SOLVED PROBLEMS/ADDITIONAL EXAMPLES/NOTES

These slides are beyond the ppt and given for better explanation, case studies and some additional topics like logistic regression apart from linear regression..

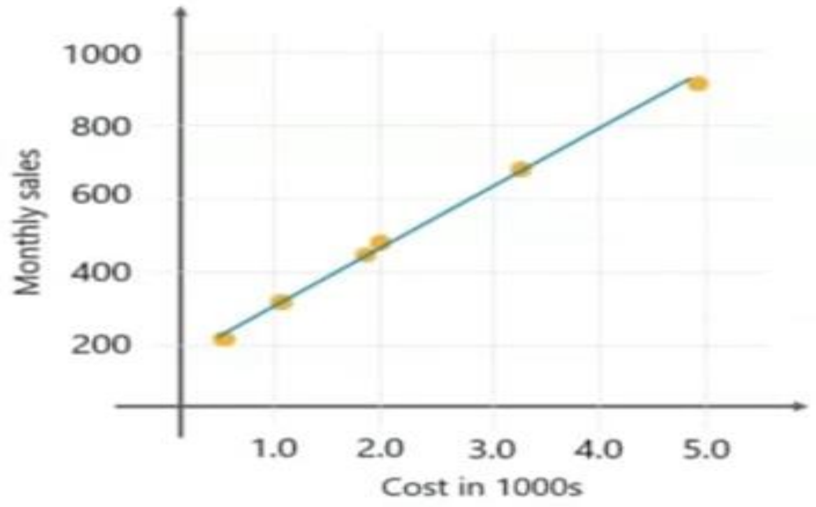
CONTENTS:

1. linear vs logistic regression
2. Calculation of accuracy and FOIL index for classification rules
3. Confusion matrix
4. Step by Step Decision Tree Induction and Classification Rules formation
5. Information Gain – Attribute selection
6. Simplified classification rules – Derived method (from decision tree)

Linear Regression Use Case

To forecast monthly sales by studying the relationship between the monthly e-commerce sales and the online advertising costs.

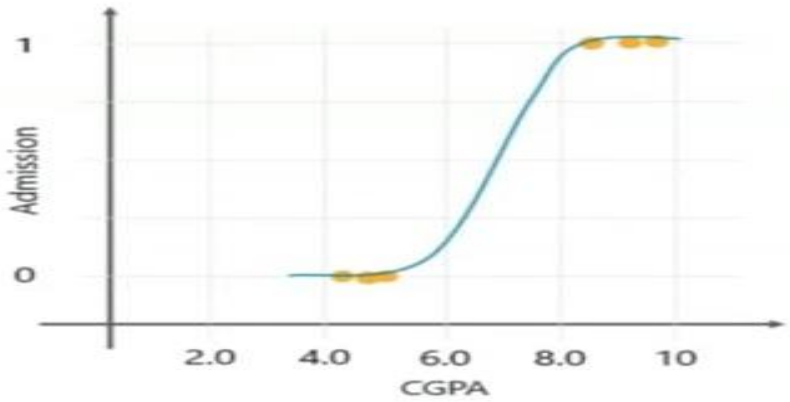
Monthly sales	Advertising cost In 1000s
200	0.5
900	5
450	1.9
680	3.2
490	2.0
300	1.0



Logistic Regression Use Case

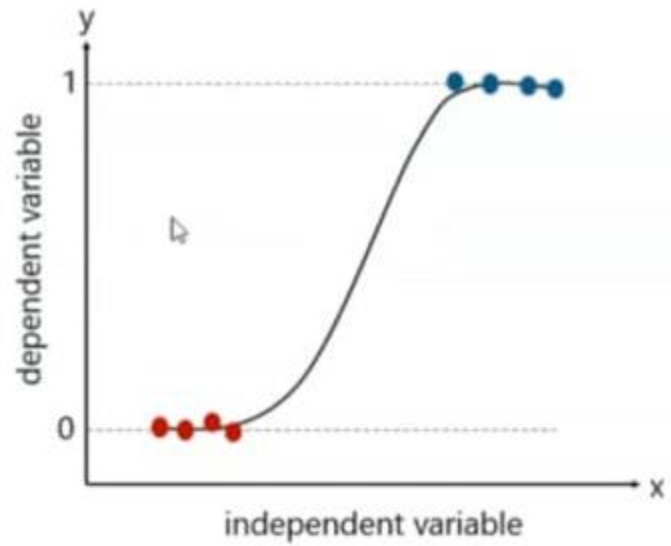
To predict if a student will get admitted to a school based on his CGPA.

Admission	CGPA
0	4.2
0	5.1
0	5.5
1	8.2
1	9.0
1	9.1



What Is Logistic Regression?

Logistic Regression is a method used to predict a dependent variable, given a set of independent variables, such that the dependent variable is categorical.



$$\log\left(\frac{Y}{1-Y}\right) = C + B1X1 + B2X2 + \ldots$$

- Y is the probability of an event to happen which you are trying to predict
- x1, x2 are the independent variables which determine the occurrence of an event i.e. Y
- C is the constant term which will be the probability of the event happening when no other factors are considered

Linear Regression Vs Logistic Regression

	Linear Regression	Logistic Regression
1 Definition	To predict a continuous dependent variable based on values of independent variables	*To predict a categorical dependent variable based on values of independent variables*
2 Variable Type	Continuous dependent variable	Categorical dependent variable
3 Estimation method	Least square estimation	Maximum like-hood estimation
4 Equation	$Y = b_0 + b_1 x + e$	$\log\left(\frac{Y}{1-Y}\right) = C + B1X1 + B2X2 + \ldots.$
5 Best fit line	Straight line	Curve
6 Relationship between DV & IV	Linear relationship between the dependent and independent variable	Linear relationship is not mandatory
7 Output	Predicted integer value	Predicted binary value (0 or 1)
8 Applications	Business domain, forecasting sales	Classification problems, cybersecurity, image processing

4. Consider a training set that contains 100 positive examples and 400 negative examples. For each of the following candidate rules,

R_1: $A \rightarrow +$ (covers 4 positive and 1 negative examples),
R_2: $B \rightarrow +$ (covers 30 positive and 10 negative examples),

R_3: $C \rightarrow +$ (covers 100 positive and 90 negative examples),

determine which is the best and worst candidate rule according to:

a) Rule accuracy.

Answer: The accuracies of the rules are 80% (for R_1), 75% (for R_2), and 52.6% (for R_3), respectively. Therefore R_1 is the best candidate and R_3 is the worst candidate according to rule accuracy.

b) FOIL's information gain.

Answer: Assume the initial rule is $\varnothing \rightarrow +$. This rule covers $p_0 = 100$ positive examples and $n_0 = 400$ negative examples.
The rule R_1 covers $p_1 = 4$ positive examples and $n_1 = 1$ negative example. Therefore, the information gain for this rule is

$$4\,[\log(4/5) - \log(100/500)] = 8.$$

The rule R_2 covers $p_1 = 30$ positive examples and $n_1 = 10$ negative examples. Therefore, the information gain for this rule is

$$30\,[\log(30/40) - \log(100/500)] = 57.2$$

The rule R_3 covers $p_1 = 100$ positive examples and $n_1 = 90$ negative examples. Therefore, the information gain for this rule is

$$100\,[\log(100/190) - \log(100/500)] = 139.6$$

Therefore, R_3 is the best candidate and R_1 is the worst candidate according to FOIL's information gain.

c) The likelihood ratio statistic.

Answer: For R_1, the expected frequency for the positive class is $5 \times 100/500 = 1$ and the expected frequency for the negative class is $5 \times 400/500 = 4$. Therefore, the likelihood ratio for R_1 is

$$2 \times [\,4 \times \log_2(4/1) + 1 \times \log_2(1/4)\,] = 12.$$

The numbers *not* on the diagonal (the **Red Boxes**) are samples the algorithm messed up.

		Actual: Has Heart Disease	Actual: Does Not Have Heart Disease
Predicted	Has Heart Disease	142	22
	Does Not Have Heart Disease	29	110

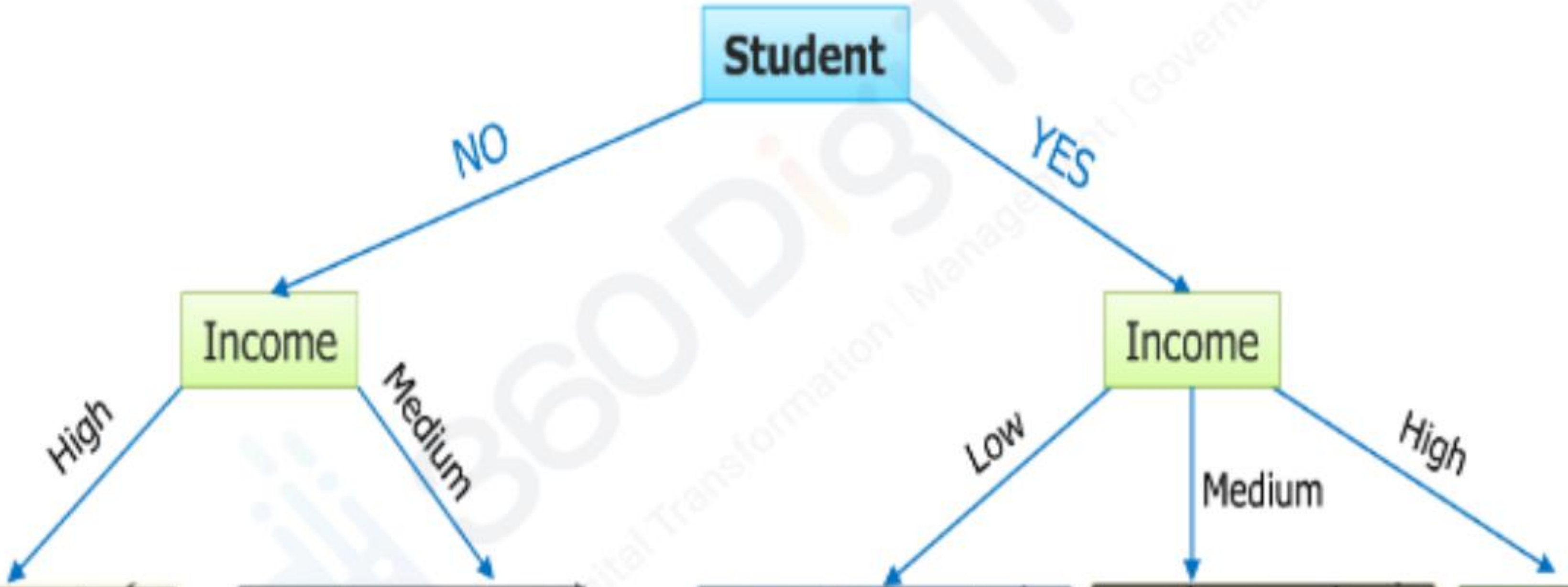
Records	Age	Income	Student	Credit_Rating	Buys_Computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	> 40	Medium	No	Fair	Yes
r5	> 40	Low	Yes	Fair	Yes
r6	> 40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	> 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

Step-1

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

Step-2



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

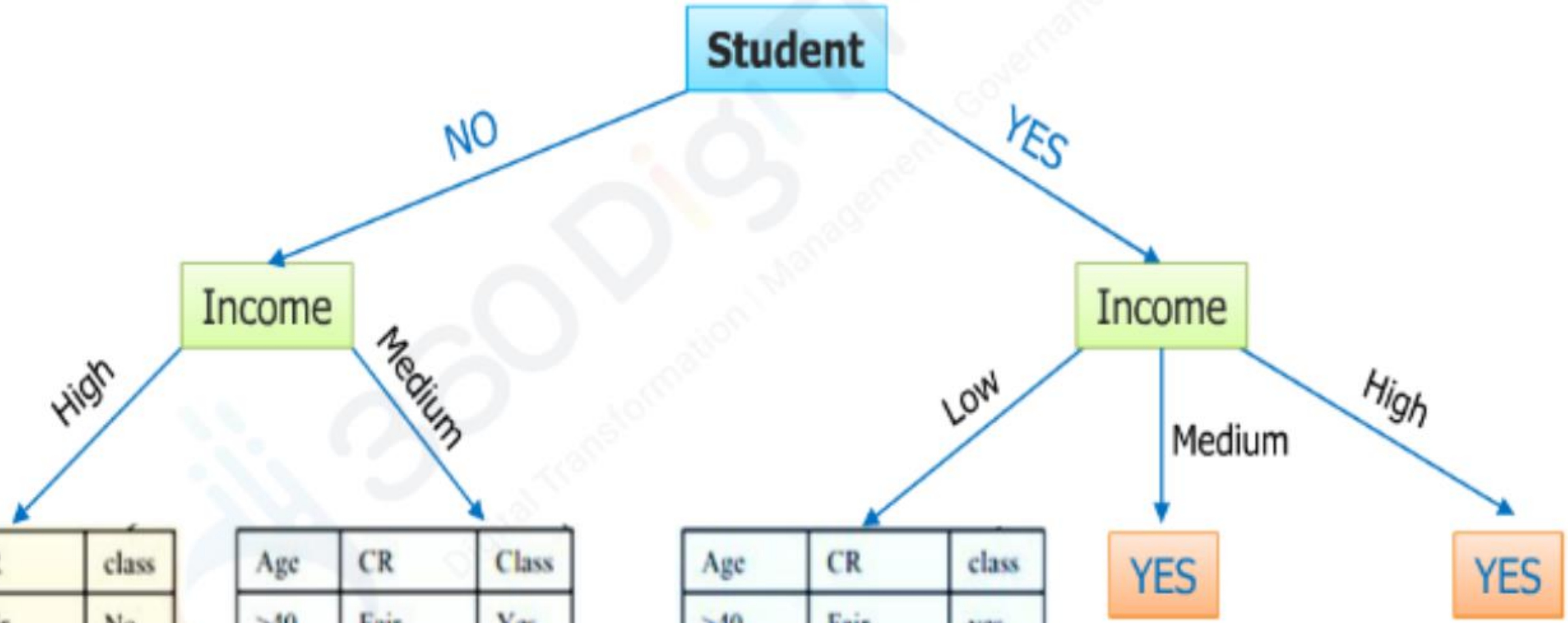
Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

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

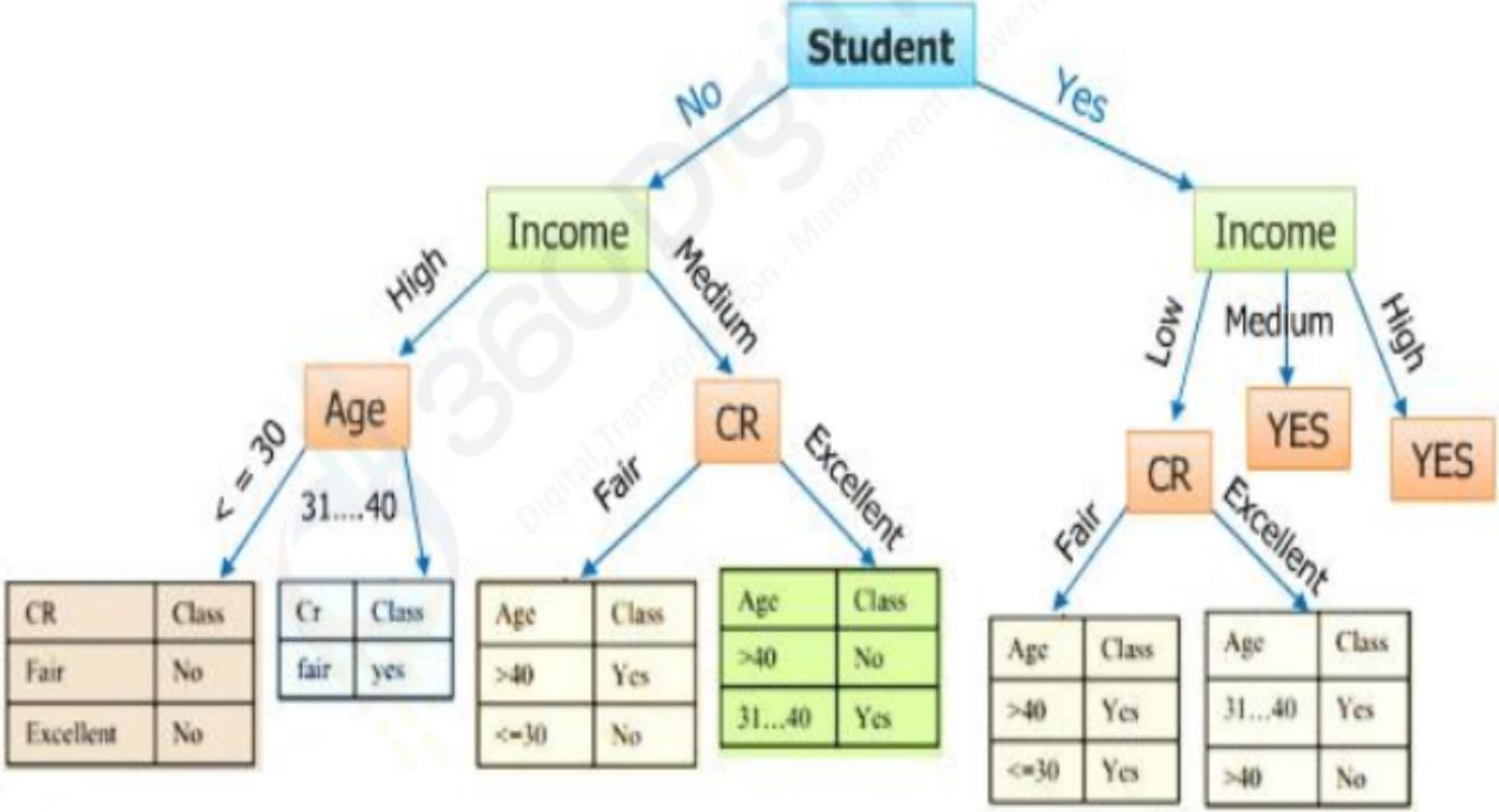


Age	CR	class
<=30	Fair	No
<=30	Excellent	No
31..40	Fair	Yes

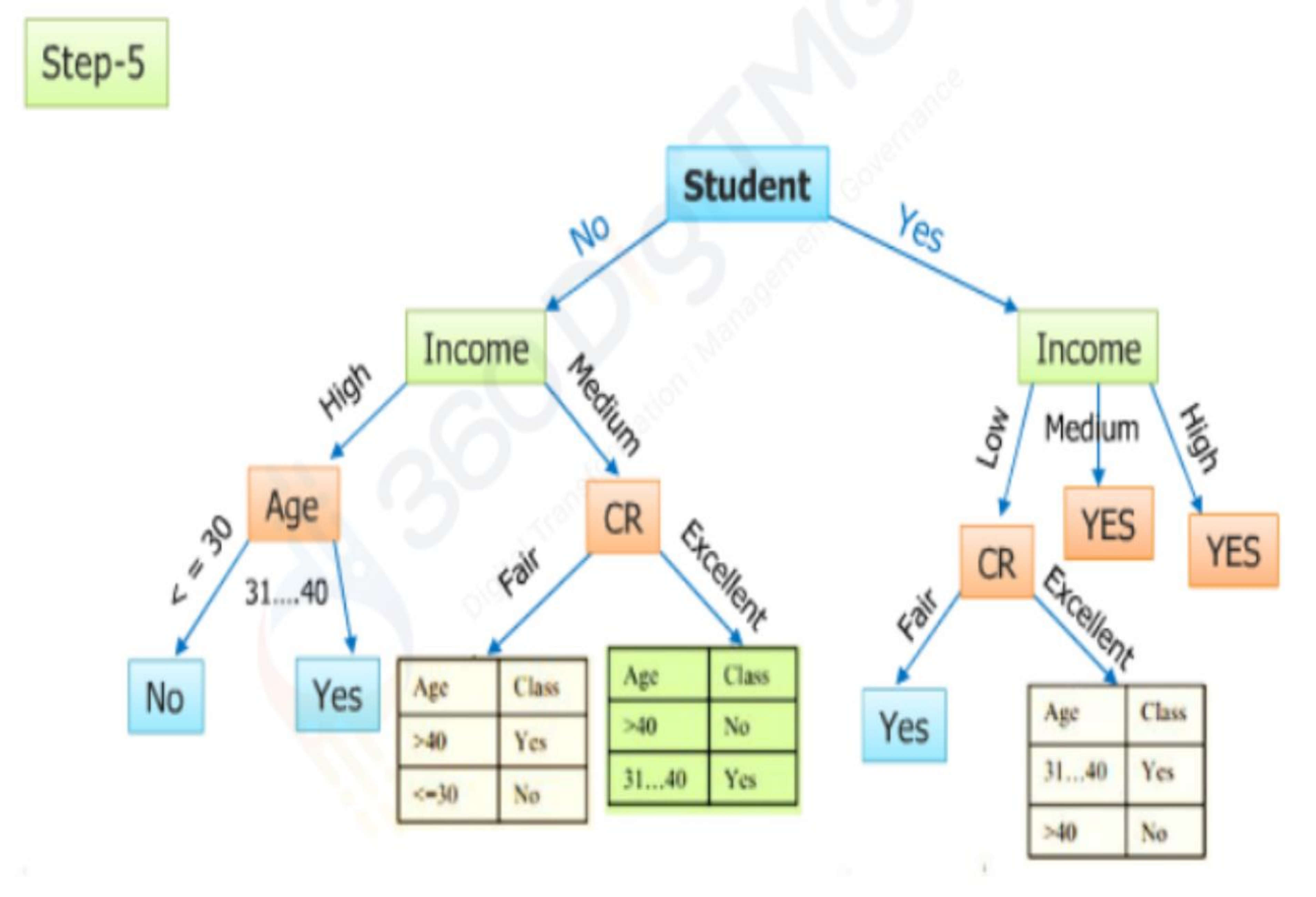
Age	CR	Class
>40	Fair	Yes
<=30	Fair	No
31..40	Excellent	Yes
>40	Excellent	no

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

Step-4

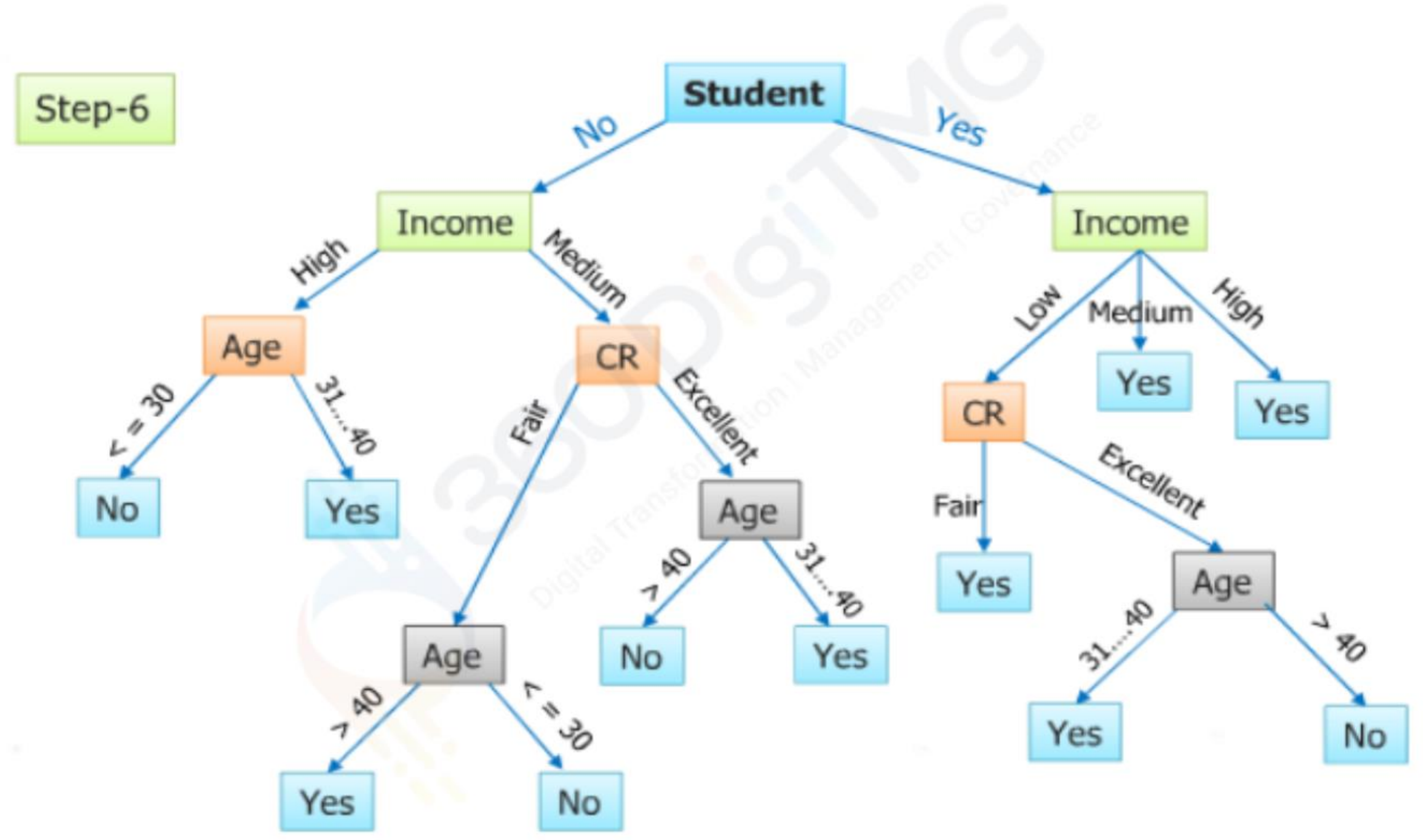


Student
No
Yes
Income
Income
High
Medium
Low
Medium
High
Age
CR
CR
YES
YES
< = 30
31....40
Fair
Excellent
Fair
Excellent
CR Class
Fair No
Excellent No
Cr Class
fair yes
Age Class
>40 Yes
<=30 No
Age Class
>40 No
31...40 Yes
Age Class
>40 Yes
<=30 Yes
Age Class
31...40 Yes
>40 No



Step-5
Student
No
Yes
Income
High
Medium
Age
< = 30
31....40
No
Yes
CR
Fair
Excellent
Age
Class
>40
Yes
<=30
No
Age
Class
>40
No
31...40
Yes
Income
Low
Medium
High
YES
YES
CR
Fair
Excellent
Yes
Age
Class
31...40
Yes
>40
No

Decision Tree 1; Root = Student



Classification 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

Now consider another approach to build the decision tree.
Let's take the attribute: Income

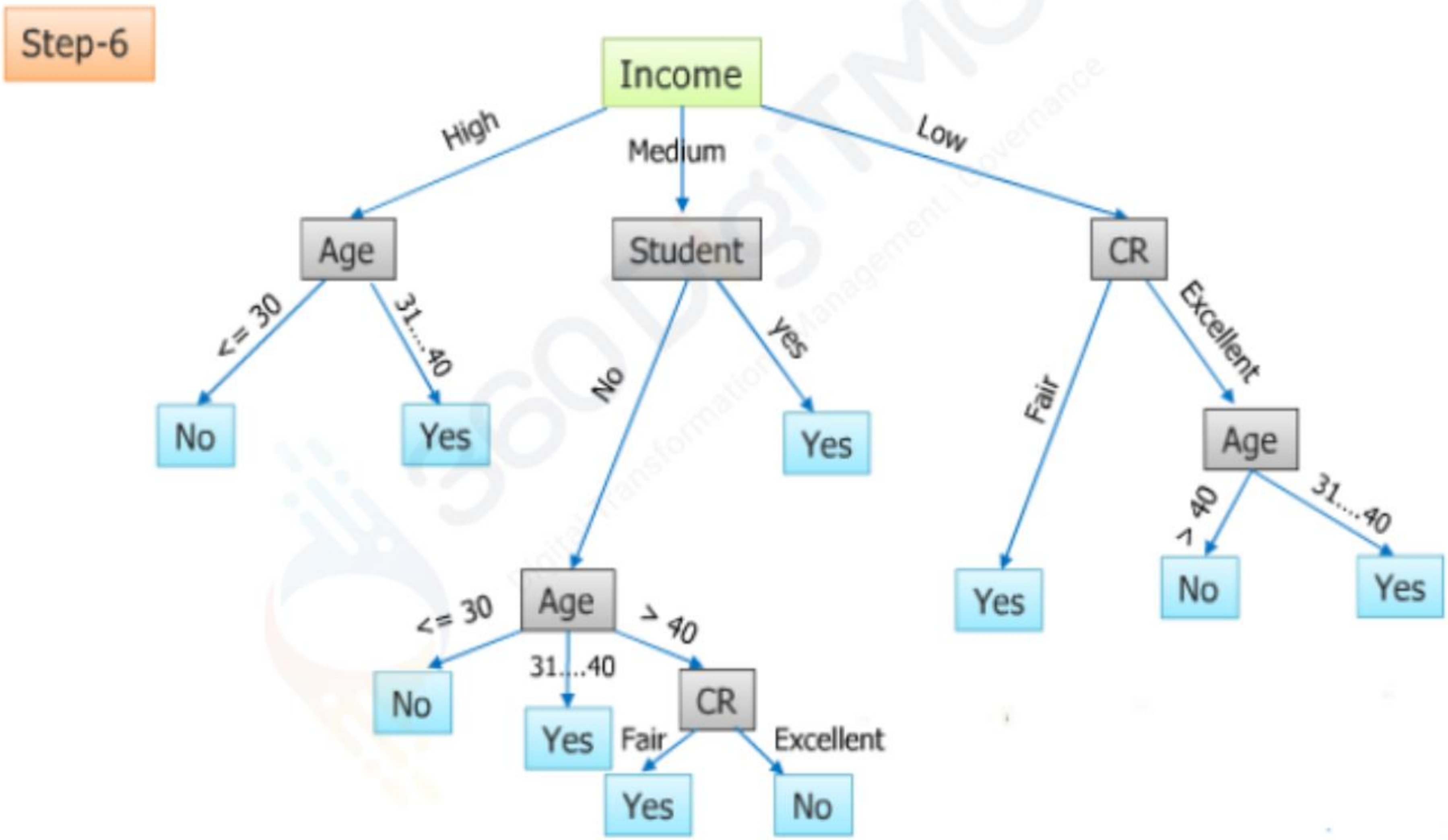
Income

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; Root = Income



Decision Tree 2; Root = Income

Classification 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)

Greedy Approach & Entropy

We want to use the attribute that does the "belt job" splitting up the training data, but can this be measured?

We use entropy and information gain

Entropy:

- → Measure of disorder or impurity
- → We will find entropy of the output values of a set of training instances
- → If output vales split 50-50% set is impure – 1
- → If output is 0, set is Pure – 0
- → If the output values are split 25-75% then entropy – 0.811

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).

Information Gain: Attribute Selection

Heuristic: Select the attribute with the highest information gain i.e., attribute that results in most homogeneous branches

Let pi be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, D|/|D|

→ Expected information (entropy) needed to classify a tuple in D:

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

→ Information needed (after using A to split D into v partitions) to classify D:

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

→ Information gained by branching on attribute A

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

Attribute Selection

→ Two classes

» Postive: buys_computer=yes $P(buys = yes) = \frac{9}{14}$

» Negative: buys_computer=no $P(buys = no) = \frac{5}{14}$

→ Entropy in D

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

Age	Income	Student	Credit Rating	Buys Computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31...40	High	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31...40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31...40	Medium	No	Excellent	Yes
31...40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

→ What is information gain if we split on "Age"?

age	pos	neg	I(pos,neg)
<=30	2	3	0.971
31...40	4	0	0
>40	3	2	0.971

$Info(Age <= 30) = 0.971$

$Info(31 <= Age <= 40) = 0$

$Info(Age > 40) = 0.971$

$$Info_{age}(D) = \frac{5}{14} Info(Age <= 30) + \frac{4}{14} Info(Age = 31..40) + \frac{5}{14} Info(Age > 40)$$

$$= \frac{5}{14} * 0.971 + \frac{4}{14} * 0 + \frac{5}{14} * 0.971$$

$$= 0.694$$

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

$$= 0.246$$

9/14 => 0.6428 ;;;   5/14 => 0.3571
-0.6428 * logbase2 (0.6428) - 0.3571*logbase2 (0.3571)
logbase2 (0.6428) => log(0.6428) / log(2) => --.1919 / 0.3010 => -0.6375
logbase2 (0.3571) => log(0.3571) / log(2) => -0.4472 / 0.3010 => -1.4857
- 0.6428 * - 0.6375 - 0.3571 * -1.4857 => 0.4097 + 0.5305 => 0.9402

-2/5 logbase2 (2/5) - 3/5 logbase2(3/5)   =>
-0.4 log base 2 (0.4) - 0.6 logbase 2 (0.6) =>
log base 2 (0.4) => log (0.4) / log (2) => -0.3979 / 0.3010 => - 1.32192
log base 2 (0.6) => log (0.6) / log (2) => -0.2218 / 0.3010 =>   -0.73687
-0.4 * -1.32192 - 0.6 * -0.73687
=> 0.52876 + 0.44212 => 0.9708 => 0.971

Attribute Selection – Information Gained

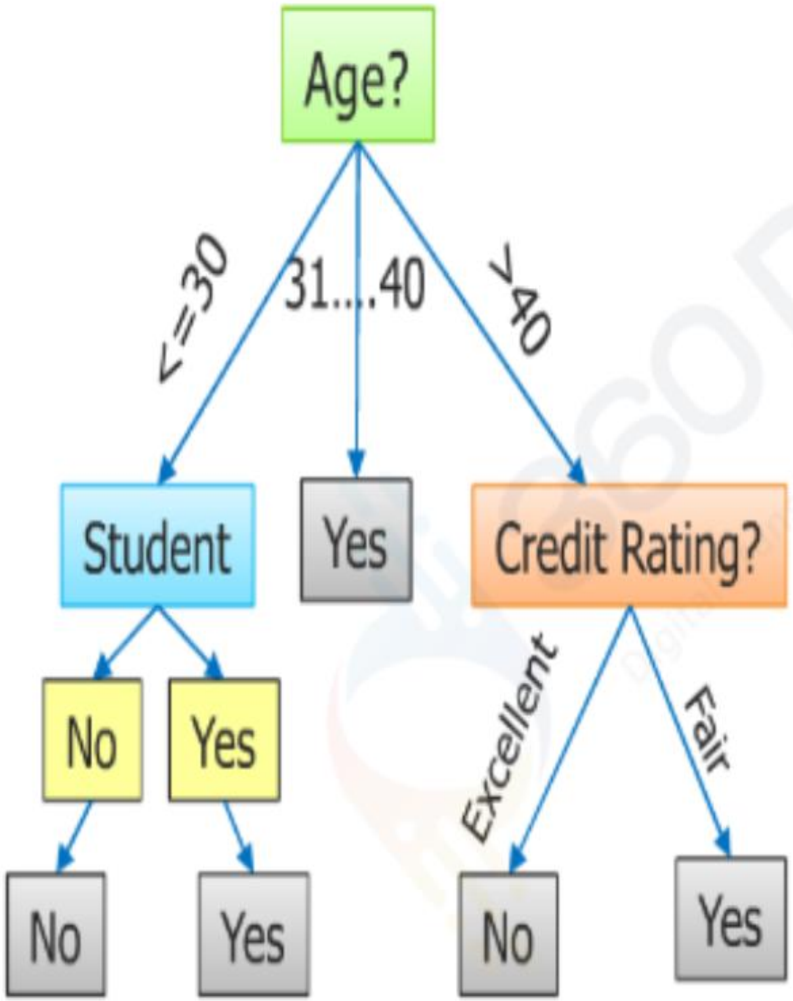
$$Gain(Age) = 0.246$$
$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Age

Age	Income	Student	Credit Rating	Buys Computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes

Age	Income	Student	Credit Rating	Buys Computer
31...40	High	No	Fair	Yes
31...40	Low	Yes	Excellent	Yes
31...40	Medium	No	Excellent	Yes
31...40	High	Yes	Fair	Yes

Age	Income	Student	Credit Rating	Buys Computer
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
>40	Medium	Yes	Fair	Yes
>40	High	No	Excellent	No



→ Age(<30) ^ student(no) = NO
→ Age(<30) ^ student(yes) = YES
→ Age(31…40) = YES
→ Age(>40) ^ credit_rating(excellent) = NO
→ Age(>40) ^ credit_rating(fair) = Yes