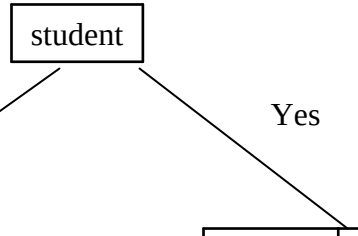
Decision Tree Examples

Professor Anita Wasilewska
Computer Science Department
Stony Brook University, NY

Training data

rec	Age	Income	Student	Credit_rating	Buys_computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<-=30	Medium	Yes	Excellent	Yes
r12	31…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

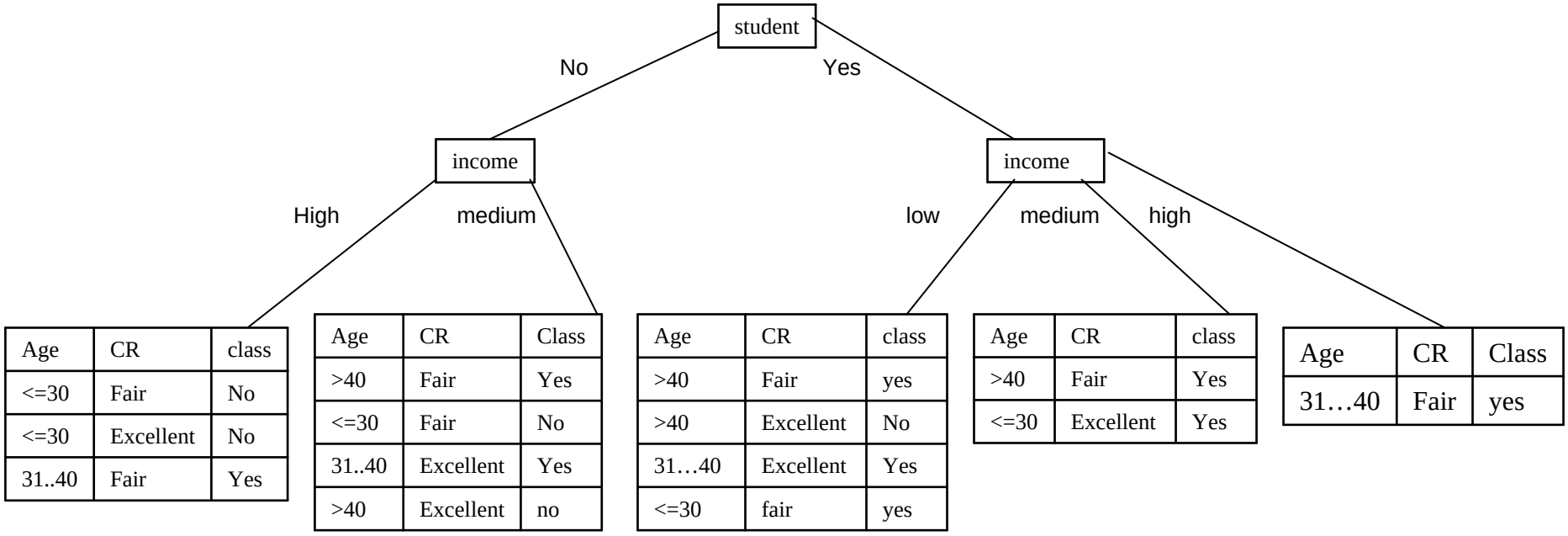
Decision Tree 1, Root: student



Age	income	CR	Class
<=30	High	Fair	No
<=30	High	Excellent	No
30…40	High	Fair	Yes
>40	Medium	Fair	Yes
<=30	Medium	Fair	No
31…40	Medium	Excellent	Yes
>40	Medium	Excellent	no

Age	income	CR	Class
>40	Low	Fair	Yes
>40	Low	excellent	No
31…40	Low	Excellent	Yes
<=30	Low	Fair	Yes
>40	Medium	Fair	Yes
<=30	Medium	Excellent	Yes
31…40	high	fair	yes

Decision Tree 1: income (L) and income (R)



Age	CR	class
<=30	Fair	No
<=30	Excellent	No
31..40	Fair	Yes

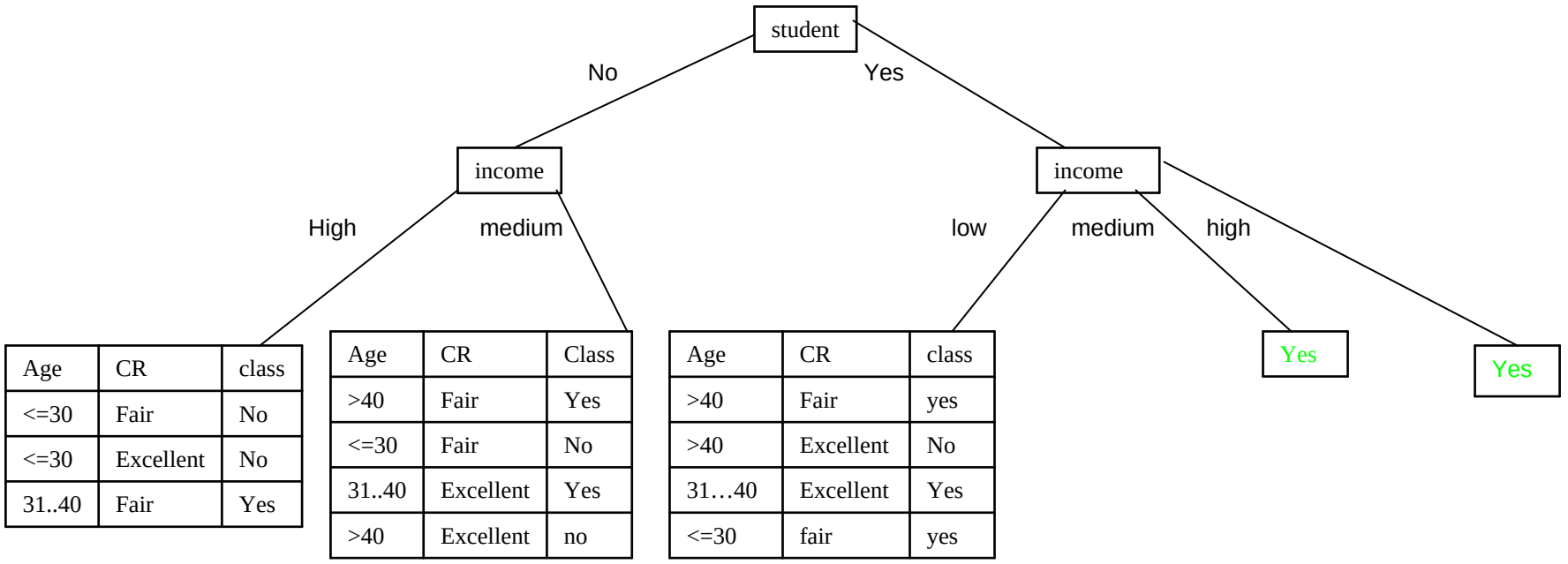
Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

Age	CR	class
>40	Fair	yes
>40	Excellent	No
31…40	Excellent	Yes
<=30	fair	yes

Age	CR	class
>40	Fair	Yes
<=30	Excellent	Yes

Age	CR	Class
31…40	Fair	yes

Decision Tree 1: next step

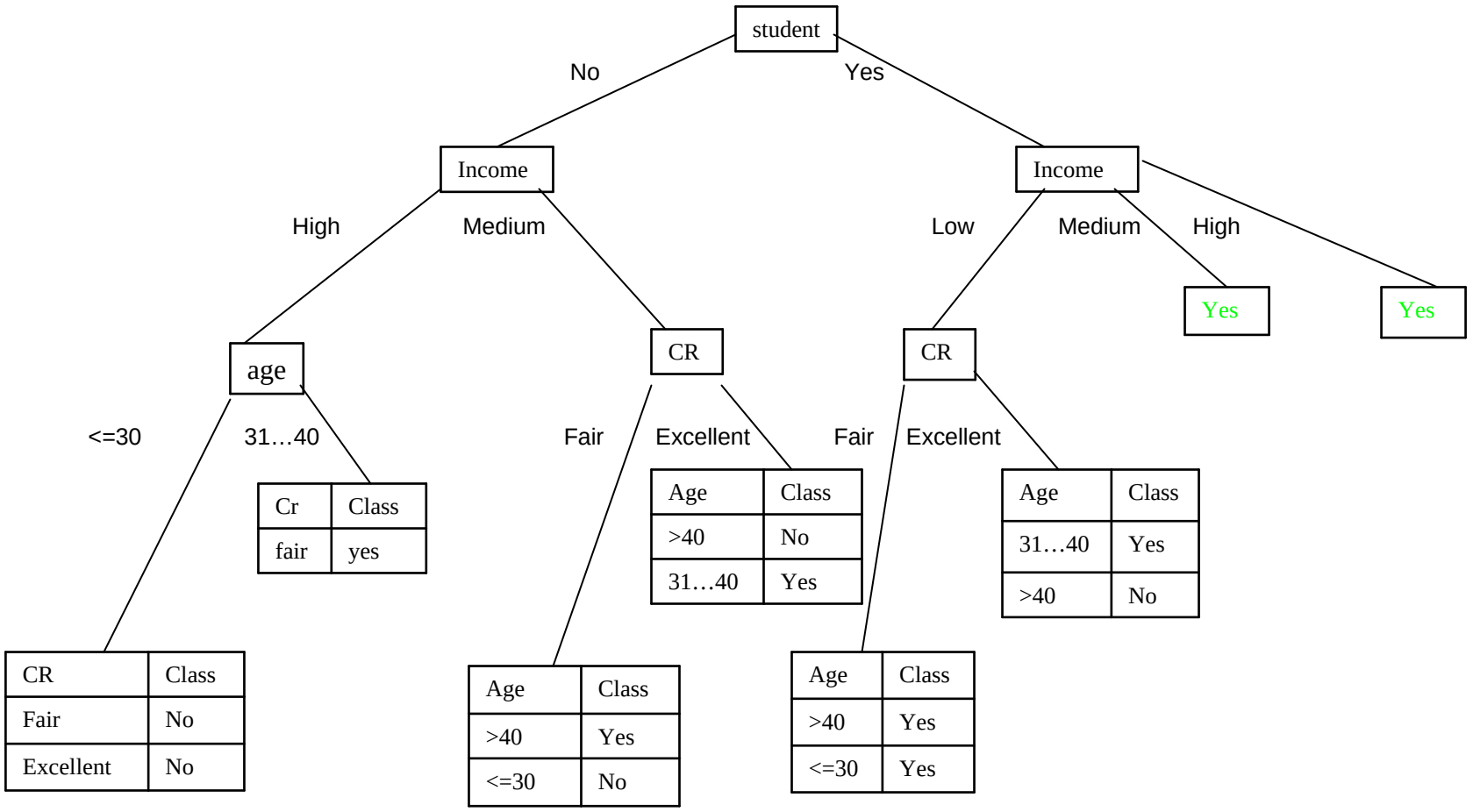


Age	CR	class
<=30	Fair	No
<=30	Excellent	No
31..40	Fair	Yes

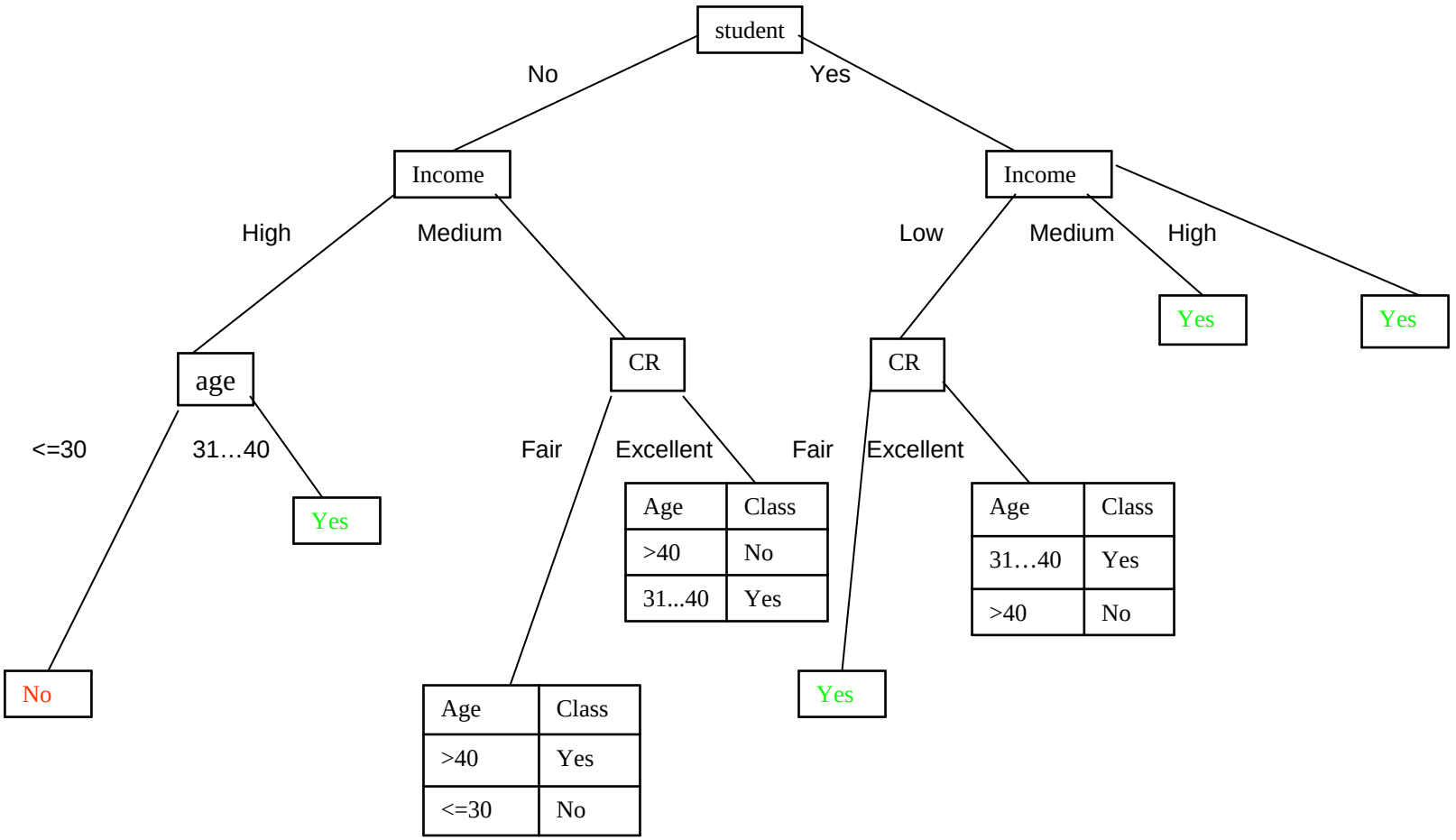
Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

Age	CR	class
>40	Fair	yes
>40	Excellent	No
31…40	Excellent	Yes
<=30	fair	yes

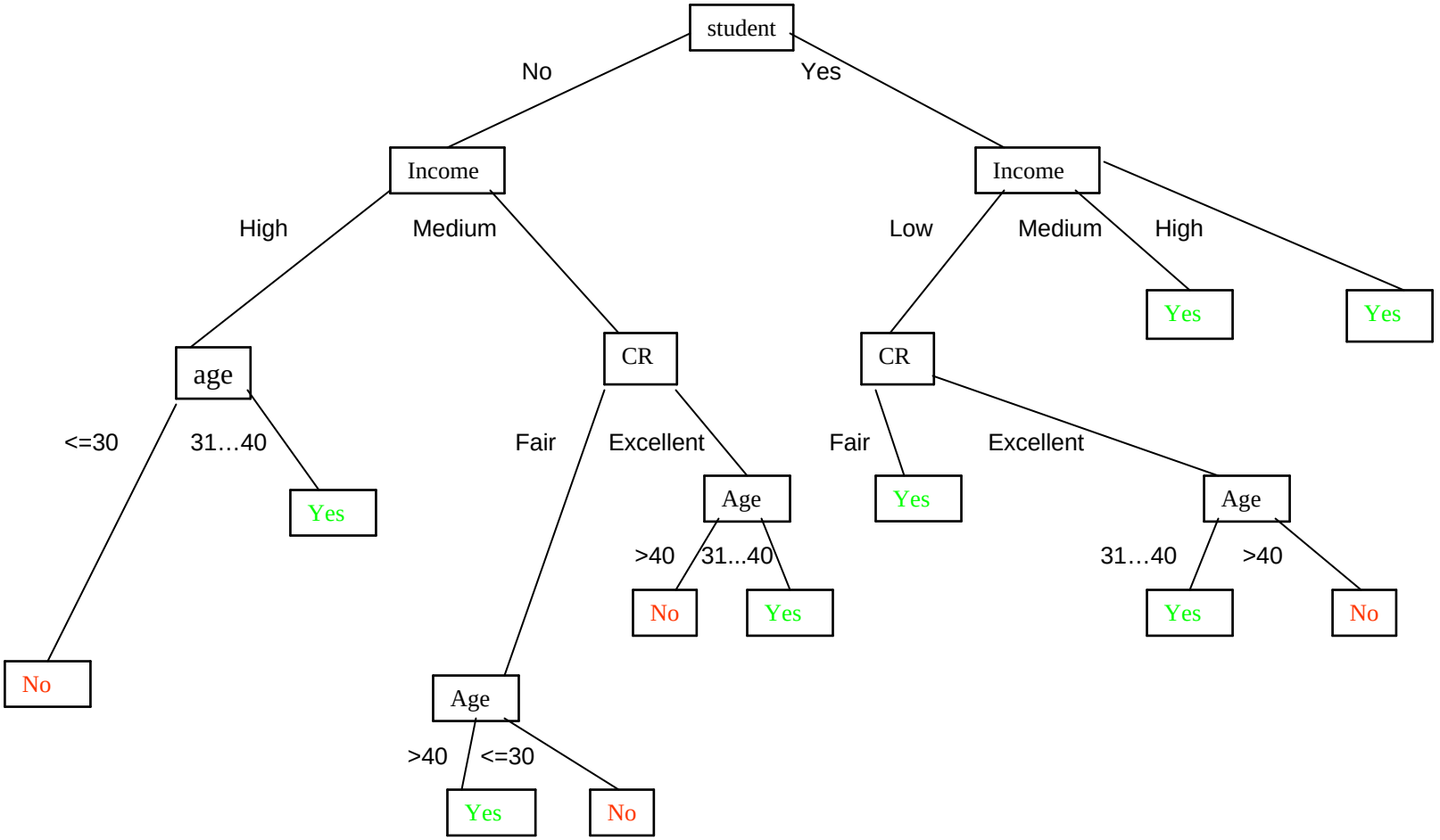
Decision Tree 1 : next step



Decision Tree 1: next step



Decision Tree 1 : last step

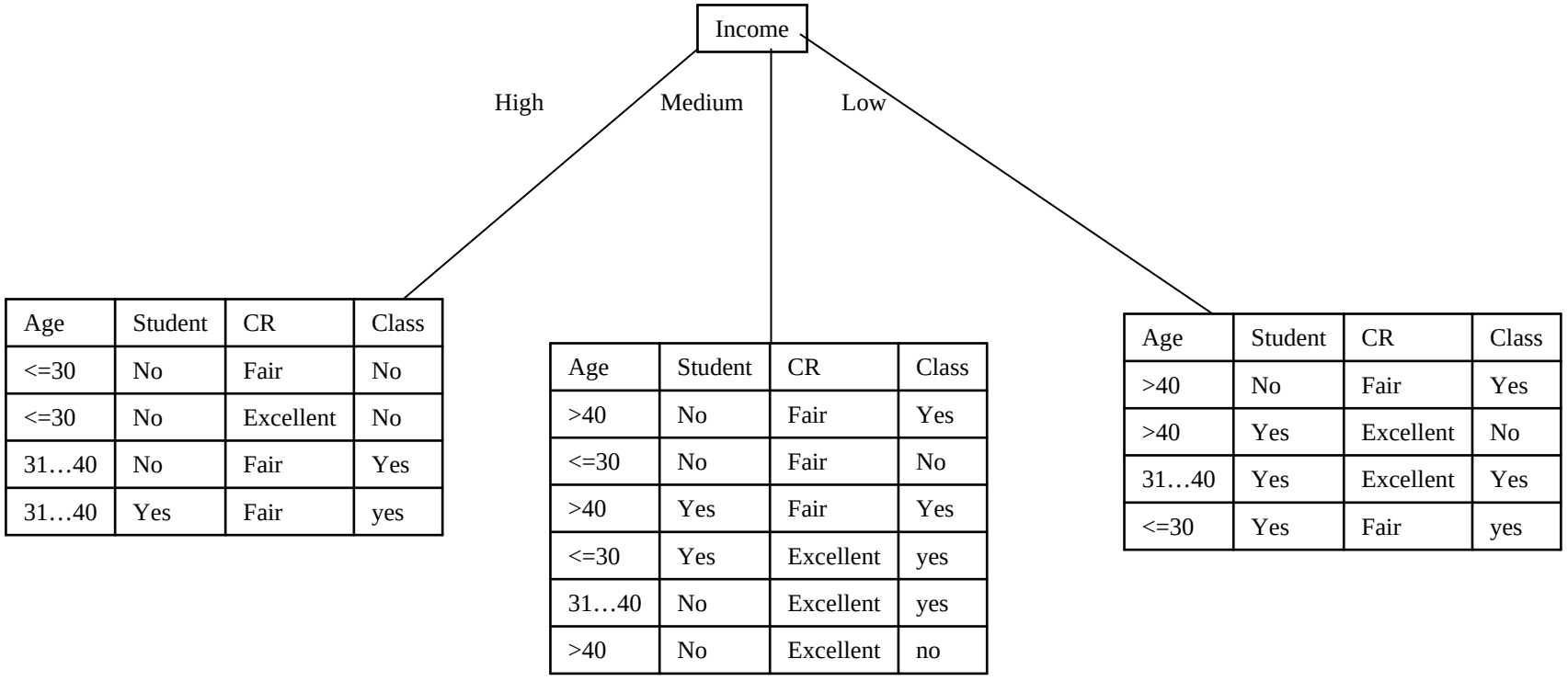


Tree 1 Classifcation rules

(Predicate form for testing)

- 1. student(x,no)^income(x,high)^age(x,<=30) => buys_computer(x,no)
- 2. student(x,no)^income(x,high)^age(x,31…40) => buys_computer(x,yes)
- 3. student(x,no)^income(medium)^CRx,fair)^age(x,>40) => buys_computer(x,yes)
- 4. student(x,no)^income(x,medium)^CR(x,fair)^age(x,b<=30) => buys_computer(x,no)
- 5. student(x,no)^income(x,medium)^CRx,(excellent)^age(x,>40) => buys_computer(x,no)
- 6. student(x,no)^income(x,medium)^CR(x,excellent)^age(x,31..40) => buys_computer(x,yes)
- 7. student(x,yes)^income(low)^CR(x,fair) => buys_computer(x,yes)
- 8. student(x,yes)^income(x,low)^CR(x,excellent)^age(x,b31..40) => buys_computer(x,yes)
- 9. student(x,yes)^income(x,low)^CR(x,excellent)^age(x,>40) => buys_computer(x,no)
- 10. student(x,yes)^income(x,medium)=> buys_computer(x,yes)
- 11. student(x,yes)^income(x,high)=> buys_computer(x,yes)

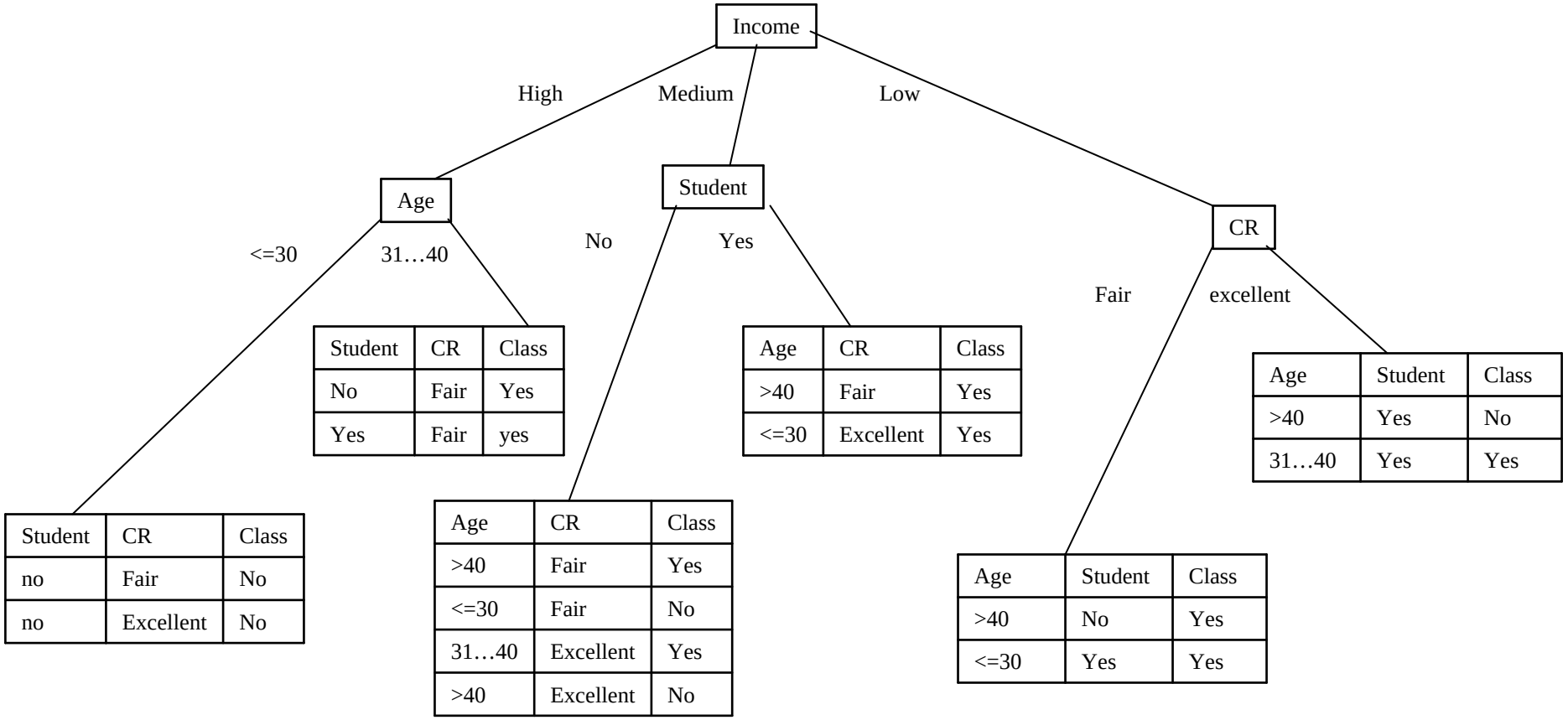
Decision Tree 2: Root Income



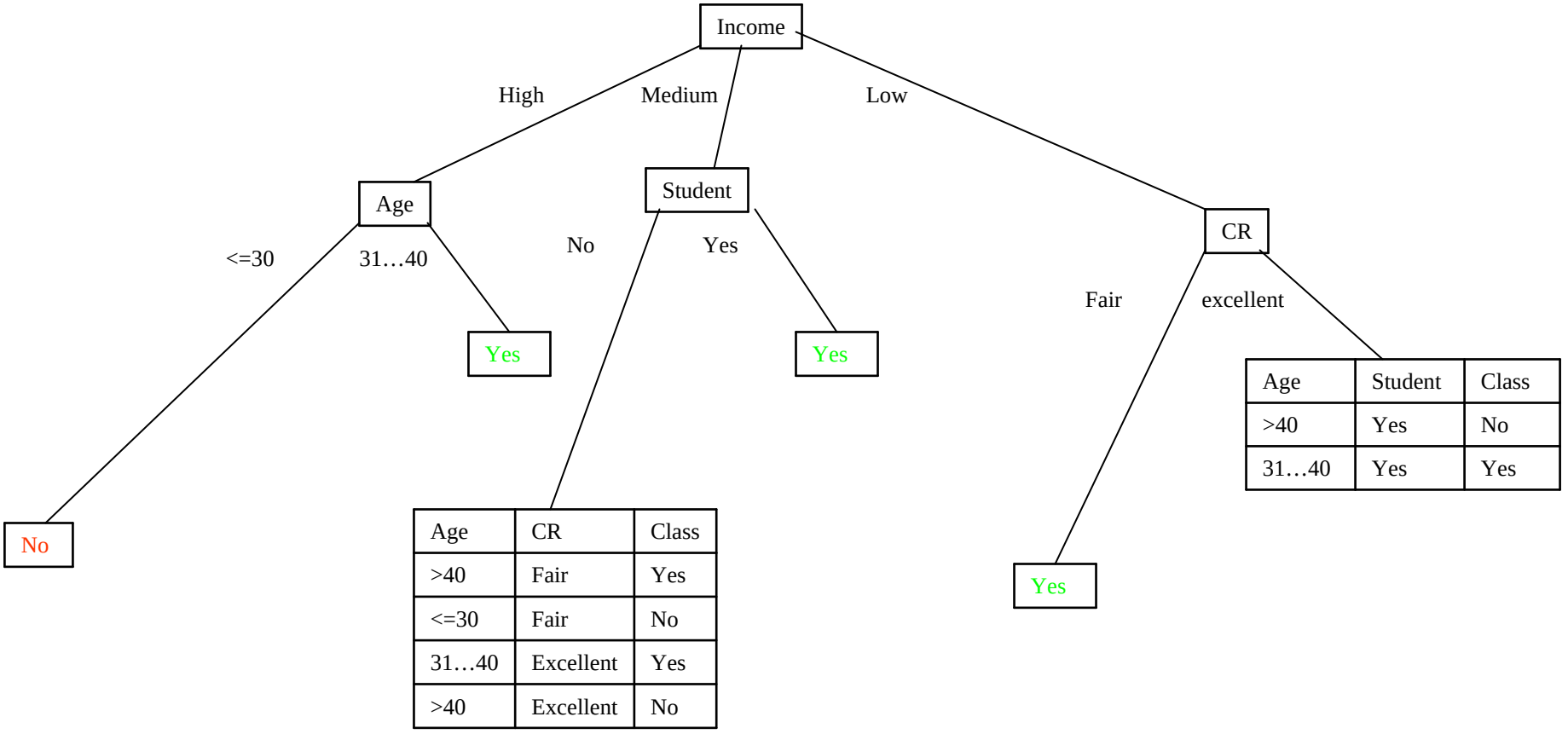
Age	Student	CR	Class
<=30	No	Fair	No
<=30	No	Excellent	No
31…40	No	Fair	Yes
31…40	Yes	Fair	yes

Age	Student	CR	Class
>40	No	Fair	Yes
<=30	No	Fair	No
>40	Yes	Fair	Yes
<=30	Yes	Excellent	yes
31…40	No	Excellent	yes
>40	No	Excellent	no

Age	Student	CR	Class
>40	No	Fair	Yes
>40	Yes	Excellent	No
31…40	Yes	Excellent	Yes
<=30	Yes	Fair	yes



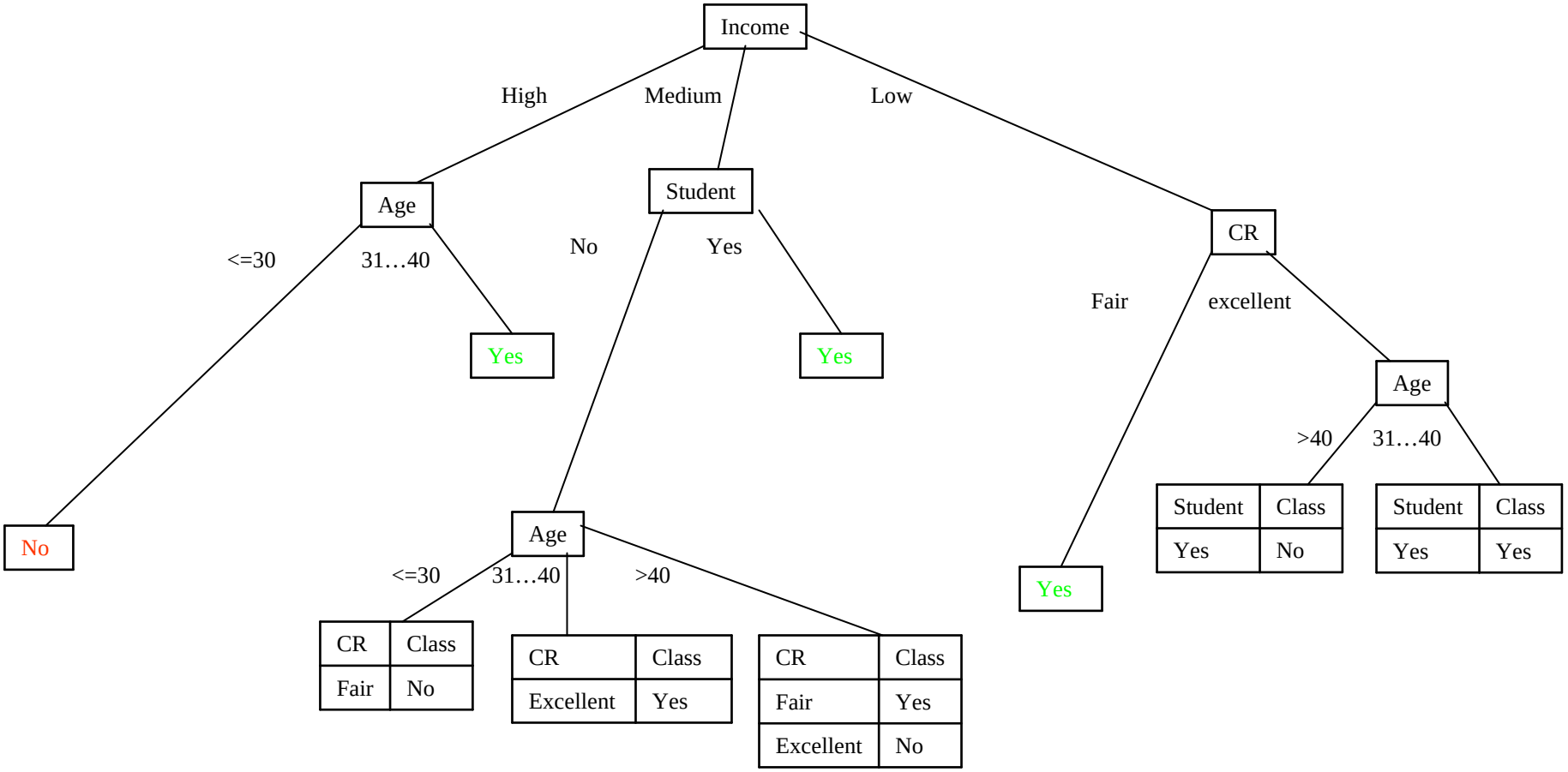
Decision Tree 2 : next step



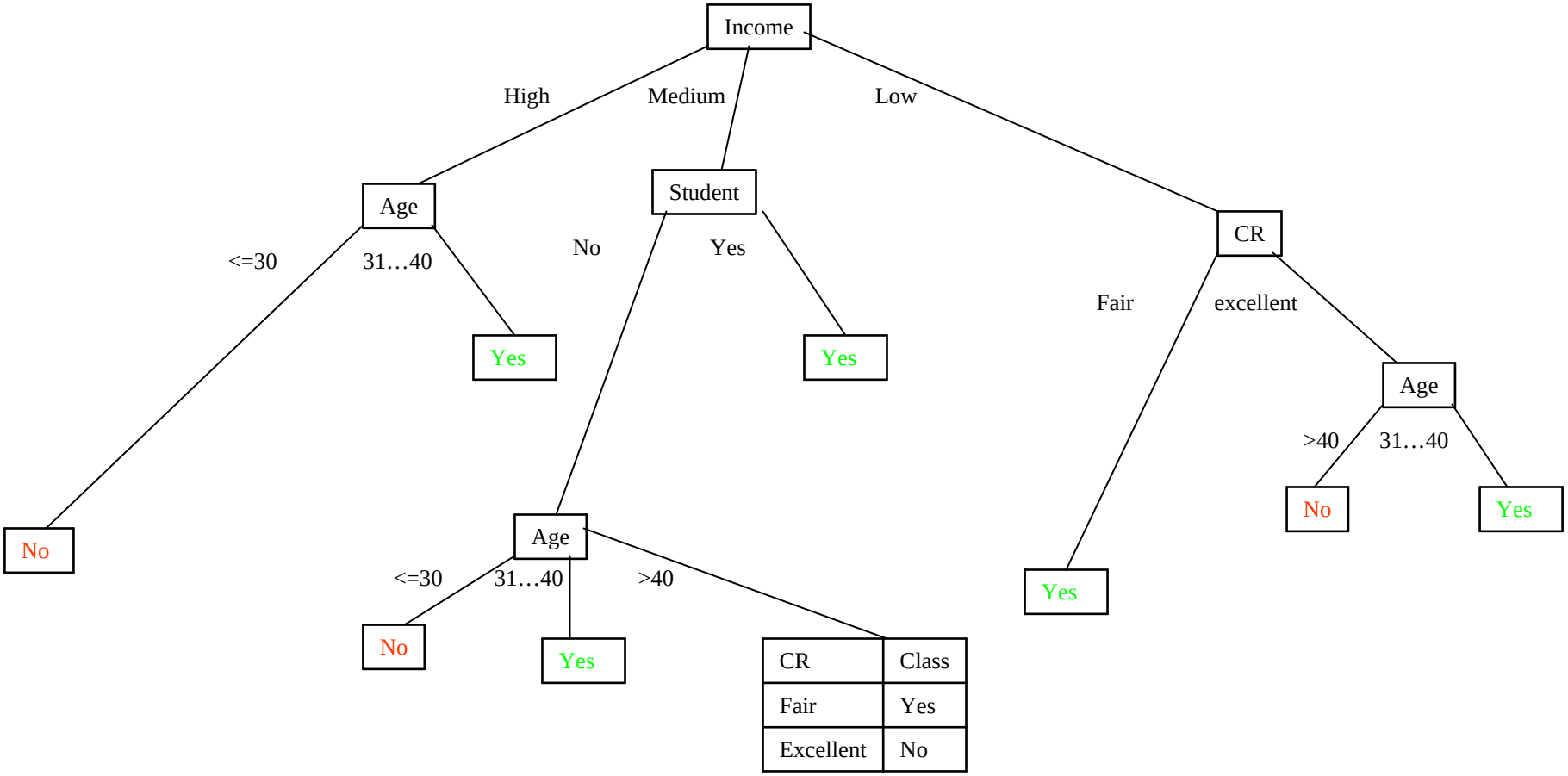
Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31…40	Excellent	Yes
>40	Excellent	No

Age	Student	Class
>40	Yes	No
31…40	Yes	Yes

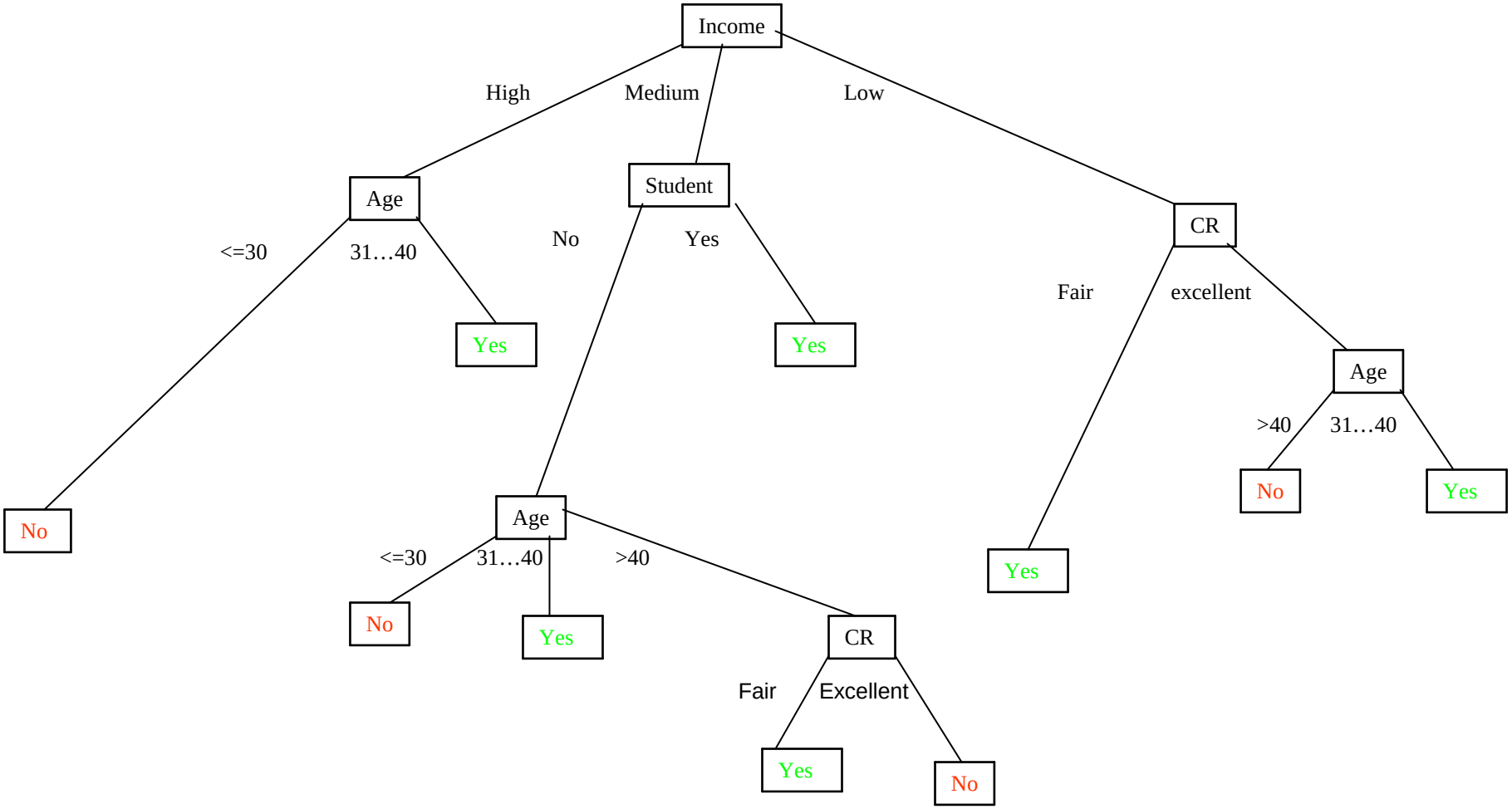
Decision Tree 2 : next step



Decision Tree 2 : next step



Decision Tree 2 : next step



Decision Tree 2 : last step

Tree 2 Classifcation rules

(Predicate form for testing)

- 1. income(x,high)^agex,(<=30) => buys_computer(x,no)
- 2. income(x,high)^age(x,31…40) => buys_computer(x,yes)
- 3. income(x,medium)^student(x,no)^age(x,<=30) => buys_computer(x,no)
- 4. income(x,medium)^student(x,no)^age(x,31…40) => buys_computer(x,yes)
- 5. income(x,medium)^student(x,no)^age(x,>40)^CR(x,fair) => buys_computer(x,yes)
- 6. income(x,medium)^student(x,no)^age(x,>40)^CR(x,excellent) => buys_computer(x,no)
- 7. income(x,medium)^student(x,yes)=> buys_computer(x,yes)
- 8. income(x,medium)^CR(x,fair)=> buys_computer(x,yes)
- 9. income(x,medium)^ CR(x,excellent)^age(x,>40)=> buys_computer(x,no)
- 10. income(x,medium)^ CR(x,excellent)^age(x,31…40)=> buys_computer(x,yes)

Formulas for information gain

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$E(A) = \sum_{i=1}^{v}\frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$Gain(A) = I(p,n) - E(A)$$

Calculations of **information gain** for Tree 1, Root: Student

- I(P,N) = - (9/(9+5))Logsub2*(9/(9+5))-(5/(9+5))logsub2*(5/(9+5))
  = -.643(-0.64)+(-.357)(-1.49) = .944
- I(Psub1,Nsub1) = -(6/(6+1)Logsub2*(6/(6+1)-(1/(6+1))logsub2*(1/(6+1))
  = -.857(-.22)+(-.143)(-2.81) = .591
- I(Psub2,Nsub2) = -(3/(3+4)Logsub2*(3/(3+4)-(4/(3+4))logsub2*(4/(3+4))
  = -.423(-1.24)+(-.571)(-0.81) = .987

Student	P	N	I(Psubi,Nsubi)
Yes	6	1	.591
No	3	4	.987

- E(Student) = (((6+1)/14) * .591) = .296 + ((3+4)/14) * .987 = .493
  = .789

Gain(Student) = .944 - .789 = .155

Calculations of **information gain** for Tree 1, Income(Left) node

- I(P,N) = -(3/(3+4)Logsub2*(3/(3+4)-(4/(3+4))logsub2*(4/(3+4))
  = -.423(-1.24)+(-.571)(-0.81) = .987
- I(Psub1,Nsub1) = -(1/(1+2)Logsub2*(1/(1+2)-(2/(1+2))logsub2*(2/(1+2))
  = -.333(-1.59)+(-.667)(-0.58) = .916
- I(Psub2,Nsub2) = -(2/(2+2)Logsub2*(2/(2+2)-(2/(2+2))logsub2*(2/(2+4))
  = -.5(-1)+(-.5)(-1) = 1

Income	P	N	I(Pi,Ni)
High	1	2	.916
Medium	2	2	1

- E(Income(L)) = (((1+2)/7) * .916) = .393 + ((2+2)/7) * 1 = .57
  = .963

Gain(Income(L)) = .987 - .963 = .024

Calculations of **information gain** for Tree 1, Income(Right) node

- I(P,N) = -(6/(6+1)Logsub2*(6/(6+1)-(1/(6+1))logsub2*(1/(6+1))
  = -.857(-.22)+(-2.81)(-.143) = .591
- I(Psub1,Nsub1) = -(3/(3+1)Logsub2*(3/(3+1)-(1/(3+1))logsub2*(1/(3+1))
  = -.75(-0.42)+(-.25)(-2) = .815
- I(Psub2,Nsub2) = -(2/(2+0)Logsub2*(2/(2+0)-(0/(2+0))logsub2*(0/(2+0))
  = -1(0)-(0)(infinity) = 0
- I(Psub3,Nsub3) = -(1/(1+0)Logsub2*(1/(1+0)-(0/(1+0))logsub2*(0/(1+0))
  = -1(0)-(0)(infinity) = 0

Income	P	N	I(Pi,Ni)
Low	3	1	.815
Medium	2	0	0
High	1	0	0

- E(Income(R)) = (((3+1)/7) * .815) = .465 + ((2+0)/7) * 0 = 0 + ((1+0)/7) * 0 = 0 = .465

Gain(Income(R) = .987 - .465 = .522

Calculations of **information gain** for Tree 1, age(1) node

- I(P,N) = -(1/(1+2)Logsub2*(1/(1+2)-(2/(1+2))logsub2*(2/(1+2))
  = -.333(-1.59)+(-.667)(-0.58) = .916

age	P	N	I(Psubi,Nsubi)
<=30	0	2	0
31…40	1	0	0

- E(Age(1)) = (((0+2)/3) * 0) = 0 + ((1+0)/3) * 0) = 0 = 0

Gain(Age(1)) = .916 – 0 = .916

Calculations of **information gain** for Tree 1, CR(Left) node

- I(P,N) = -(2/(2+2)Logsub2*(2/(2+2)-(2/(2+2))logsub2*(2/(2+2))
  = -.5(-1)+(-.5)(-1) = 1

CR	P	N	I(Psubi,Nsubi)
Fair	1	1	1
Excellent	1	1	1

- E(CR(R)) = (((1+1)/4) * 1) = .5 + ((1+1)/4) * 1) = .5 = 1

Gain(CR(R)) = 1 – 1 = 0

Calculations of **information gain** for Tree 1, CR(right) node

- I(P,N) = -(3/(3+1)Logsub2*(3/(3+1)-(1/(3+1))logsub2*(1/(3+1))
  = -.75(-0.42)+(-.25)(-2) = .815

CR	P	N	I(Psubi,Nsubi)
Fair	2	0	0
Excellent	1	1	1

- E(CR(R)) = (((2+0)/4) * 0) = 0 + ((1+1)/4) * 1) = .5 = ,5

Gain(CR(R)) = .815 – .5 = .315

Calculations of **information gain** for Tree 1, age(2) node

- I(P,N) = -(1/(1+1)Logsub2*(1/(1+1)-(1/(1+1))logsub2*(1/(1+1))
  = -.5(-1)+(-.5)(-1) = 1

Income	P	N	I(Psubi,Nsubi)
>40	1	0	0
<=30	0	1	0

- E(Age(2)) = (((1+0)/2) * 0) = 0 + ((0+1)/2) * 0) = 0 = 0

Gain(Age(2)) = 1 – 0 = 1

Calculations of **information gain** for Tree 1, age(3) node

- I(P,N) = -(1/(1+1)Logsub2*(1/(1+1)-(1/(1+1))logsub2*(1/(1+1))
  = -.5(-1)+(-.5)(-1) = 1

Income	P	N	I(Psubi,Nsubi)
>40	0	1	0
<=30	1	0	0

- E(Age(3)) = (((0+1)/2) * 0) = 0 + ((1+0)/2) * 0) = 0 = 0

Gain(Age(3)) = 1 – 0 = 1

Calculations of **information gain** for Tree 1, age(4) node

- I(P,N) = -(1/(1+1)Logsub2*(1/(1+1)-(1/(1+1))logsub2*(1/(1+1))
  = -.5(-1)+(-.5)(-1) = 1

age	P	N	I(Psubi,Nsubi)
31…40	1	0	0
>40	0	1	0

- E(Age(4)) = (((1+0)/2) * 0) = 0 + ((0+1)/2) * 0) = 0 = 0

Gain(Age(4)) = 1 – 0 = 1

Information gain measure

Gain(student) =.155

Gain(income(L)) = .024

Gain(income(R)) = .522

Gain(age(1)) = .916

Gain(CR(L)) = 0

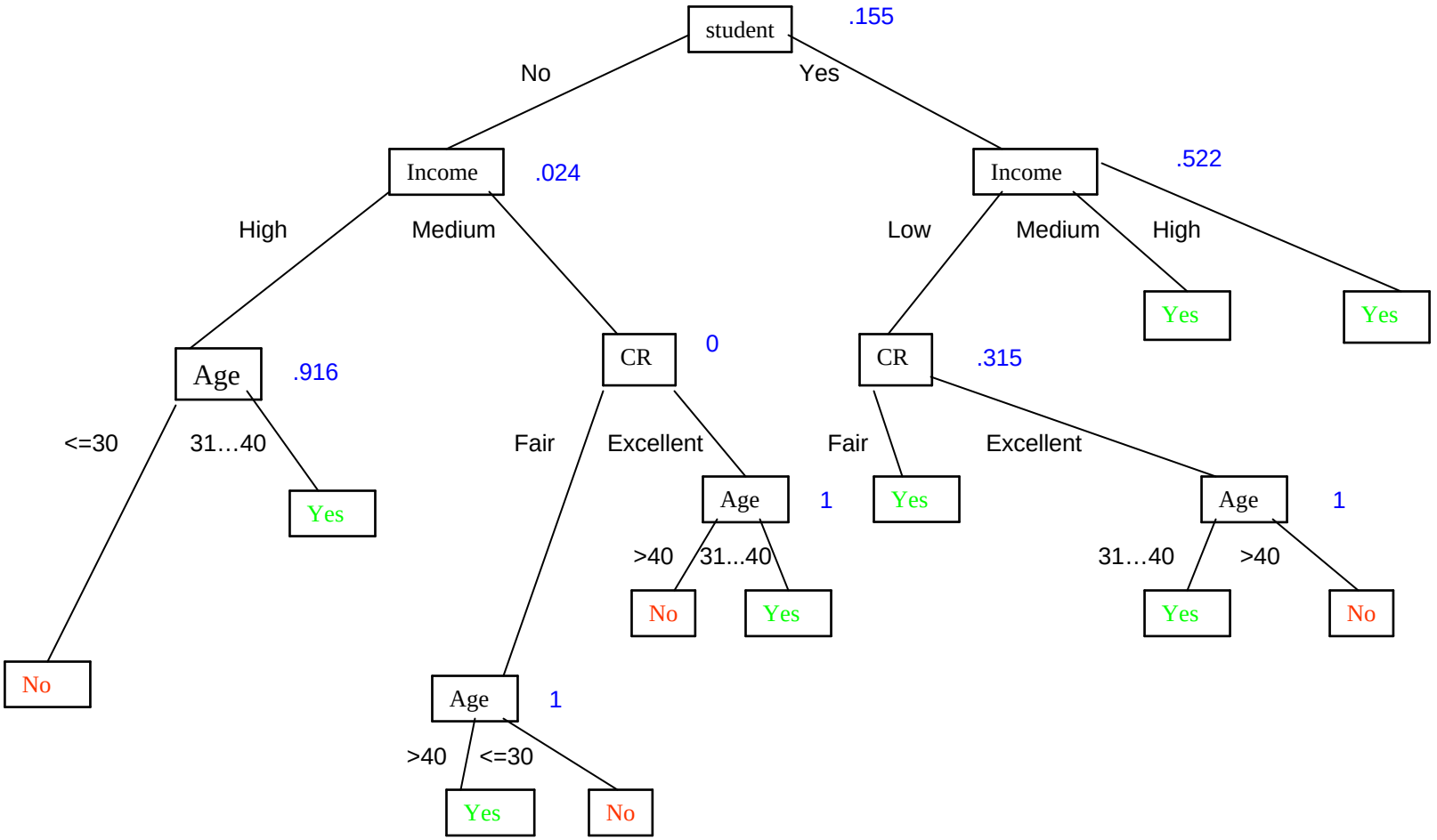
Gain(CR(R)) = .315

Gain(age(2)) = 1

Gain(age(3)) = 1

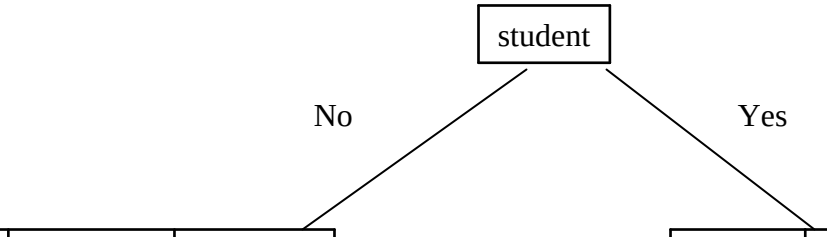
Gain(age(4)) = 1

Information Gain for each node



Tree 3: root student plus majority voting at any node

Heuristics: use majority voting at any chosen NODE of the tree

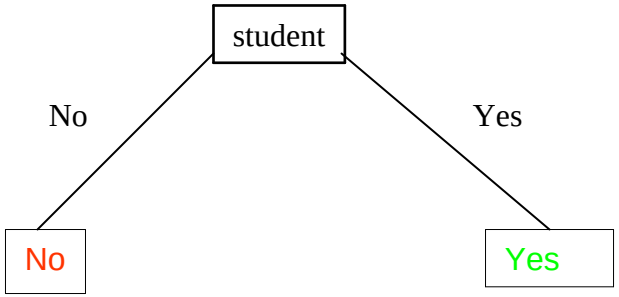


Age	income	CR	Class
<=30	High	Fair	No
<=30	High	Excellent	No
30…40	High	Fair	Yes
>40	Medium	Fair	Yes
<=30	Medium	Fair	No
31…40	Medium	Excellent	Yes
>40	Medium	Excellent	no

Age	income	CR	Class
>40	Low	Fair	Yes
>40	Low	excellent	No
31…40	Low	Excellent	Yes
<=30	Low	Fair	Yes
>40	Medium	Fair	Yes
<=30	Medium	Excellent	Yes
31…40	high	fair	yes

Tree 3: Student root plus majority voting

Heuristics: use majority voting at any chosen NODE of the tree



Tree 3 (majority voting) rules and their accuracy

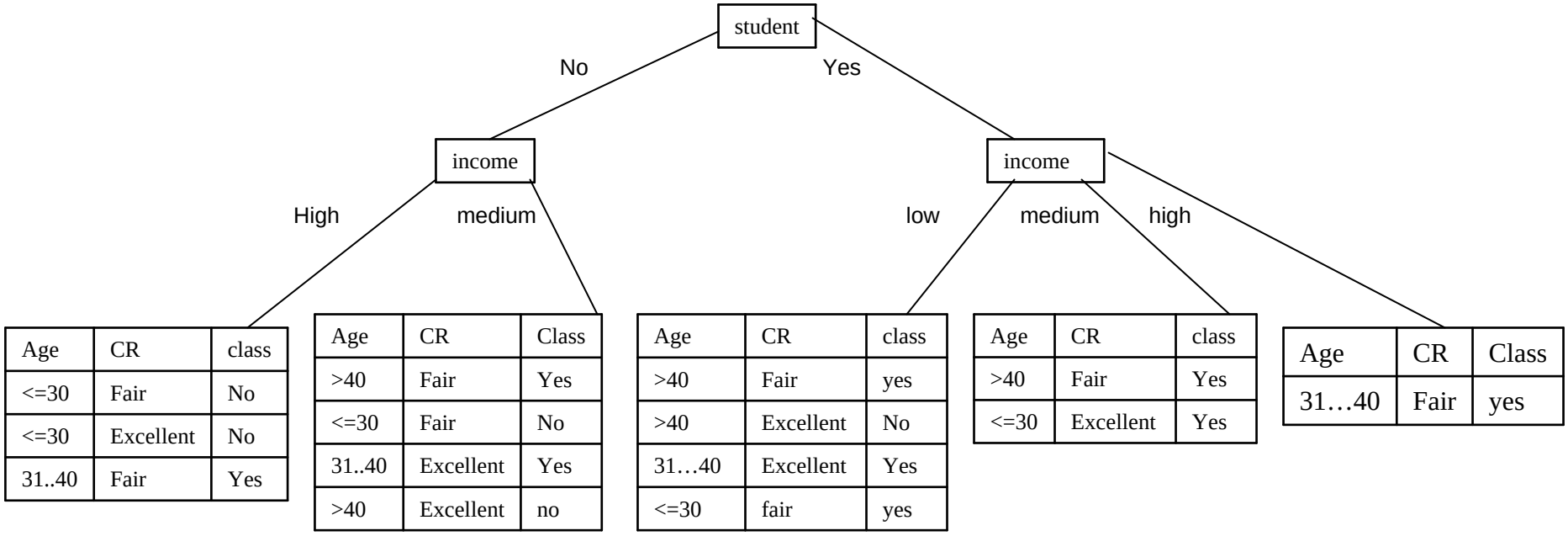
(Predicate form for testing)

- **RULES:**
- Student(x,no) => buys_computer(x,no)
- Student(x,yes) => buys_ computer(x,yes)

- Since 10 out of 14 records match the rules
- **Rules accuracy:** 10/14 = 0.714 = 71.4%
- **Error Rate:** 4/14= 0.28.6 = 28.6%

Tree 4: root student with majority voting on branch income-high

Heuristics: use majority voting at any chosen NODE of the tree



Age	CR	class
<=30	Fair	No
<=30	Excellent	No
31..40	Fair	Yes

Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

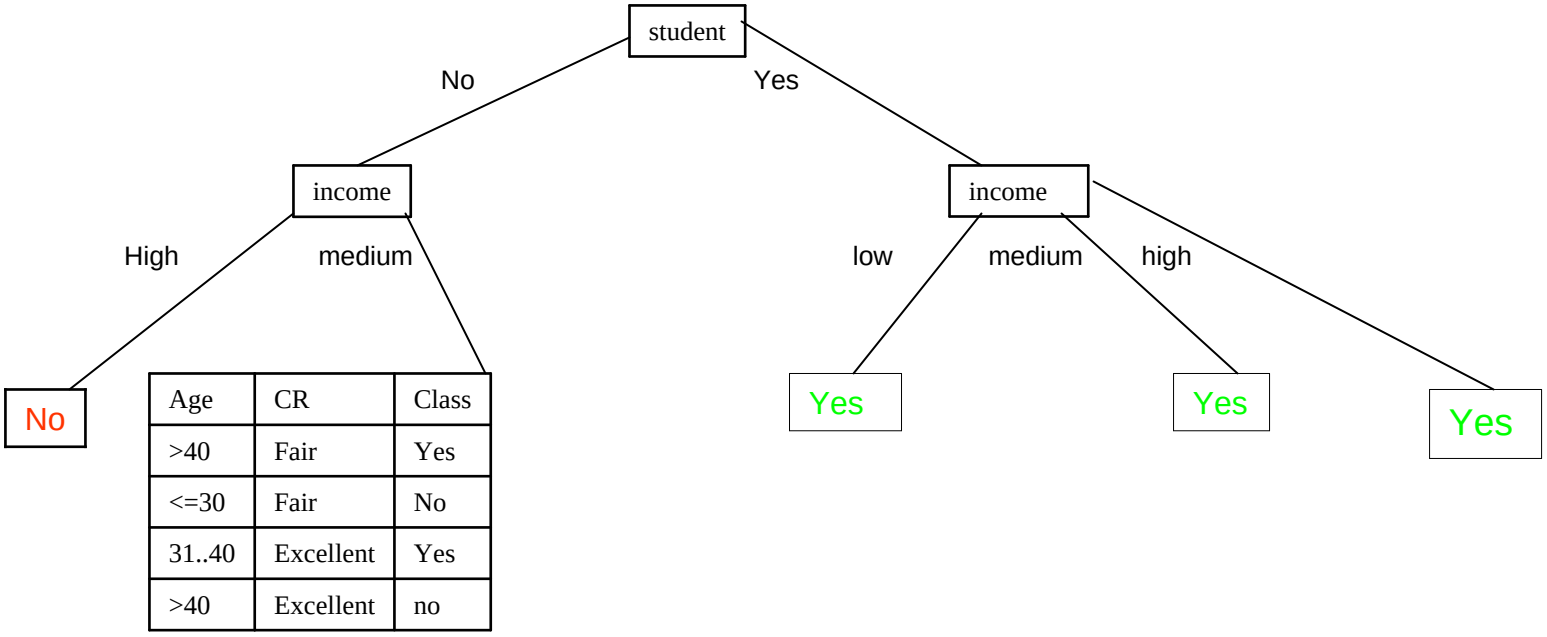
Age	CR	class
>40	Fair	yes
>40	Excellent	No
31…40	Excellent	Yes
<=30	fair	yes

Age	CR	class
>40	Fair	Yes
<=30	Excellent	Yes

Age	CR	Class
31…40	Fair	yes

Tree 4: root student with majority voting

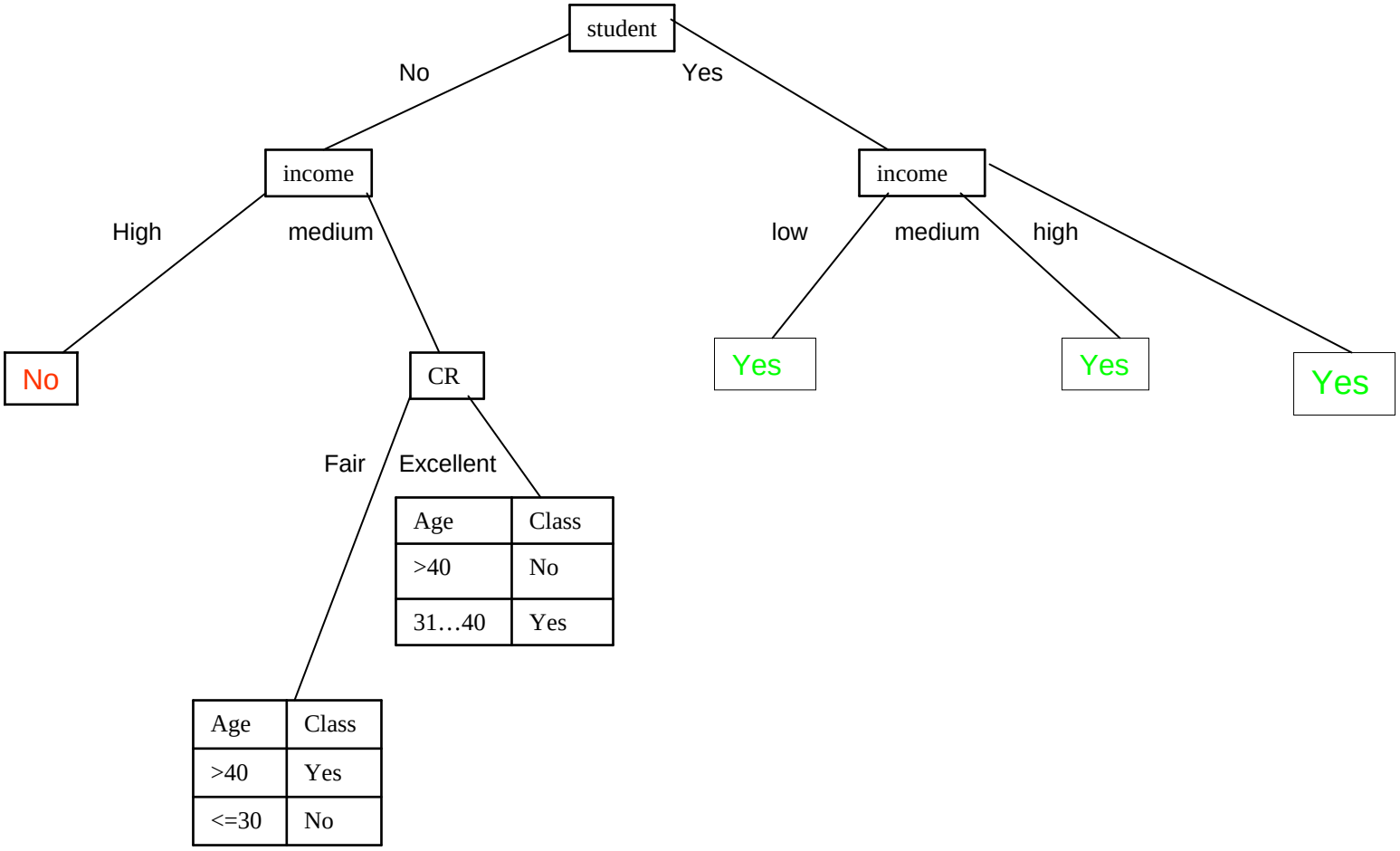
Heuristics: use majority voting at any chosen NODE of the tree



Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

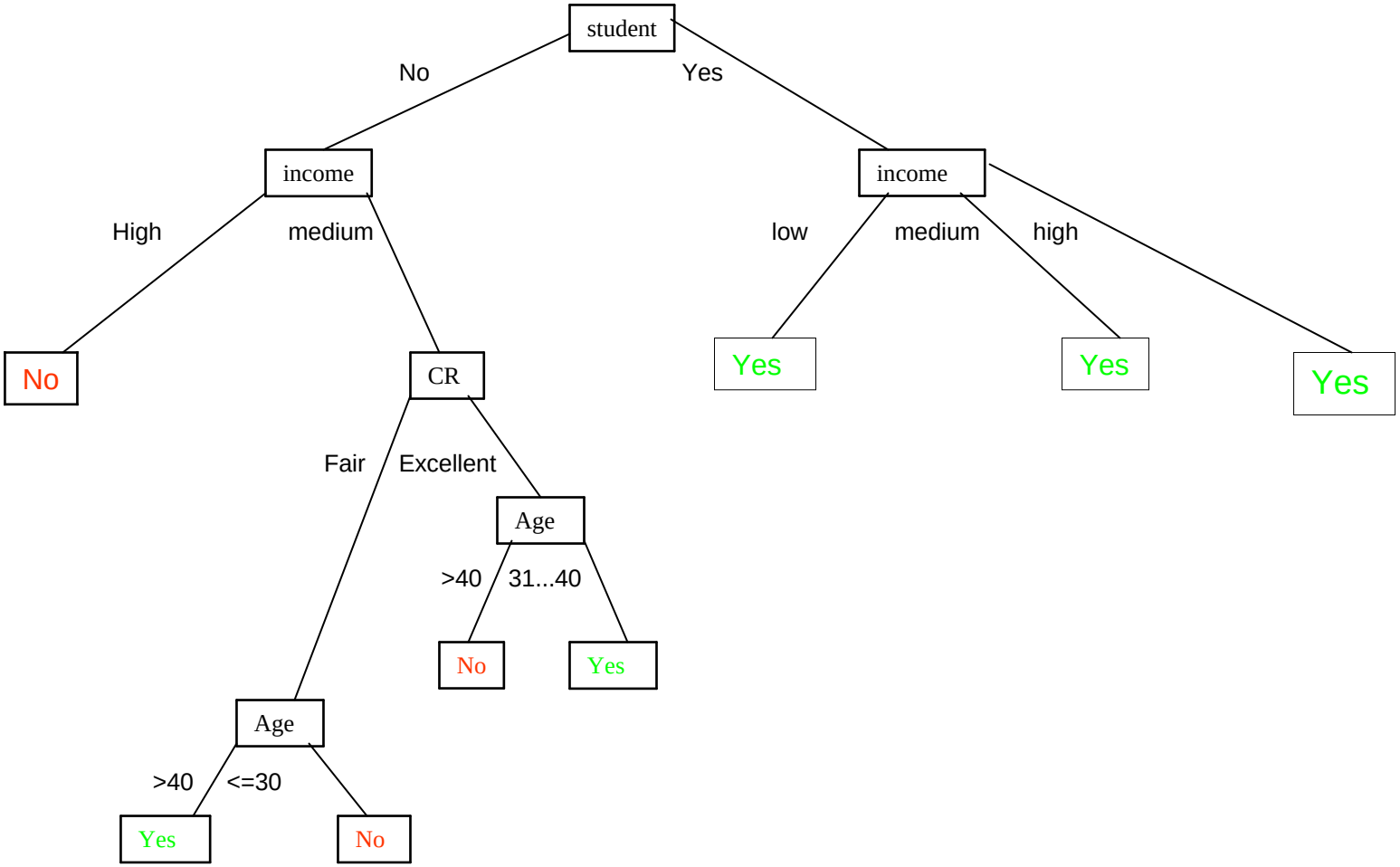
Tree 4: root student with majority voting

Heuristics: use majority voting at any chosen NODE of the tree



Tree 4: root student with majority voting

Heuristics: use majority voting at any chosen NODE of the tree



Tree 4: Classifcation rules and their accuracy

(Predicate form for testing)

- Student(x,no)^income(x,high) => buys_computer(x,no)
- Student(x,no)^income(x,medium)^CR(x,fair)^age(x,>40) => buys_computer(x,yes)
- Student(x,no)^income(x,medium)^CR(x,fair)^age(x,<=30) => buys_computer(x,no)
- Student(x,no)^income(x,medium)^CR(x,excellent)^age(x,>40) => buys_computer(x,no)
- Student(x,no)^income(x,medium)^CR(x,fair)^age(x,31…40) => buys_computer(x,yes)
- Student(x,yes)^income(x,low) => buys_computer(x,yes)
- Student(x,yes)^income(x,medium) => buys_computer(x,yes)
- Student(x,yes)^income(x,high) => buys_computer(x,yes)

- Since 11 out of 14 records match the rules **the accuracy of the rules** = 11/14 = 0.786 = 78.6%

Training data plus (red) test data

REC	Age	Income	Student	Credit_rating	Buys_computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<-=30	Medium	Yes	Excellent	Yes
r12	31…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No
r15	<=30	Medium	No	Excellent	No
r16	<=30	Low	No	Fair	No
r17	<=30	Low	No	Excellent	No
r18	31…40	Low	Yes	Fair	Yes
r19	>40	Medium	Yes	Excellent	Yes
r20	31…40	High	No	Excellent	Yes

Tree1 classifcation rules

(Predicate form for testing)

- 1. student(x,no)^income(x,high)^agex,<=30) => buys_computer(x,no)
- 2. student(x,no)^income(x,high)^age(x,31…40) => buys_computer(x,yes)
- 3. student(x,no)^income(x,medium)^CR(x,fair)^age(x,>40) => buys_computer(x,yes)
- 4. student(x,no)^income(x,medium)^CR(x,fair)^agex,(<=30) => buys_computer(x,no)
- 5. student(x,no)^income(x,medium)^CR(x,excellent)^age(x,>40) => buys_computer(x,no)
- 6. student(x,no)^income(x,medium)^CR(x,excellent)^age(x,31..40) => buys_computer(x,yes)
- 7. student(x,yes)^income(x,low)^CR(x,fair) => buys_computer(x,yes)
- 8. student(x,yes)^income(x,low)^CR(x,excellent)^age(x,31..40) => buys_computer(x,yes)
- 9. student(x,yes)^income(x,low)^CR(x,excellent)^age(x,>40) => buys_computer(x,no)
- 10. student(x,yes)^income(x,medium)=> buys_computer(x,yes)
- 11. student(x,yes)^income(x,high)=> buys_computer(x,yes)

Book classification rules

- Age(x,<=30)^student(x,no)=> buys_computer(x,no)
- Age(x,<=30)^student(x,yes)=> buys_computer(x,yes)
- Age(x,31…40)=> buys_computer(x,yes)
- Age(x,>40)^credit_rating(x,excellent)=> buys_computer(x,no)
- Age(x,<=30)^credit_rating(x,fair)=> buys_computer(x,yes)

Book Rules accuracy

- For book rules 14 of 14 records from the training data match, so the **accuracy of the rules** is 14/14= 100% accurate

Tree 2 Classifcation rules

(Predicate form for testing)

- 1. income(x,high)^age(x,<=30) => buys_computer(x,no)
- 2. income(x,high)^age(x,31…40) => buys_computer(x,yes)
- 3. income(x,medium)^student(x,no)^age(x,b<=30) => buys_computer(x,no)
- 4. income(x,medium)^student(x,no)^age(x,31…40) => buys_computer(x,yes)
- 5. income(x,medium)^student(x,no)^age(x,>40)^CR(x,fair) => buys_computer(x,yes)
- 6. income(x,medium)^student(x,no)^age(x,>40)^CR(x,excellent) => buys_computer(x,no)
- 7. income(x,medium)^student(x,yes)=> buys_computer(x,yes)
- 8. income(x,medium)^CR(x,fair)=> buys_computer(x,yes)
- 9. income(x,medium)^ CR(x,excellent)^age(x,>40)=> buys_computer(x,no)
- 10. income(x,medium)^ CR(x,excellent)^age(x,31…40)=> buys_computer(x,yes)

Predictive accuracy for (red) test data

- For the book rules: 5 of 6 records from the red test data match, so the predictive accuracy of the book rules is 5/6= 83.33%

- For the tree 2 rules: 2 out of 6 records from the test data match, so

  the predictive accuracy of the rules 2 is 2/6 = 33.33%

Tree 3 (majority voting) rules and their accuracy

- **RULES:**
- Student(x,no) => buys_computer(x,no)
- Student(x,yes) => buys_ computer(x,yes)

- **Rules accuracy:** 10/14 = 0.714 = 71.4%
- **Predictive accuracy** (with respect to red test data):
- 5/6= 83.33%

- **Observe that the 100% accurate book rules had also predictive accuracy 83.33%**