DataInquest

Big Data Technologies and Data Science training for all!!

Data Science

Topics for Today

- ✓ Association Rule Mining
- ✓ Supervised Learning
- ✓ Decision Tree Classifier
    - What are Decision Trees
    - Decision Tree Examples
    - How to build Decision trees
- ✓ Application of Technique on smaller datasets for better understanding using R software

Association Rule Mining

Association Rule Mining

- In data mining, **association rule learning** is a popular and well researched method for discovering interesting relations between variables in large databases.

- It is intended to identify strong rules discovered in databases using different measures of interests.

- The rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat.

- Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

Sample Input Data

transaction_id	items
1	citrus fruit
1	semi-finished bread
1	margarine
1	ready soups
2	tropical fruit
2	yogurt
2	coffee
3	whole milk
4	pip fruit
4	yogurt
4	cream cheese
4	meat spreads
5	other vegetables
5	whole milk

Association Rule Mining

	lhs	rhs	support	confidence	lift
89	Hard cheese	Whole milk	0.01006609	0.41078838	1.6076815
90	Whole milk	Hard cheese	0.01006609	0.03939515	1.6076815
91	Butter milk	Other vegetables	0.01037112	0.37090909	1.9169159
92	Other vegetables	Butter milk	0.01037112	0.05359958	1.9169159
93	Butter milk	Whole milk	0.01159126	0.41454545	1.6223854
94	Whole milk	Butter milk	0.01159126	0.04536411	1.6223854
95	ham	Whole milk	0.01148958	0.44140625	1.7275091
96	Whole milk	ham	0.01148958	0.04496618	1.7275091
97	Sliced cheese	Whole milk	0.01077783	0.43983402	1.7213560
98	Whole milk	Sliced cheese	0.01077783	0.04218066	1.7213560
99	oil	Whole milk	0.01128622	0.40217391	1.5739675
100	Whole milk	Oil	0.01128622	0.04417031	1.5739675
101	onions	Other vegetables	0.01423488	0.45901639	2.3722681
102	Other vegetables	Onions	0.01423488	0.07356805	2.3722681
103	onions	Whole milk	0.01209964	0.39016393	1.5269647
104	Whole milk	Onions	0.01209964	0.04735376	1.5269647
105	berries	yogurt	0.01057448	0.31804281	2.2798477

Association Rule Mining - Concepts

Constraints on below measures are used to select useful and best rules of all the rules given by R
After analyzing these values for all the rules, best rules for WB have been obtained.

Support

- The support Supp(X)=proportion of transactions in the data set which contain the interest.

Confidence

- The confidence of a rule: Conf(x=>y)= Supp(X U Y)/Supp(X)

Lift

- The lift of a rule: Lift(X=>Y)= Supp(X U Y)

  (Supp(X) X Supp(Y))

E.g.:- Consider rule: {Jack the Ripper (1988)} => {Strawberry Blonde}
Let Jack the Ripper =X and Strawberry Blonde =Y, Then

Support(X U Y)= No of transactions involving both Jack the Ripper and Strawberry Blonde/ Total no of transactions

Confidence= No of transactions where Strawberry Blonde was also bought when Jack the Ripper was bought/ No of transactions where Jack the Ripper was bought

Lift = Ratio of observed support to the expected support

Association Rule Mining - Concepts

Association rule generation is usually split up into two separate steps:

Step #1:

Minimum support is applied to find all frequent itemsets in a database.

Step #2:

These frequent itemsets and the minimum confidence constraint are used to form rules.

Association Rule Mining-Single Cardinality

S No.	Rules	Support	Confidence	Lift
1	{Strawberry Blonde} => {Canterville Ghost}	6.91%	35.91%	1.838296285
2	{Canterville Ghost} => {Strawberry Blonde}	6.91%	35.38%	1.838296285
3	{Doc Savage: The Man of Bronze} => {Green Slime}	8.28%	38.98%	1.791373861
4	{Green Slime} => {Doc Savage: The Man of Bronze}	8.28%	38.06%	1.791373861
5	{Green Slime} => {She}	8.22%	37.80%	1.769506084
6	{She} => {Green Slime}	8.22%	38.50%	1.769506084
7	{Jack the Ripper (1988)} => {She}	5.94%	35.14%	1.644963145
8	{She} => {Jack the Ripper (1988)}	5.94%	27.81%	1.644963145
9	{Pretty Maids All In A Row} => {Dark of the Sun}	7.37%	34.22%	1.580866863
10	{Dark of the Sun} => {Pretty Maids All In A Row}	7.37%	34.04%	1.580866863
11	{Doc Savage: The Man of Bronze} => {She}	6.97%	32.80%	1.535434995
12	{She} => {Doc Savage: The Man of Bronze}	6.97%	32.62%	1.535434995
13	{Pretty Maids All In A Row} => {Green Slime}	6.85%	31.83%	1.462854278
14	{Green Slime} => {Pretty Maids All In A Row}	6.85%	31.50%	1.462854278
15	{Pretty Maids All In A Row} => {She}	6.62%	30.77%	1.440559441

Sample Interpretation for Rule 1: Those customers buying Strawberry Blonde are usually more prone to also buy Canterville Ghost.

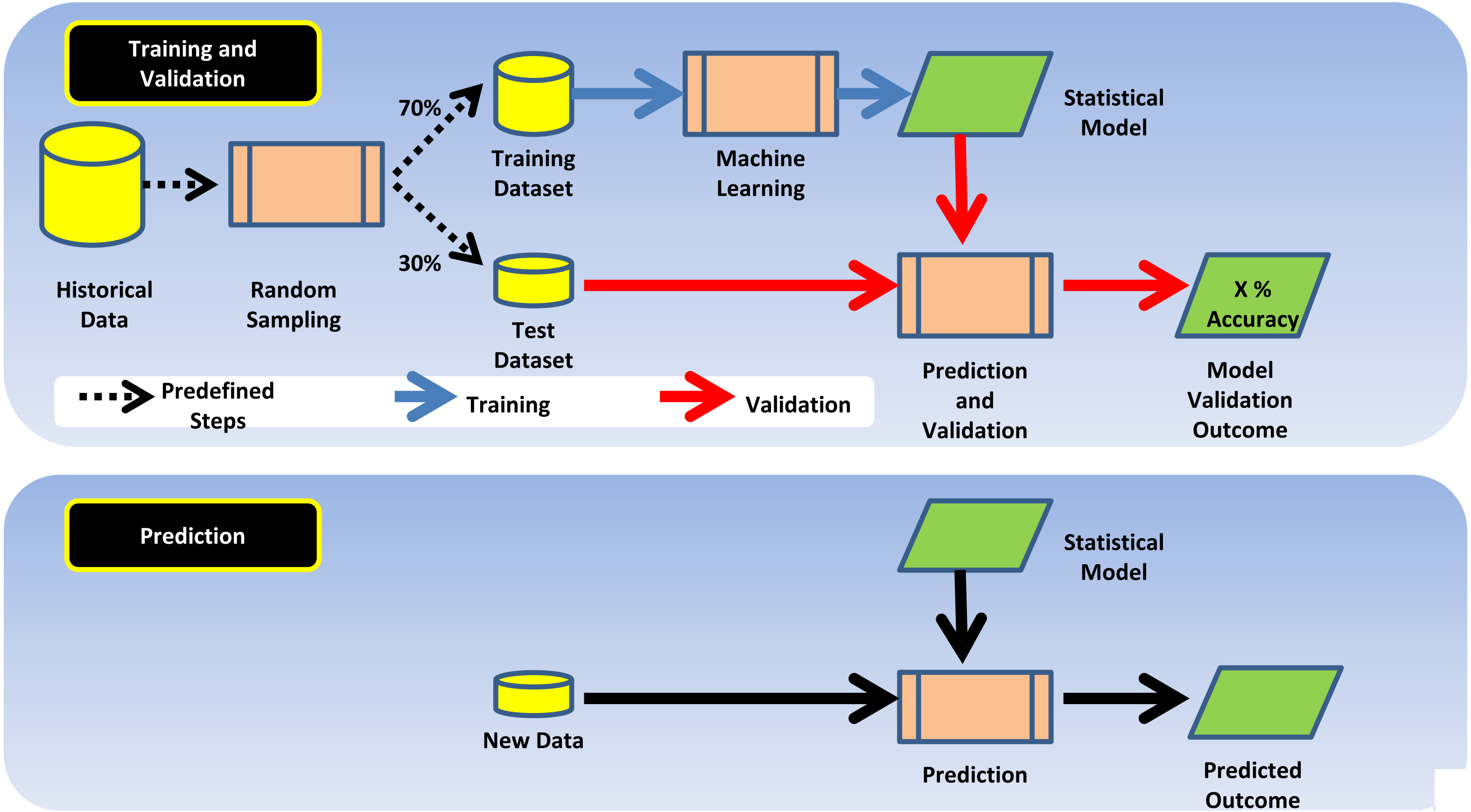
Association Rule Mining-Multiple Cardinalities

S No.	Rules	Support	Confidence	Lift
1	{Green Slime,Jack the Ripper (1988)} => {She}	3.14%	75.34%	3.52739726
2	{Canterville Ghost,Dark of the Sun} => {Strawberry Blonde}	2.57%	66.18%	3.4384273
3	{Jack the Ripper (1988),Strawberry Blonde} => {Canterville Ghost}	2.51%	65.67%	3.362311251
4	{She,Strawberry Blonde} => {Canterville Ghost}	2.51%	63.77%	3.264852954
5	{Canterville Ghost,Pretty Maids All In A Row} => {Strawberry Blonde}	2.57%	62.50%	3.247403561
6	{Dark of the Sun,Doc Savage: The Man of Bronze,She} => {Green Slime}	2.11%	69.81%	3.208389046
7	{Dark of the Sun,Doc Savage: The Man of Bronze,Green Slime} => {She}	2.11%	68.52%	3.207912458
8	{Doc Savage: The Man of Bronze,Pretty Maids All In A Row,She} => {Green Slime}	2.06%	69.23%	3.181708056
9	{Dark of the Sun,Strawberry Blonde} => {Canterville Ghost}	2.57%	60.81%	3.11344239
10	{Pretty Maids All In A Row,Strawberry Blonde} => {Canterville Ghost}	2.57%	60.00%	3.071929825
11	{Doc Savage: The Man of Bronze,Pretty Maids All In A Row} => {Green Slime}	3.26%	66.28%	3.046053836
12	{Doc Savage: The Man of Bronze,Jack the Ripper (1988)} => {She}	2.23%	65.00%	3.043181818
13	{Canterville Ghost,Jack the Ripper (1988)} => {Strawberry Blonde}	2.51%	57.89%	3.008121193
14	{Doc Savage: The Man of Bronze,Green Slime,Pretty Maids All In A Row} => {She}	2.06%	63.16%	2.956937799
15	{Doc Savage: The Man of Bronze,Stranger on the Third Floor} => {Green Slime}	2.57%	64.29%	2.954443195

Sample Interpretation for Rule 1: Those customers who buy 'Green Slime' and 'Jack the Ripper' are generally more prone to buy 'She' also.

Supervised Learning- Process Flow

Supervised Learning- Process Flow



Decision Tree

Learning and Decision Trees to learning

- What is learning?

✓ **More than just memorizing facts.**
✓ **Learning the underlying structure of the problem or data.**

- A fundamental aspect of learning is generalization:

✓ Given a few examples, can you generalize to others?

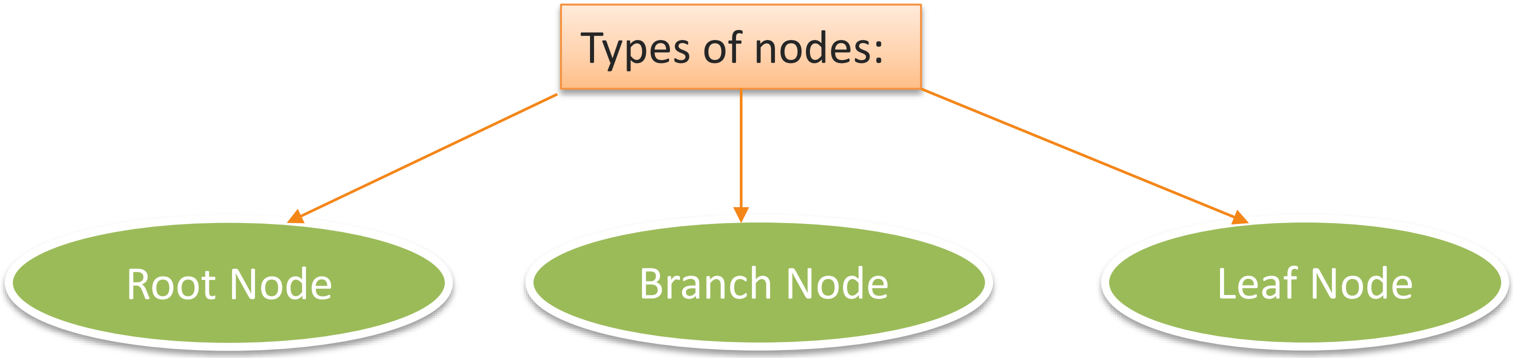
- Learning is ubiquitous:

✓ Medical Diagnosis: Identify new disorders from observations.
✓ Loan Applications: Predict risk of default.
✓ Prediction: (climate, stocks, etc.) Predict future from current and past data.
✓ Speech/Object Recognition: From examples, generalize to others.

What are Decision trees?

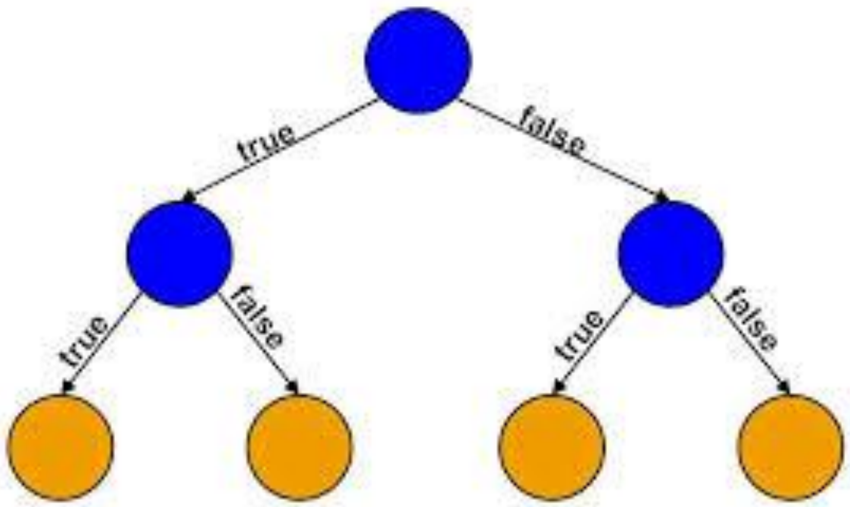
A decision tree is a tree-like structure in which internal node represents test on an attribute, each branch represents outcome of test and each leaf node represents class label (decision taken after computing all attributes). A path from root to leaf represents classification rules.

Thus, a decision tree consists of 3 types of nodes:

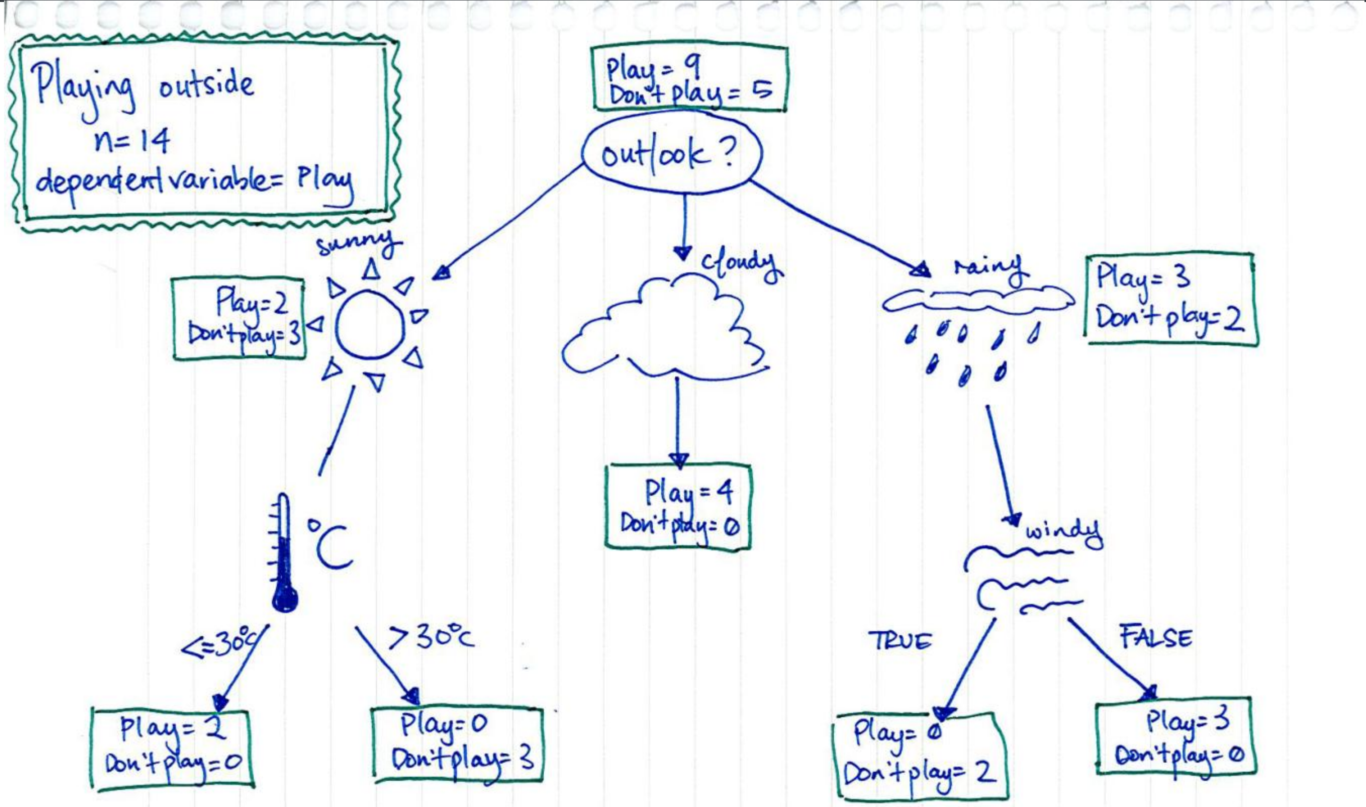


Decision Trees

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value.



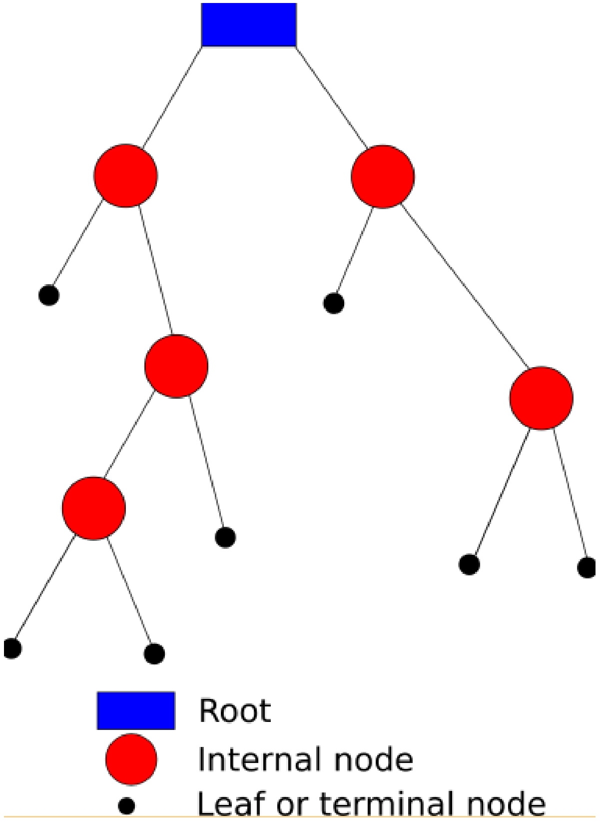
Decision Trees : Example



How to build decision trees?

Use training data to build model Tree generator determines:

- ✓ Which variable to split at a node and the value of the split
- ✓ Decision to stop (make a terminal note) or split again
- ✓ Assign terminal nodes to a class



Decision Tree Examples

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

Step-1:

student

No

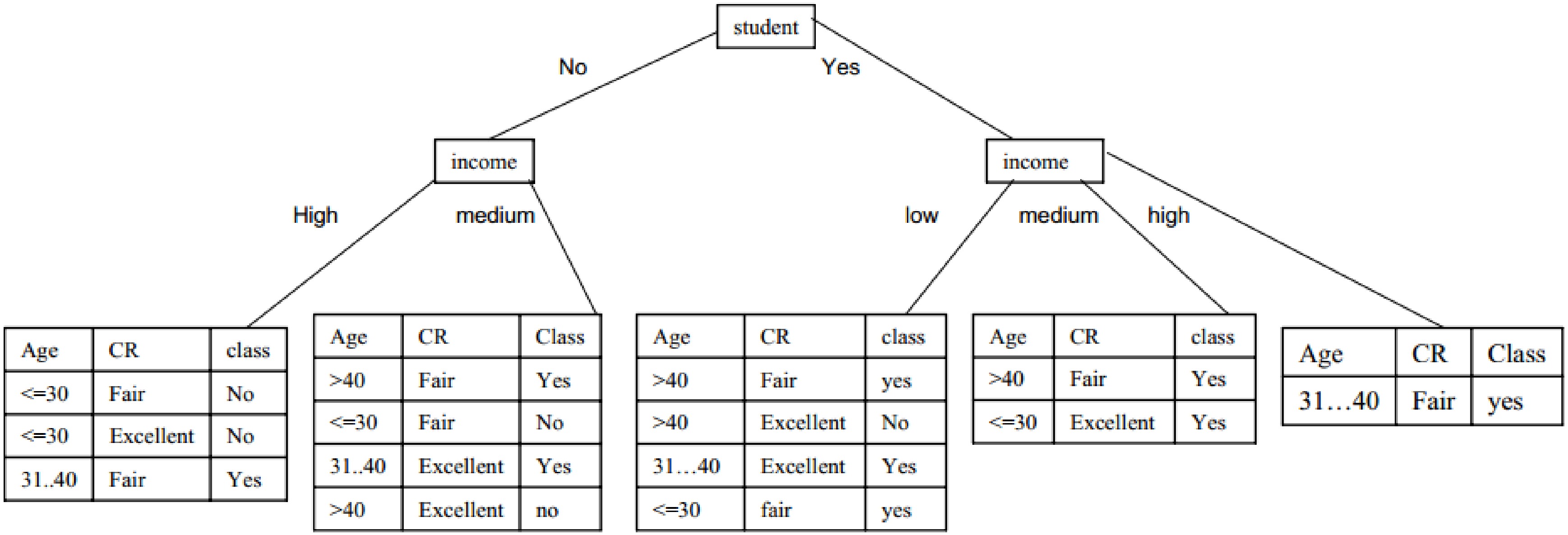
Age	income	CR	Class
<=30	High	Fair	No
<=30	High	Excellent	No
30…40	High	Fair	Yes
>40	Medium	Fair	Yes
<=30	Medium	Fair	No
31…40	Medium	Excellent	Yes
>40	Medium	Excellent	no

Yes

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, Root : Student

Step-2:



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

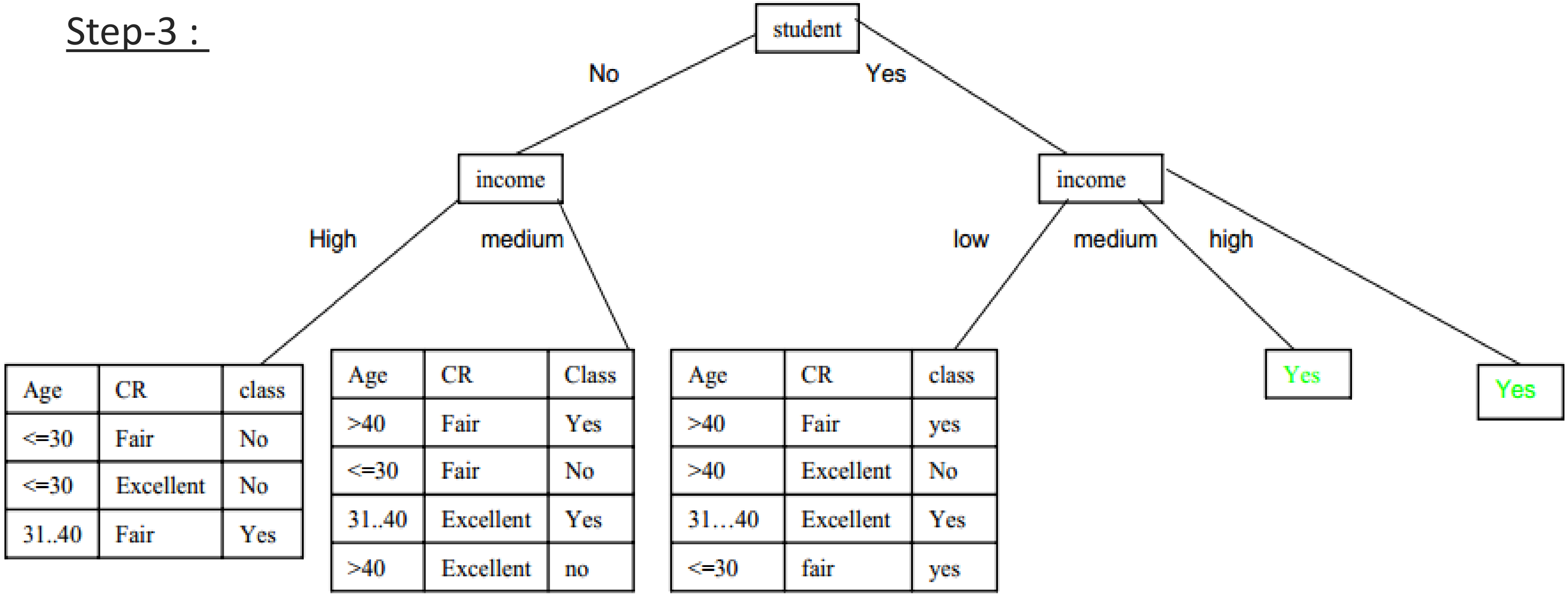
Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

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

Step-3 :



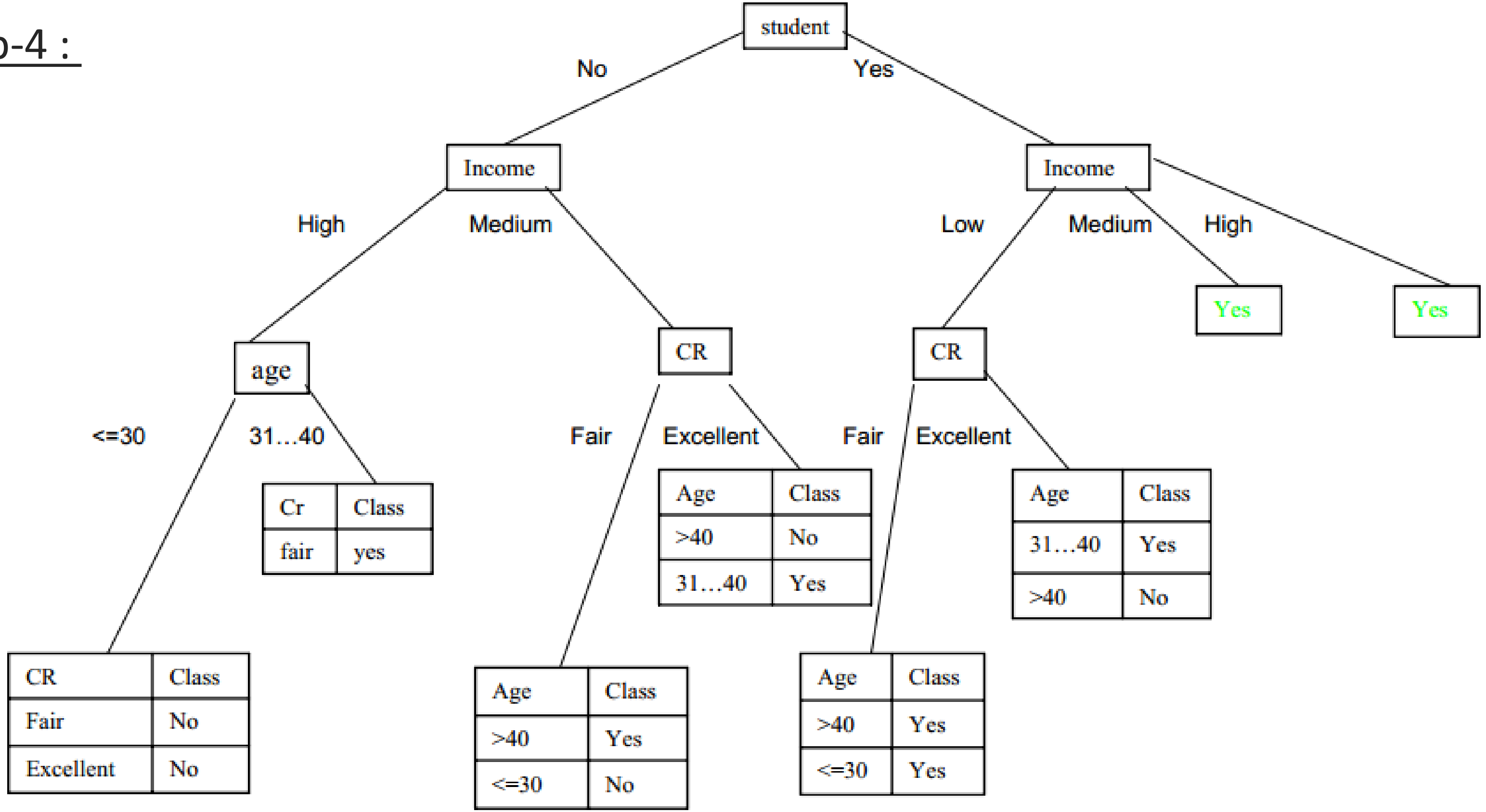
Age	CR	class
<=30	Fair	No
<=30	Excellent	No
31..40	Fair	Yes

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

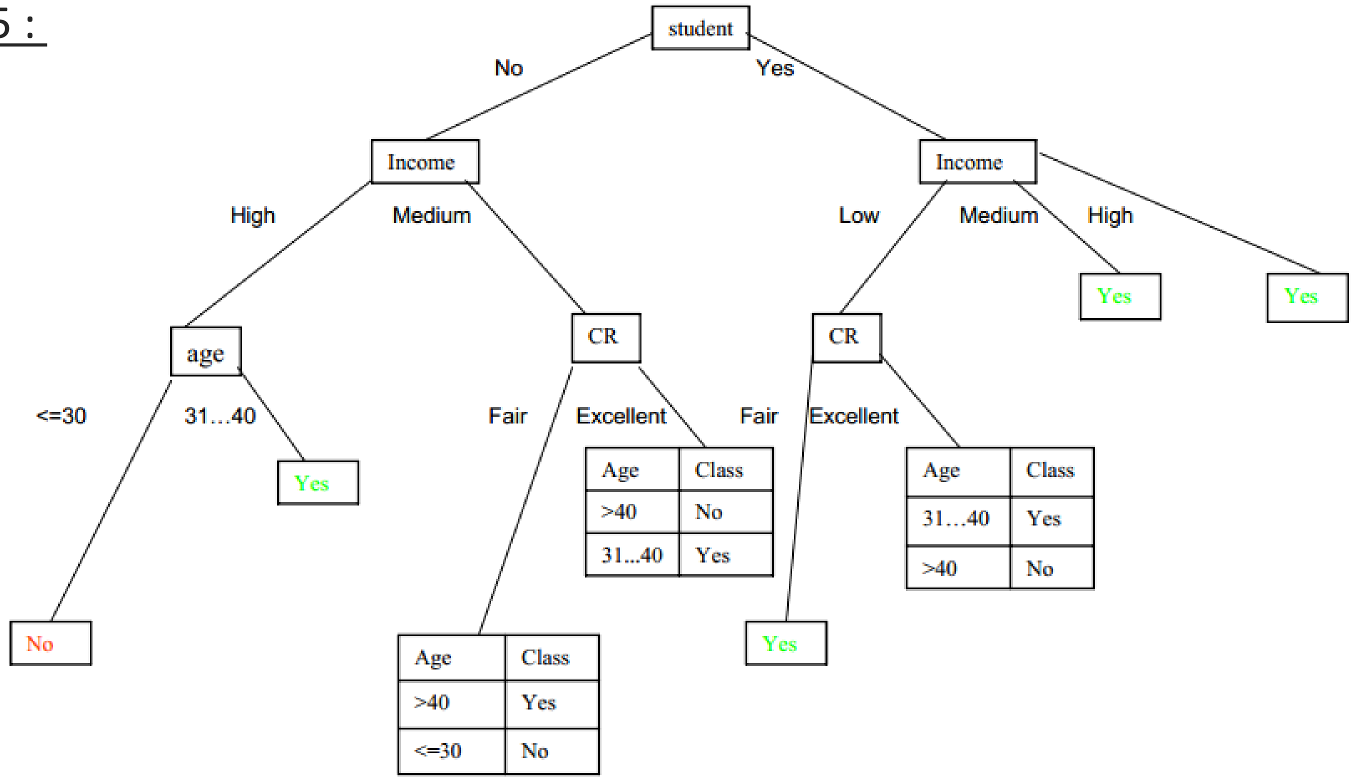
Decision Tree 1, Root : Student

Step-4 :

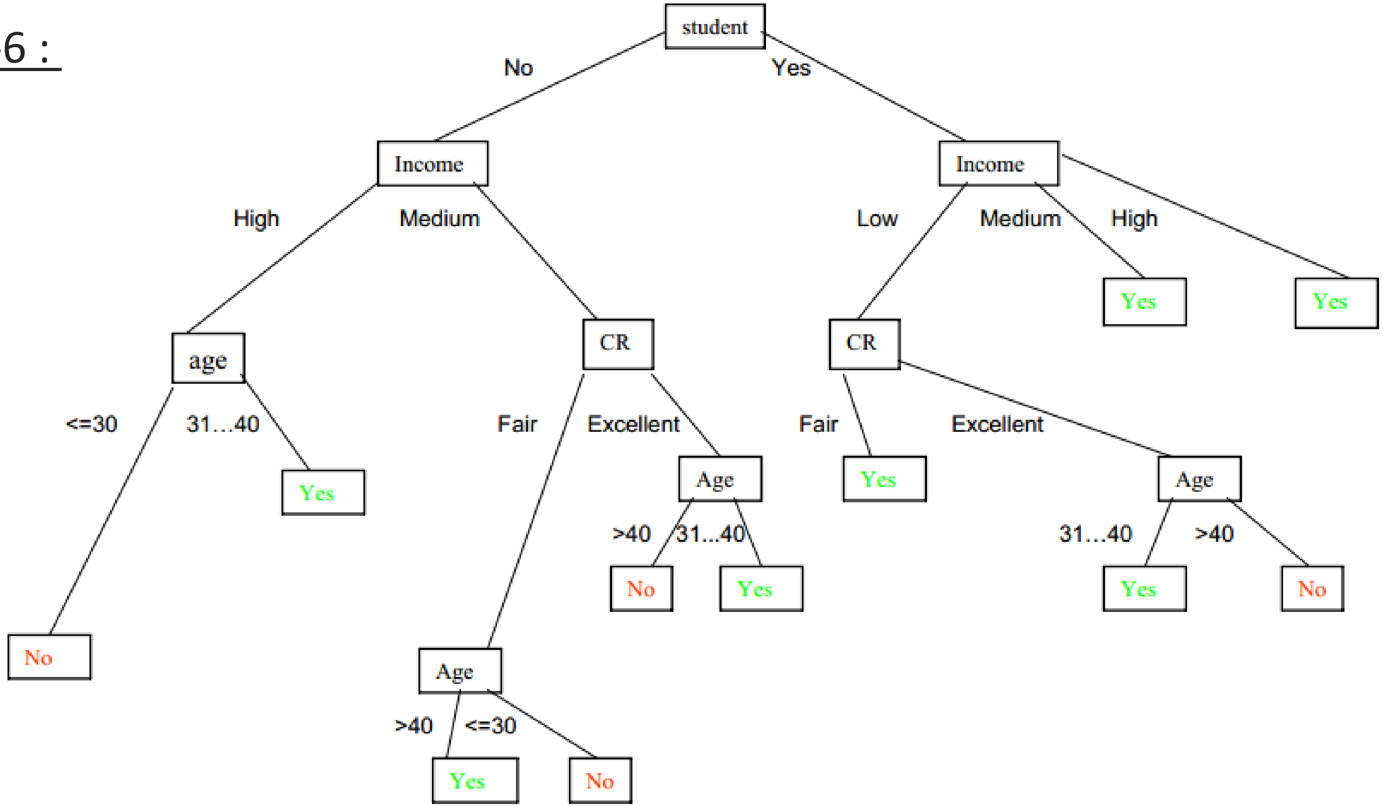


Decision Tree 1, Root : Student

Step-5 :



Step-6 :


student
No
Yes
Income
High
Medium
Income
Low
Medium
High
Yes
Yes
age
<=30
31…40
Yes
No
CR
Fair
Excellent
Age
>40
31...40
No
Yes
Age
>40
<=30
Yes
No
CR
Fair
Excellent
Yes
Age
31…40
>40
Yes
No

Decision Tree 1, Root : Student

Classifcation rules :

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

Decision Tree 2, Root : Income

Step-1 :

Income

High Medium Low

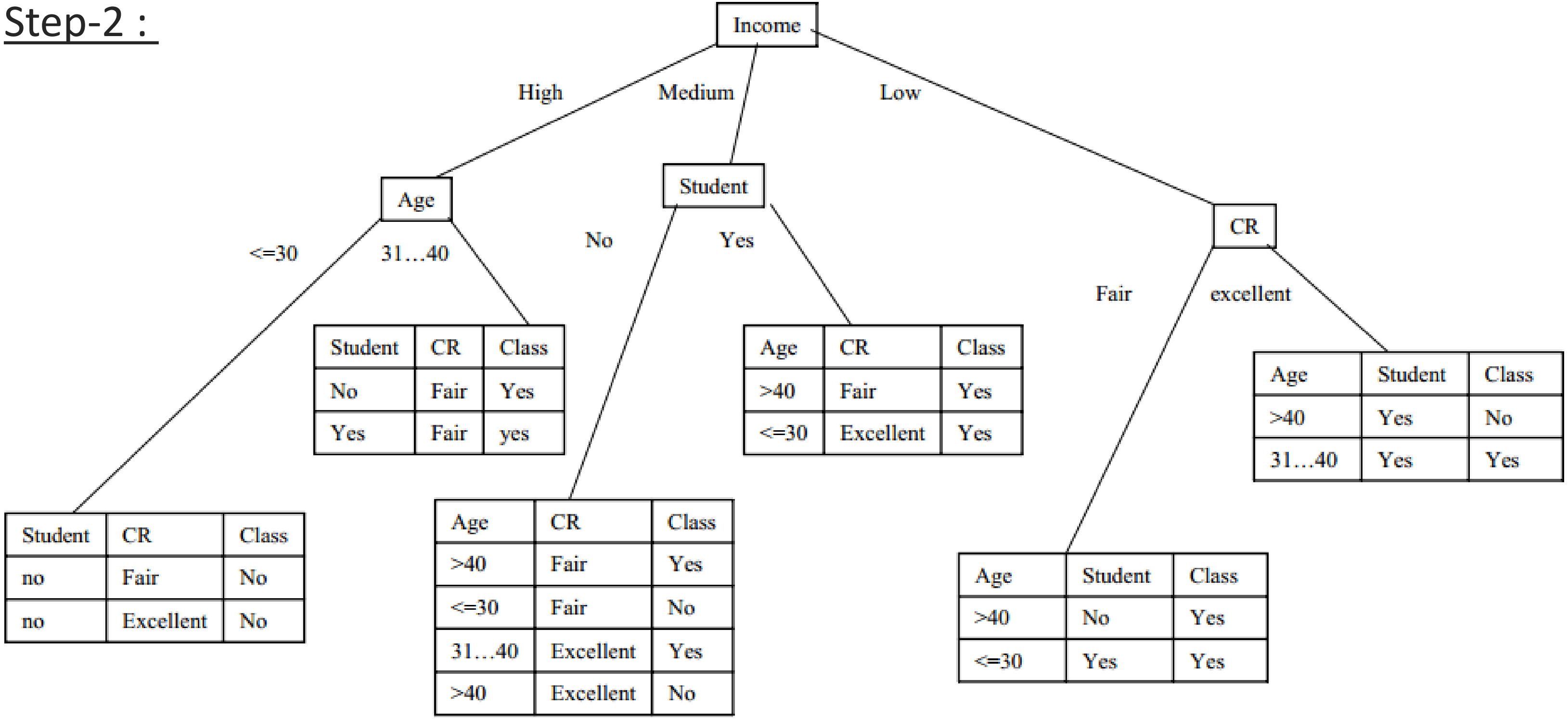
Age	Student	CR	Class
<=30	No	Fair	No
<=30	No	Excellent	No
31…40	No	Fair	Yes
31…40	Yes	Fair	yes

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, Root : Income

Step-2 :



Student	CR	Class
no	Fair	No
no	Excellent	No

Student	CR	Class
No	Fair	Yes
Yes	Fair	yes

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

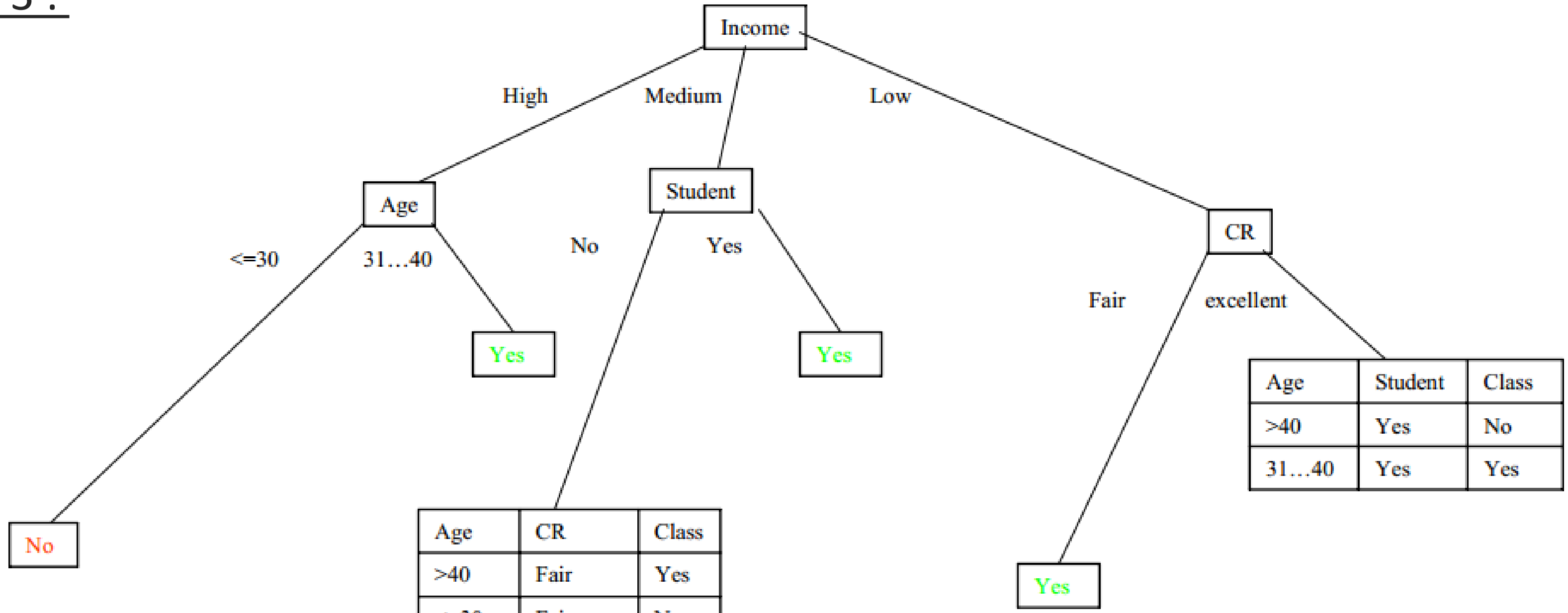
Age	CR	Class
>40	Fair	Yes
<=30	Excellent	Yes

Age	Student	Class
>40	No	Yes
<=30	Yes	Yes

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

Decision Tree 2, Root : Income

Step-3 :

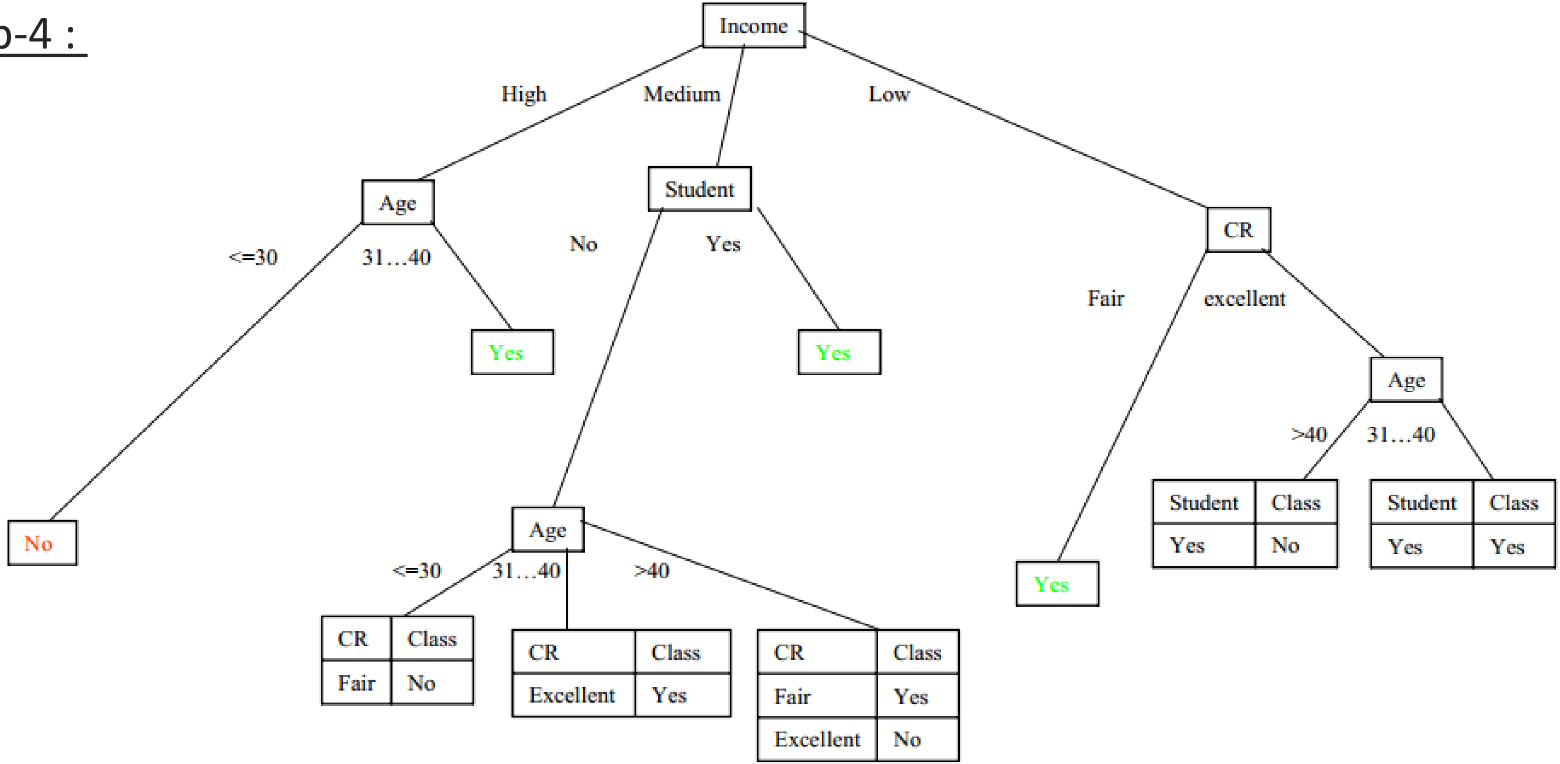


Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31…40	Excellent	Yes
>40	Excellent	No

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

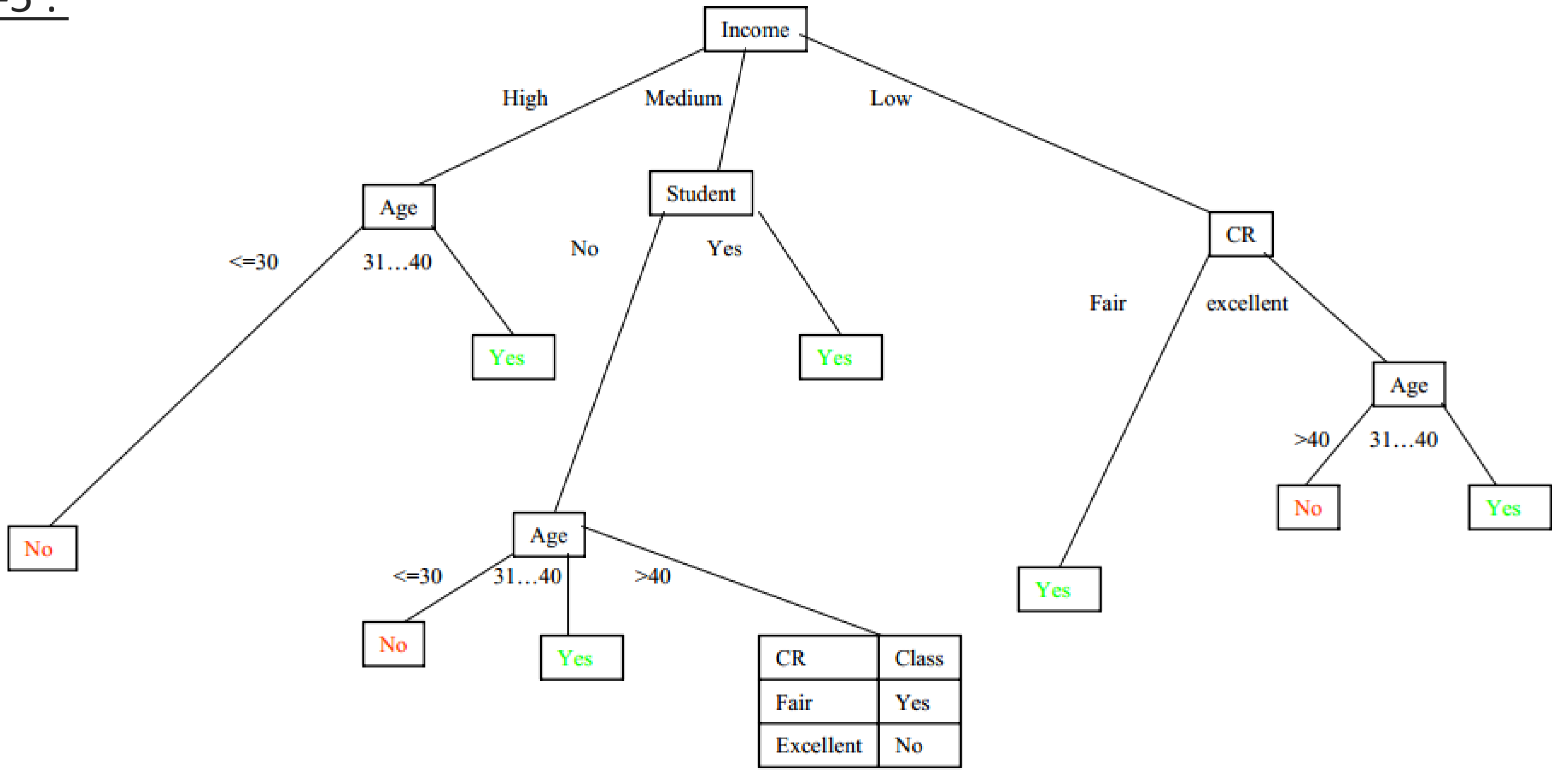
Decision Tree 2, Root : Income

Step-4 :



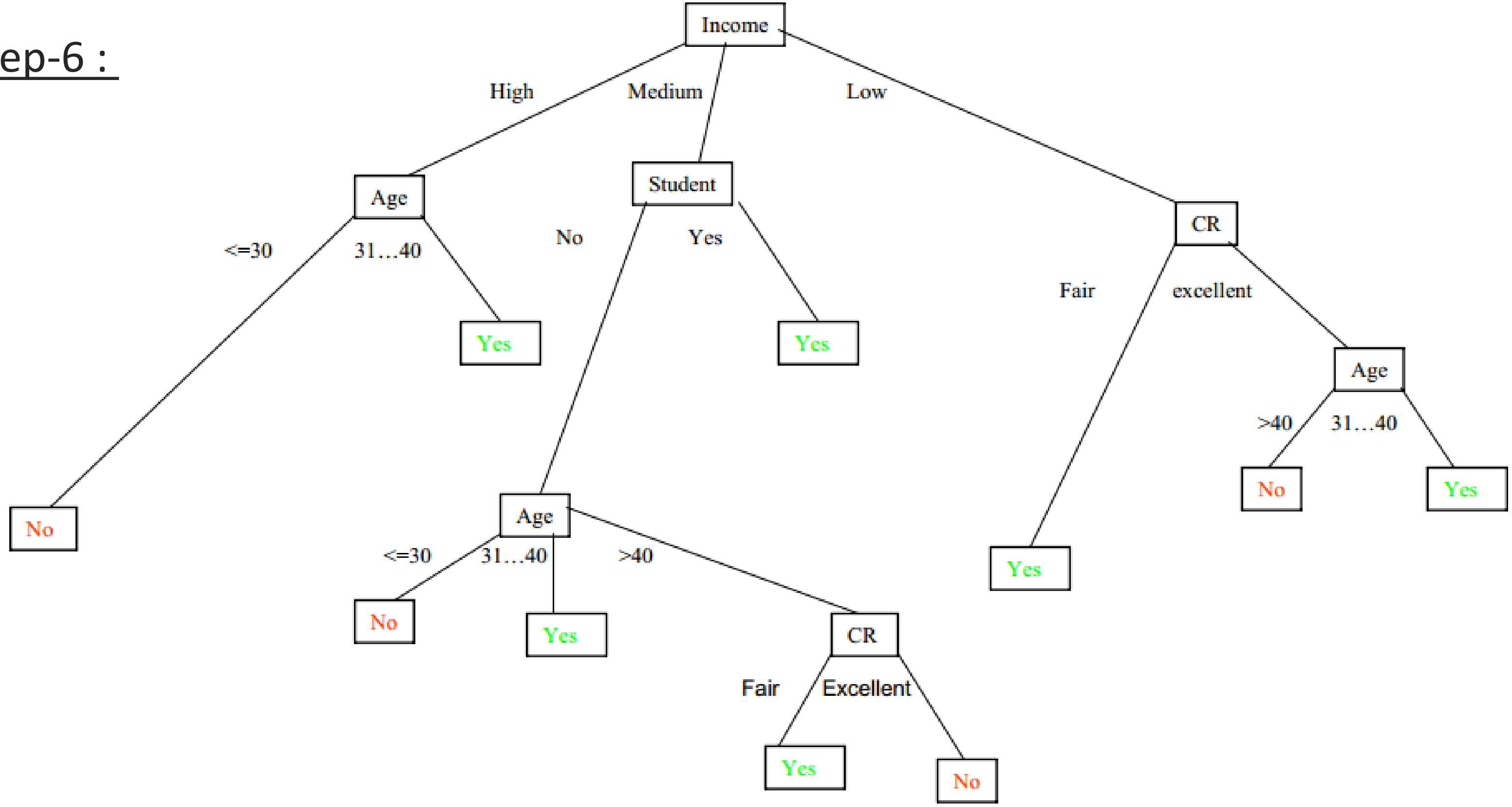
Decision Tree 2, Root : Income

<u>Step-5 :</u>



Decision Tree 2, Root : Income

Step-6 :



Decision Tree 2, Root : Income

Classifcation rules :

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

Which Tree to choose?

The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree.

The topmost decision node in a tree which corresponds to the best predictor is called root node.

Information Gain:

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

Formulas for information gain

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

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

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

Which Tree to choose?

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

Which Tree to choose?

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

Which Tree to choose?

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

Which Tree to choose?

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