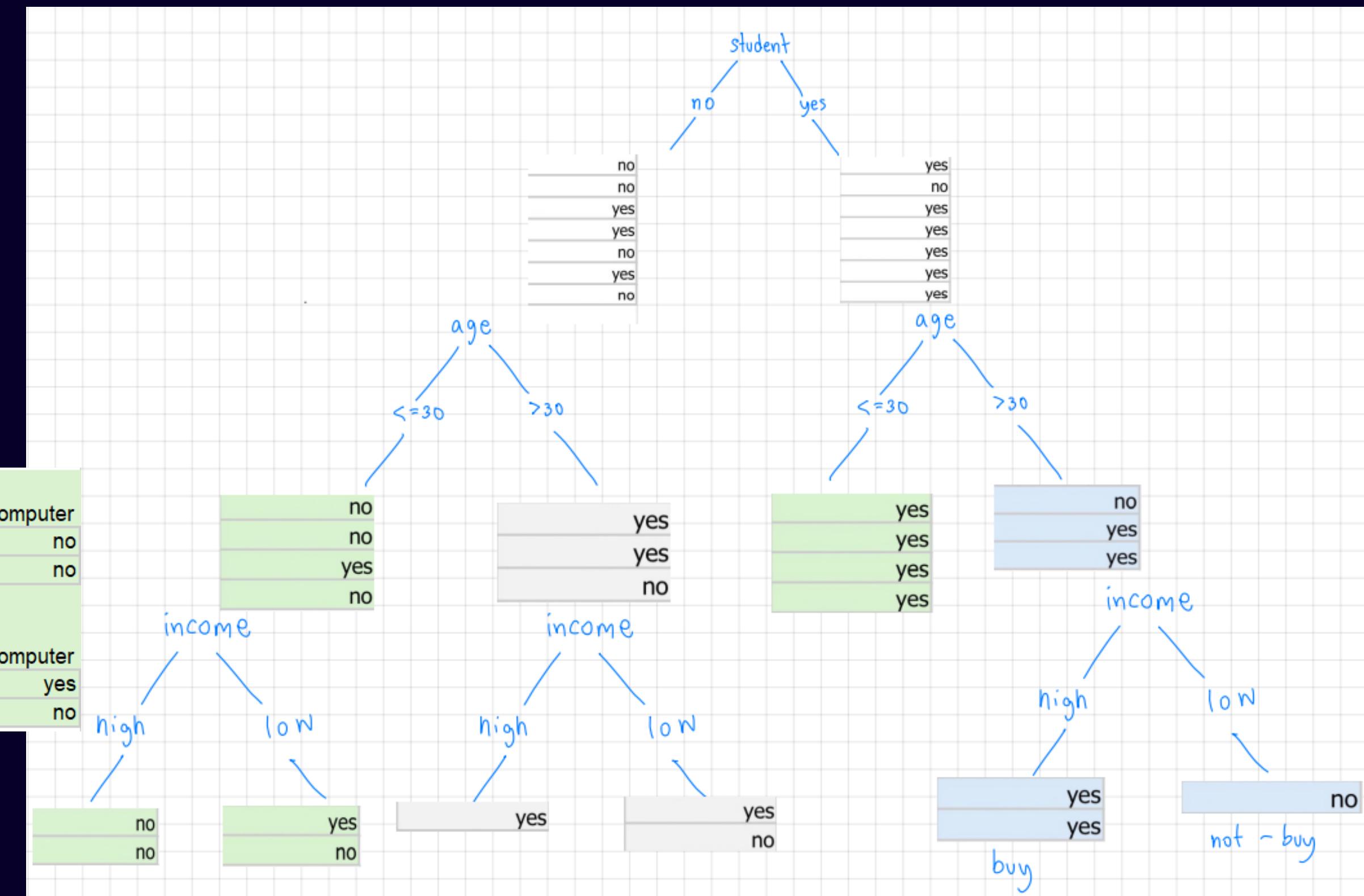
$$\text{Gain}(\text{student: no}, \text{age} > 30, \text{credit_rating}) = \text{Info}_{\text{student: no}, \text{age} > 30: \text{no}(D)} - \text{Info}_{\text{credit_rating}}(D) = 0.918 - 0.667 = 0.251$$

student	no	age <= 30	income = high	
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
student	no	age <= 30	income = low	
age	income	student	credit_rating	buys_computer
<=30	low	no	fair	yes
<=30	low	no	fair	no



MAX_LEAF_NODES = 7

student	no	age <= 30	income = high		
age	income	student	credit_rating	buys_computer	
<=30	high	no	fair		no
<=30	high	no	excellent		no
student	no	age <= 30	income = low		
age	income	student	credit_rating	buys_computer	
<=30	low	no	fair		yes
<=30	low	no	fair		no



student	no	age > 30	income = high		
age	income	student	credit_rating	buys_computer	
>30	high	no	fair	yes	
student	no	age > 30	income = low		
age	income	student	credit_rating	buys_computer	
>30	low	no	excellent	yes	
>30	low	no	excellent	no	

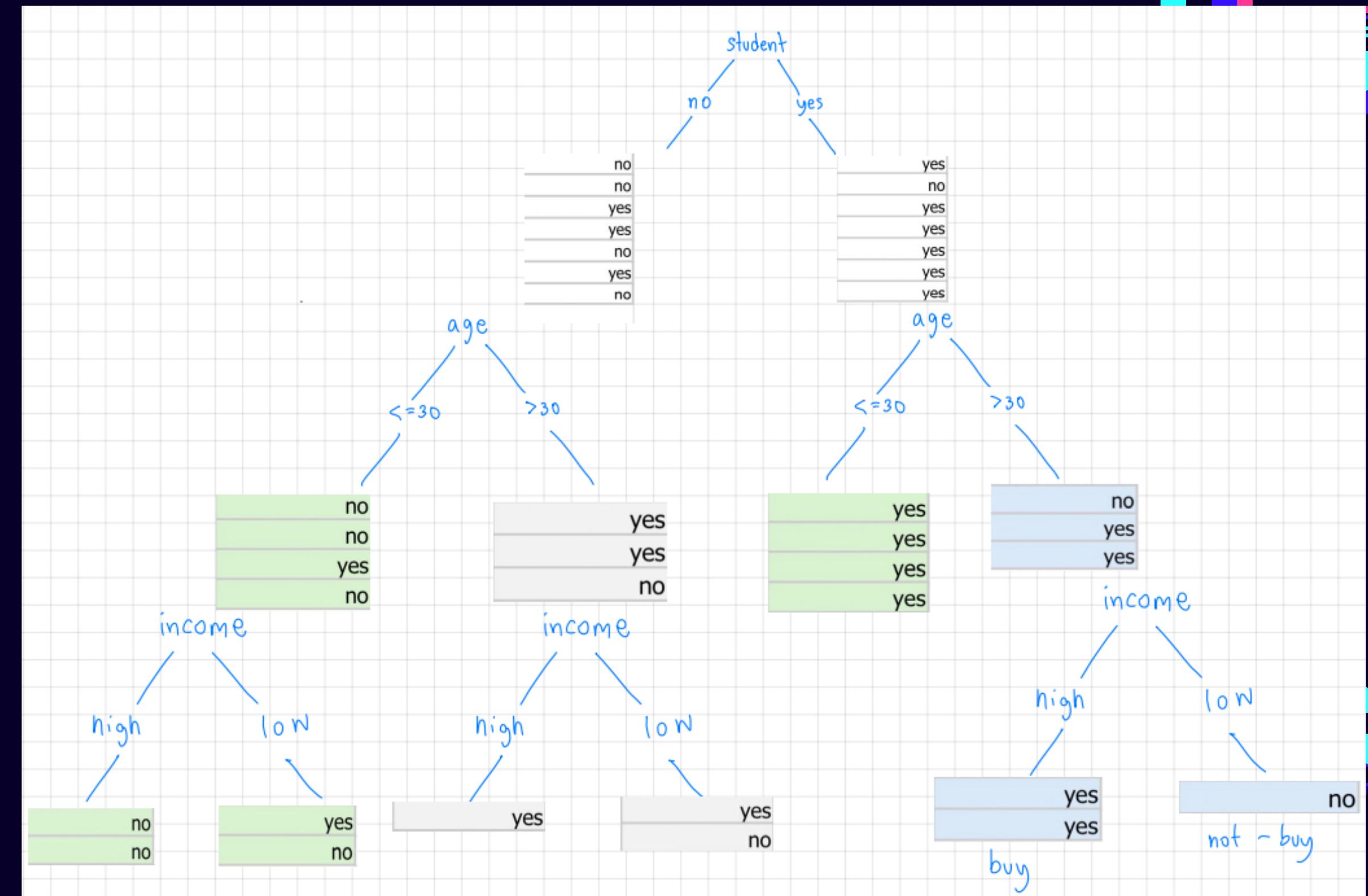
student	no	age > 30	credit = fair		
age	income	student	credit_rating	buys_computer	
>30	high	no	fair	yes	
student	no	age > 30	credit = excellent		
age	income	student	credit_rating	buys_computer	
>30	low	no	excellent	yes	
>30	low	no	excellent	no	

$$\text{Gain}(\text{student: no, age} > 30, \text{income}) = \text{Info}_{\text{student: no, age} > 30: \text{no}(D)} - \text{Info}_{\text{income}(D)} = 0.918 - 0.667 = 0.251$$

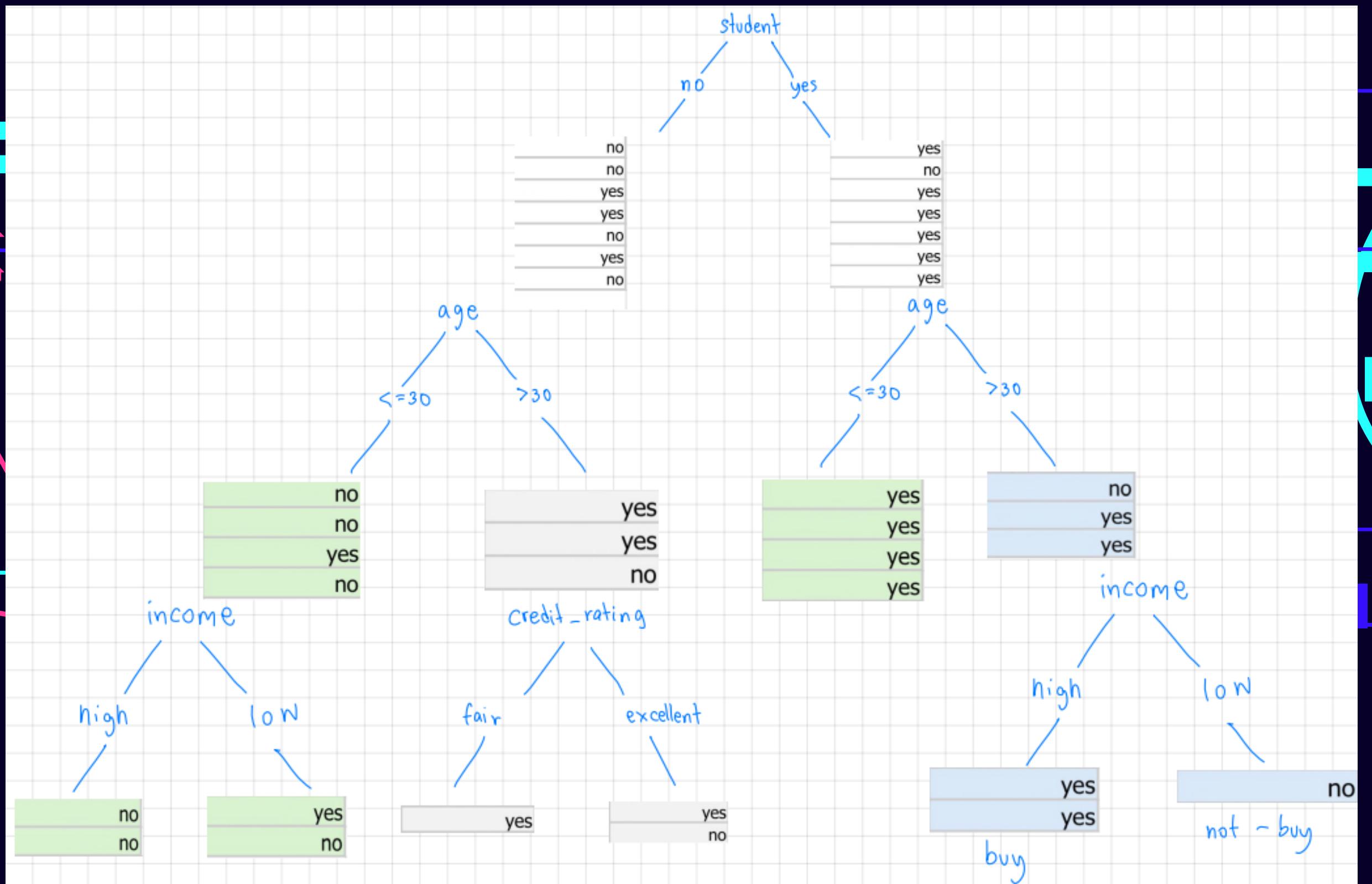
$$\text{Gain}(\text{student: no, age} > 30, \text{credit_rating}) = \text{Info}_{\text{student: no, age} > 30: \text{no}(D)} - \text{Info}_{\text{credit_rating}(D)} = 0.918 - 0.667 = 0.251$$

student	no	age > 30	income = high	
age	income	student	credit_rating	buys_computer
>30	high	no	fair	yes
student	no	age > 30	income = low	
age	income	student	credit_rating	buys_computer
>30	low	no	excellent	yes
>30	low	no	excellent	no
student	no	age > 30	credit = fair	
age	income	student	credit_rating	buys_computer
>30	high	no	fair	yes
student	no	age > 30	credit = excellent	
age	income	student	credit_rating	buys_computer
>30	low	no	excellent	yes
>30	low	no	excellent	no

ແບບທີ 1 INCOME ເປັນ DECISION NODE



ແບບທີ 2 CREDIT_RATING ເລື່ອ DECISION NODE



CONCLUSION

Max_leaf_nodes เมื่อต้องการดูว่าต้นไม้จะโตไปทางไหน ให้ดูที่ค่า Gain ของแต่ละ features หาก Gain ของ features ใด มีค่ามาก การตัดสินใจที่จะโตของต้นไม้จะเลือกโตไปใน features นั้น ก็จะนี้ในการโตของต้นไม้จะต้องเกียบไปแต่ละ features เพื่อเลือกทิศทางในการโต และในการโตของต้นไม้นั้นจะโตได้ไม่เกินที่กำหนด max leaf nodes ไว้

ALGORITHM

```
function BFTree ( $A$ : a set of attributes,  
                 $E$ : the training instances,  
                 $N$ : the number of expansions,  
                 $M$ : the minimal number of instances at a terminal node  
    ) return a decision tree  
    begin  
        If  $E$  is empty, return failure;  
        Calculate the reduction of impurity for each attribute  
                in  $A$  on  $E$  at the root node  $RN$ ;  
        Find the best attribute  $A_b$  in  $A$ ;  
        Initialise an empty list  $NL$  to store nodes;  
        Add  $RN$  (with  $E$  and  $A_b$ ) into  $NL$ ;  
        expandTree( $NL$ ,  $N$ ,  $M$ );  
        return a tree with the root  $RN$ ;  
    end
```

```
expandTree( $NL$ ,  $N$ ,  $M$ )
begin
    If  $NL$  is empty, return;
    Get the first node  $FN$  from  $NL$ ;
    Retrieve training instances  $E$  and the best splitting attribute  $A_b$  of  $FN$ ;
    If  $E$  is empty, return failure;
    If the reduction of impurity of  $FN$  is 0 or  $N$  is reached,
        Make all nodes in  $NL$  into terminal nodes;
        return;
    If the split of  $FN$  on  $A_b$  would result in a successor node
        with less than  $M$  instances,
        Make  $FN$  into the terminal node;
        Remove  $FN$  from  $NL$ ;
        expandTree( $NL$ ,  $N$ ,  $M$ );
    Let  $SN_1$  and  $SN_2$  be the successor nodes generated by
        splitting  $FN$  on  $A_b$  on  $E$ ;
    Increment the number of expansions by one;
    Let  $E_1$  and  $E_2$  be the subsets of instances corresponding to
         $SN_1$  and  $SN_2$ ;
    Find the corresponding best attributes  $A_{b_1}$  for  $SN_1$ ;
    Find the corresponding best attributes  $A_{b_2}$  for  $SN_2$ ;
    Put  $SN_1$  (with  $E_1$  and  $A_{b_1}$ ) and  $SN_2$  (with  $E_2$  and  $A_{b_2}$ )
        into  $NL$  according to the reduction of impurity;
    Remove  $FN$  from  $NL$ ;
    expandTree( $NL$ ,  $N$ ,  $M$ );
end
```

THANK YOU

