PROBLEM 2 DECISION TREES (30pts)]

Part 1 (5pts)

1. List and explain shortly the Decision Tree Algorithm Attribute Selection Measures.

2. Describe in your words what is their role in the Decision Tree Construction and which kind of trees they produce". Draw a picture as an example.

Part 2 (25pts) BUILDING DECISION TREE CLASSIFIER

Given Classification DB

O	a1	a2	C
o1	1	1	1
o2	0	0	0
o3	0	1	0
o4	0	0	0
o5	1	1	1
o6	1	1	0
o7	0	0	0
o8	1	0	1

Use the above DB and **repeated two fold cross validation holdout** to build a CLASSIFIER using the DT BASIC Algorithm with **a1** as the root. Follow **Stages 1- 4** of the process of building a classifier.

Remember that division into 2-folds is an ARBITRARY partition of records into 2 disjoint sets; so there may be many answers depending on the partitions.

Repeat the two fold cross validation holdout 4 times.

Build your final CLASSIFIER using the following folds for each repetitions round **1.- 4.**.

1. **f1** = $\{o1, o2, o3, o4\}$ for training - rest for testing.
2. **f2** = $\{o5, o6, o7, o8\}$ for training - rest for testing.
3. **f2** = $\{o1, o3, o5, o7\}$ for training - rest for testing.
4. **f2** = $\{o2, o4, o6, o8\}$ for training - rest for testing.

Here are the **STEPS** you must follow

STEP 1 (10pts)

Follow **Stages 1-3** to build, for each repetitions round **1.- 4.**, a learned classifier (base classifier, learned model) and name them **F1** , **F2** , **F3** , **F4** , respectively.

Write the learned classifiers **F1 - F4** in as a set of **discriminant rules** in the predicate form.

STEP 2 (10pts)

Perform the **Stage 4** as follows: use the learned classifiers **F1 - F4** and their predictive accuracy and rules accuracy as metrics to choose ONE as your final CLASSIFIER **F**

STEP 3 (5pts)

Construct as your final CLASSIFIER the **bagged** Ensemble Classifier $\mathbf{F}^*$

Use your CLASSIFIERS **F** and $\mathbf{F}^*$ to classify the following records and compare the results.

O	a1	a2
o1	0	1
o2	0	0
o3	1	0
o4	1	1

Problem-2 Decision Trees.

Part-1.

① Decision Tree Algorithm Attribute Selection measures:

Given a training data set, there are many ways to choose the root and nodes attributes while constructing a decision tree. These methods of choosing attributes are called Attribution selection measures.

Some possible choices of selecting attributes can be Random, Attribute with Smallest/Largest number of values etc.

The three attribute selection methods that give good results are:

i) Information Gain:

In this measure we have a special order i.e information gain as a measure of goodness of split. The attribute with highest information gain is always chosen as the split decision attribute for the current node while building a tree.

Let p_i be the probability that the arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$

- Expected information (entropy) needed to classify a tuple in D is

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

- Information needed (after using A to split D into v partitions) to classify D is

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

- Information gained by branching on attribute is

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

For continues-valued attribute A, to determine the best split point for A:

- sort the value A in increasing order.
- Typically, the midpoint between each pair of adjacent values is considered as possible split point.
- The point with the minimum expected information requirement for A is selected as split point for A.

- This attribute with highest information gain is selected as splitting attribute.

2) Gain Ratio:

Information gain attribute selection measure is biased towards attributes with a large number of values. Gain ratio is used to overcome this problem.

Gain ratio simply applies normalization to information gain.

$$\text{GainRatio}(A) = \text{Gain}(A) / \text{SplitInfo}(A)$$

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

The attribute with maximum gain ratio is selected as splitting attribute.

3) Gini index:

If a data set D contains examples from n classes, gini index, gini(D) is defined as

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

where p_j is the relative frequency of class j in D

- If a dataset D is split on A into two subsets D_1 and D_2, the giniIndex gini(D) is defined as

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

- Reduction in impurity $\Delta gini(A) = gini(D) - gini_A(D)$
- The attribute that provides smallest $gini_{split}(D)$ i.e (the largest reduction in impurity) is chosen to split the node.
- This approach is biased to multivalued attributes.
- When the number of classes increases, this approach faces difficulty. It also favors tests that result in equal sized partitions.

Some other attribute selection measures are:

CHAID - The measure is based on χ^2-test.

C-SEP - This measure sometimes may perform better than information gain and gini index.

G-statistics - This gives close approximation to x^2-distribution.

MD2 (Minimal Description Length) - This measure prefers the simplest solution.

CART - This measure will perform multivariant split based on linear combination of attributes

Most of the attribute selection methods give good results.

② Splitting Criterion:

Splitting Criterion tells us which attributes to test at Node N by determining the "best" way to separate or partition the tuples in D into individual classes.

The splitting criterion indicates the splitting attribute may also indicate either a

split point or a splitting subset.

- Using any of the three attribute selection measures out of Information Gain, Gain Ratio and Gini Index, the splitting criterion can be determined.
  - The kind of tree that gets produced while building the decision tree depends on the type of data that exists in the splitting attribute.
  - There are three scenarios on the type of data that can present for an attribute.

  i) Discrete valued:

  In this case, the outcomes of the test at Node N correspond directly to the known values of A. A branch is created for each known value a_j, of A and labeled with that value.

  ii) Continuous valued:

  In this case, the test at node N has two possible outcomes, corresponding to the conditions $A \leq$ split-point and $A >$ split-point, where

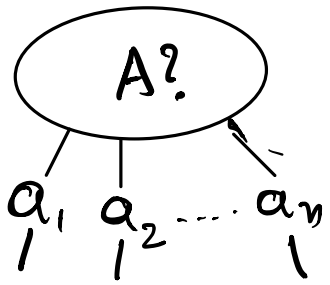
split_point is the split-point returned by Attribute_selection_measure as part of splitting criterion.

iii) Discrete valued & Binary tree must be produced:

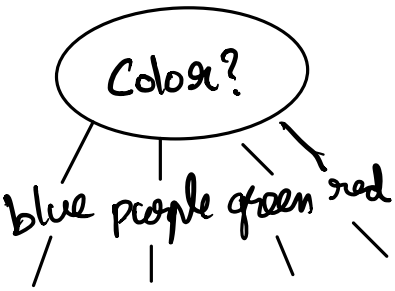
In this case the test at node N is of the form "$A \in S_A$?", where S_A is splitting subset of A, returned by Attribute selection measure as part of splitting criterion. Two branches are grown from N.

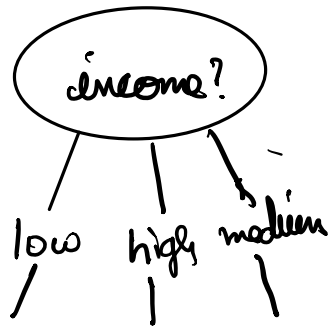
Partitioning scenarios | examples.

(a)

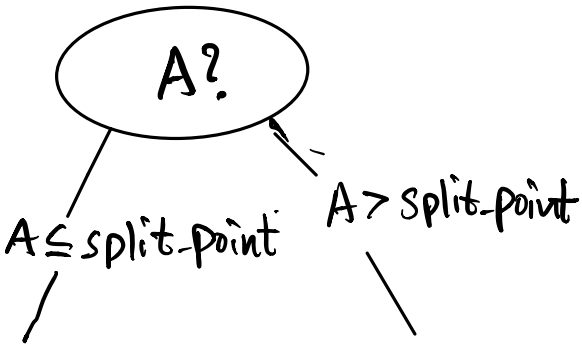


(discrete valued)

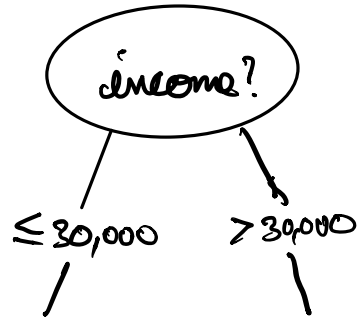




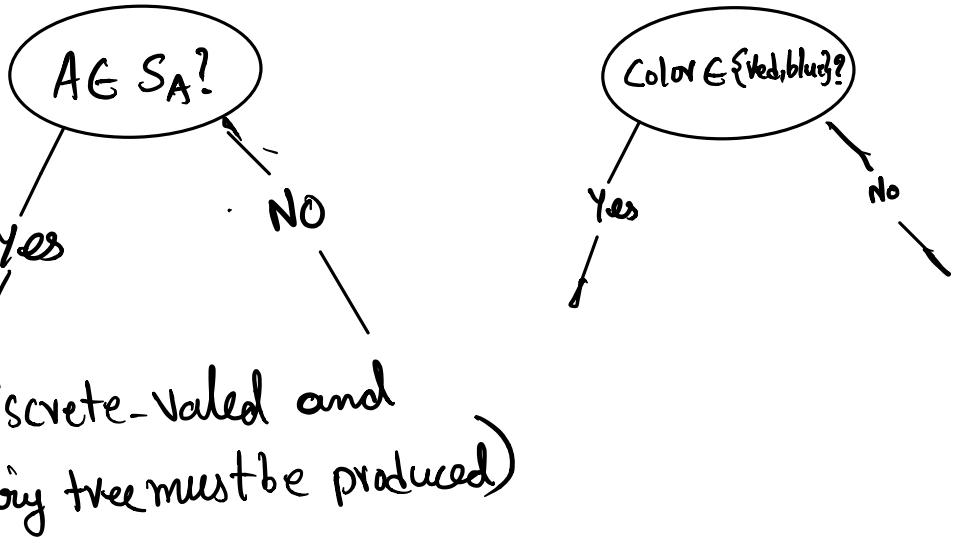
(b)



(Continuous valued)



(G)



Detailed examples for various attribute selection measures are described below:

Information Gain:

In Information Gain type of attribute selection measure, the attribute with highest information gain is always chosen as the split decision attribute for the current node while building a tree.

Information gain measure is biased towards multivalued attributes.

The decision tree that gets built using this measure need not be a binary tree or a skewed tree to one side.

Example:

RID	age	income	Student	Credit_rating	Class: buys_Comp
1.	Youth	high	no	fair	no
2.	Youth	high	no	excellent	no
3.	middle-aged	high	no	fair	yes
4.	senior	medium	no	fair	yes
5.	senior	low	yes	fair	yes
6.	senior	low	yes	excellent	no
7.	middle-aged	low	yes	excellent	yes
8.	Youth	medium	no	fair	no
9.	Youth	low	yes	fair	yes
10.	senior	medium	yes	fair	yes
11.	Youth	medium	yes	excellent	yes
12.	middle-aged	medium	no	excellent	yes
13.	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Computing gain on each attribute:

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)$$

$$+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right)$$

$$+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2 \frac{3}{5} - \frac{2}{5}\log_2 \frac{2}{5}\right)$$

$$= 0.694$$

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

$$\therefore \text{Gain}(age) = \text{Info}(D) - \text{Info}_{age}(D) = 0.940 - 0.694$$
$$= 0.246 \text{ bits}$$

Similarly, calculating gain for other attributes we get

$$\text{Gain}(income) = \text{Info}(D) - \text{Info}_{income}(D)$$

$$= 0.029 \text{ bits}$$

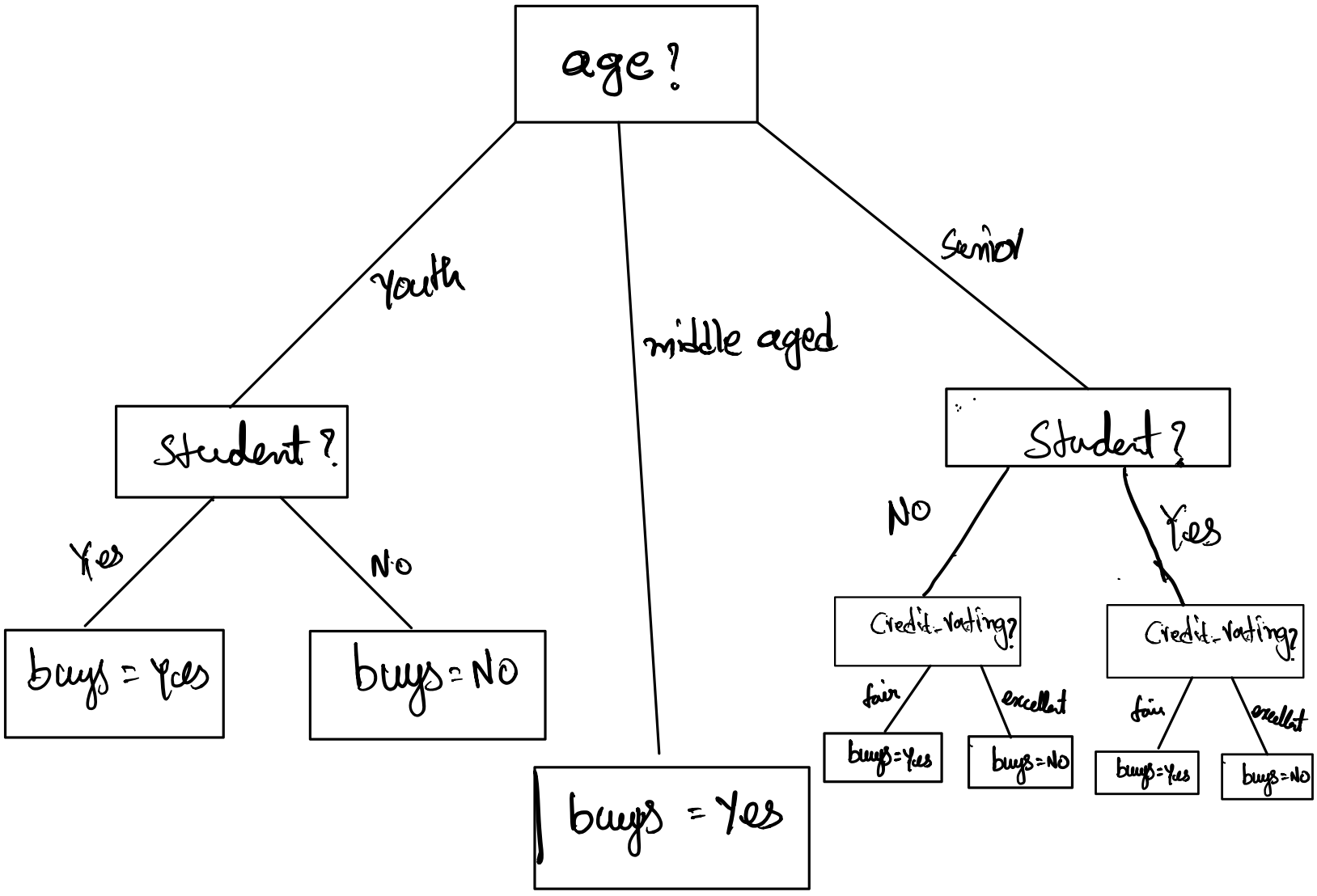
$$\text{Gain}(student) = \text{Info}(D) - \text{Info}_{student}(D)$$

$$= 0.151 \text{ bits}$$

$$\text{Gain}(credit\_rating) = \text{Info}(D) - \text{Info}_{credit\_rating}(D)$$

$$= 0.048 \text{ bits.}$$

Since the age attribute has the highest information gain and therefore becomes the splitting attribute at the root node of decision tree.



Gain Ratio:

In this measure, the attribute with maximum gain ratio is selected as the splitting attribute for the current node.

- Gain Ratio tends to prefer unbalanced splits in which one partition is much smaller than others.

For the previous table, calculating the Gain Ratio's to determine the split:

From the previous example we know

$$\text{Gain (income)} = 0.029$$

$$\text{Gain (age)} = 0.246$$

$$\text{Gain (student)} = 0.151$$

$$\text{Gain (credit\_rating)} = 0.048.$$

$$\text{SplitInfo}_{(\text{income})}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \log_2\left(\frac{4}{14}\right)$$

$$= 1.557$$

$$\text{Gain Ratio (income)} = 0.029/1.557 = 0.019.$$

$$\text{SplitInfo}_{(\text{age})}(D) = -\frac{5}{14} \log_2\left(\frac{5}{14}\right) - \frac{4}{14} \log_2\left(\frac{4}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right)$$

$$= 0.5305 + 0.5163 + 0.5305$$

$$= 1.5773$$

$$\therefore \text{Gain Ratio (age)} = 0.246/1.5773$$

$$= 0.1559.$$

$$\text{SplitInfo}_{(\text{student})}(D) = -\frac{1}{2}(-1) - \frac{1}{2}(1)$$

$$= 1$$

$$\therefore \text{Gain Ratio (student)} = 0.151/1 = 0.151$$

$$\text{SplitInfo}_{\text{Credit\_rating}}(D) = -\frac{8}{14}\log_2(8/14) - \frac{6}{14}\log_2(6/14)$$

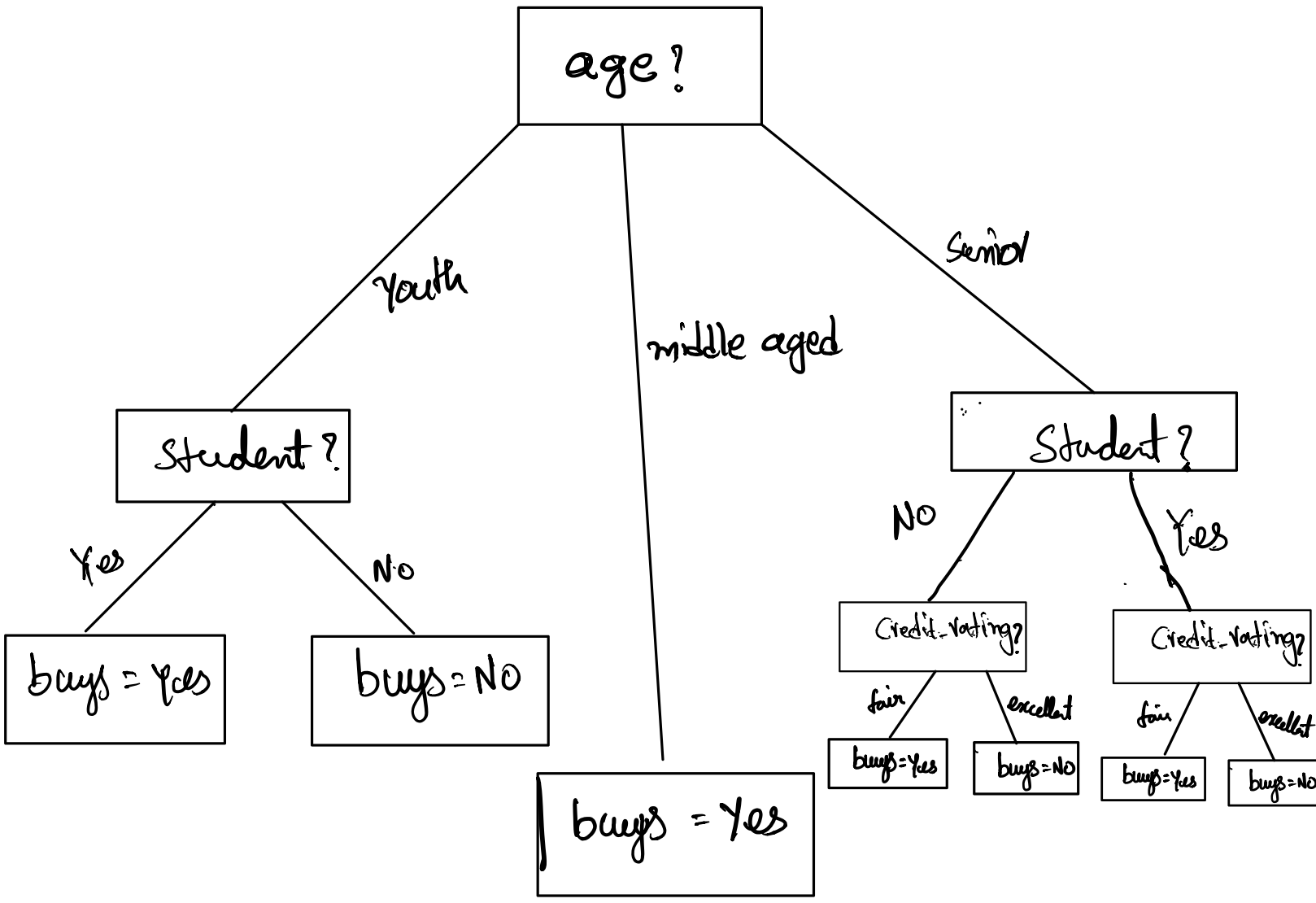
$$= 0.4613 + 0.5238$$

$$= 0.9851$$

$$\text{Gain Ratio (Credit\_rating)} = 0.048/0.9851$$

$$= 0.0487.$$

We see that highest Gain ratio is for age, followed by student & then followed by Credit_rating and then by income.



The order of attributes for splitting remained same for Information Gain and Gain Ratio, so the resulting tree looks similar.

Gini Index:

The attribute that provides smallest $gini_{split}(D)$ i.e (the largest reduction in impurity) is chosen to split the node.

- Gini Index is biased to multivalued attributes
- It has difficulty when number of classes is large.
- tends to favor tests that result in equal sized partitions and purity in both partitions.
- It forms a binary tree.

Example:

A1	A2	Class C
3.1	1.6	Yes

2.3	1.5	Yes
2.8	4.7	NO
3.1	4.6	NO
3.4	1.6	Yes
3.5	1.5	No.

Selecting random values for each attribute to calculate Gini Index.

for A1 A2

>=3 >=4

<3 <4

Calculating Gini Index for A1:

Attribute A1 >=3 & Class Yes: 2/6

A1 >=3 & Class No: 2/6

$$\text{Gini}(2,2) = 1 - \left[\left(\frac{2}{6}\right)^2 + \left(\frac{2}{6}\right)^2\right]$$

$$= 0.777$$

Attribute $A_1 < 3$ & class Yes : 1/6

$A_1 < 3$ & class No : 1/6

$$\text{Gini}(1,1) = 1 - \left[\left(\tfrac{1}{6}\right)^2 + \left(\tfrac{1}{6}\right)^2\right]$$

$$= 0.944$$

By adding weight and sum each of gini indices :

$$\text{gini}(\text{Target}, A_1) = \left(\frac{4}{16}\right) \times (0.777) + \left(\frac{2}{16}\right) \times (0.944)$$

$$= 0.194 + 0.118$$

$$= 0.312.$$

Calculating Gini Index for A2 :

Attribute $A_2 >= 4$ & Class Yes : 0/6

$A_2 >= 4$ & Class No : 2/6

$$\text{Gini}(0,2) = 1 - \left[\left(\tfrac{0}{6}\right)^2 + \left(\tfrac{2}{6}\right)^2\right]$$

$$= 0.888$$

Attribute $A_2 < 4$ & class Yes : 3/6

$A_2 < 4$ & class No : 1/6

$$Gini(3,1) = 1 - \left[\left(\frac{3}{6}\right)^2 + \left(\frac{1}{6}\right)^2\right]$$

$$= 0.722$$

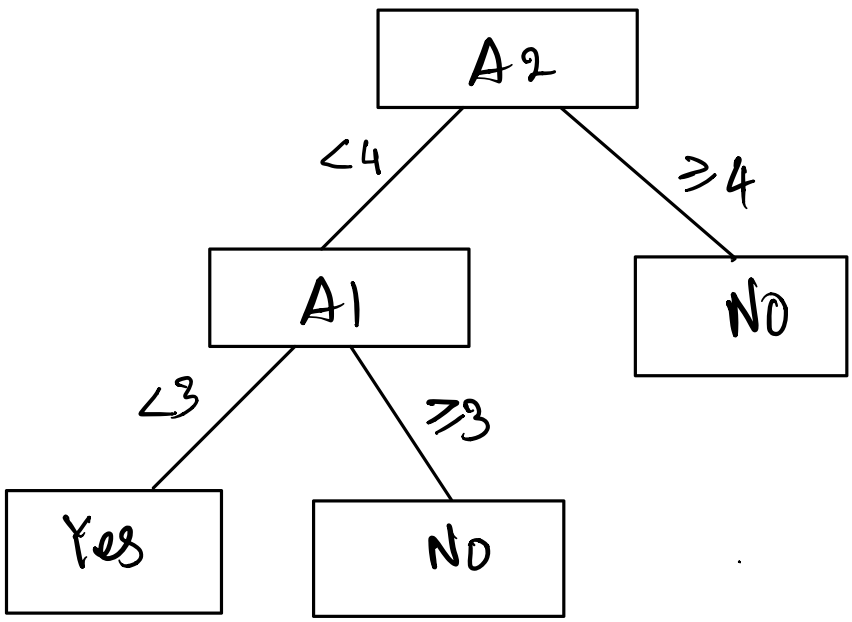
By adding weight and sum each of gini indices:

$$gini(Target, A_2) = \left(\frac{2}{16}\right) \times (0.888) + \left(\frac{4}{16}\right) \times (0.722)$$

$$= 0.111 + 0.1805$$

$$= 0.2915.$$

The least Gini Index is for A2 followed by A1.



Tree constructed using Gini Index for above data.

PART-2

PROBLEM-2

PART-2

We were asked to use this DB & implement repeated two fold cross validation holdout.

We were also given the folds for each repitition too & the no. of repititions given are 4.

We need to use a_1 as the root.

We also considered the class C as nominal.

O	a_1	a_2	C
O_1	1	1	1
O_2	0	0	0
O_3	0	1	0
O_4	0	0	0
O_5	1	1	1
O_6	1	1	0
O_7	0	0	0
O_8	1	0	1

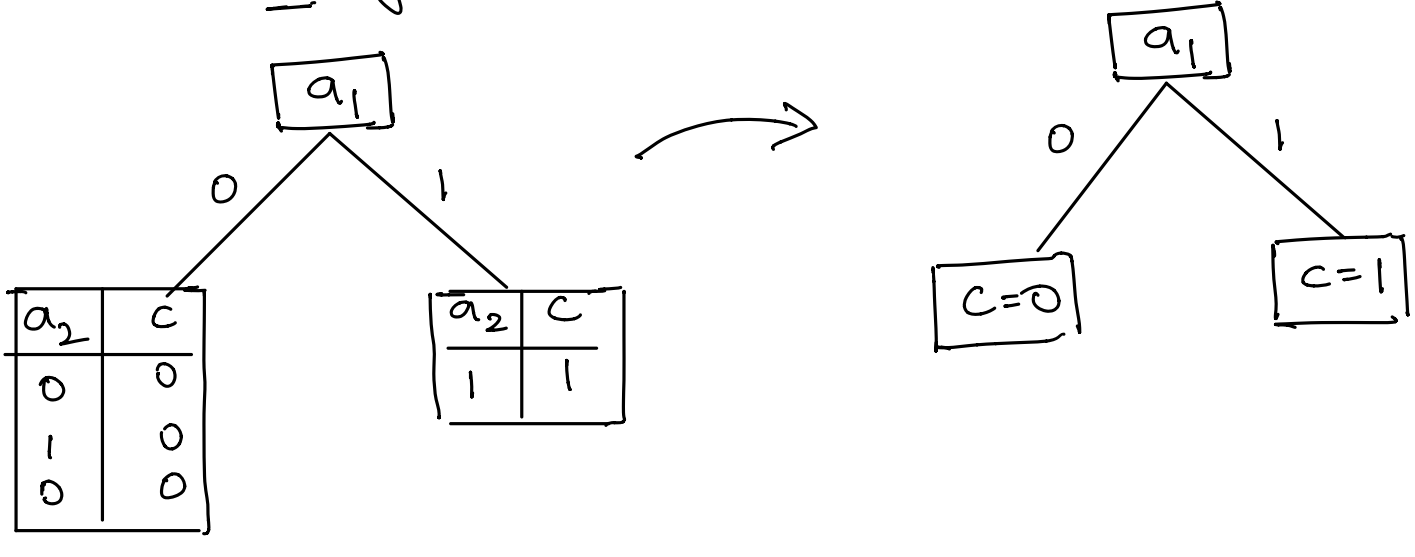
Step 1:

Repetition 1:

(1) train = $\{O_1, O_2, O_3, O_4\}$ test = $\{O_5, O_6, O_7, O_8\}$

So following the stages.

Training: (Stage 1)



As the decision tree is completed.

Discriminant rules :

Rule 1 : IF $a_1(x_1 = 0)$ THEN $c(x, = 0)$

Rule 2 : IF $a_1(x_1 = 1)$ THEN $c(x, = 1)$

We'll try to resubstitute as the part of Stage 2.

train : $\{O_1, O_2, O_3, O_4\}$

The rule accuracy can be calculated as : 100%

O_1 is classified well using Rule 2.

O_2, O_3, O_4 are also classified well using Rule 1.

So stage 3 :

test : $\{O_5, O_6, O_7, O_8\}$.

O_5, O_8 are classified well using Rule 2.

O_7 is classified well using Rule 1.

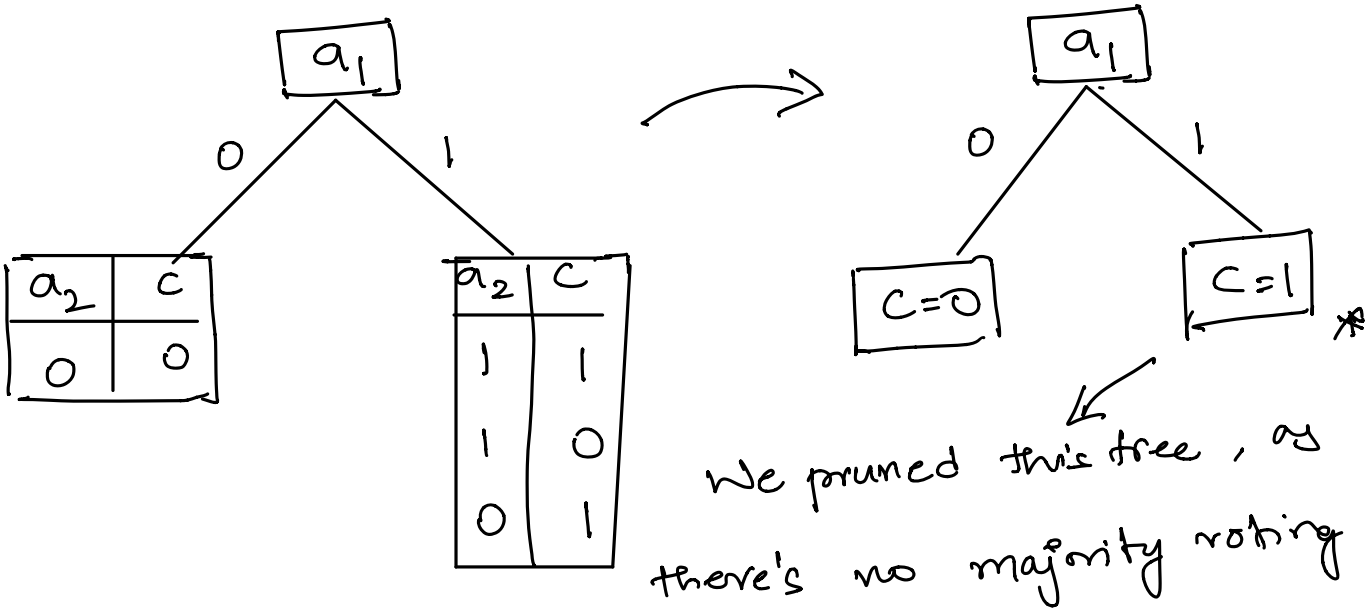
O_6 is misclassified from Rule 2.

So the predictive accuracy would be $3/4 = 75\%$.

(2) Now for the second fold as training set.

train = $\{O_5, O_6, O_7, O_8\}$ test = $\{O_1, O_2, O_3, O_4\}$

So following the stages.



We pruned this tree, as there's no majority voting present when $a_2 = 1$ $\{1, 0\}$ where as for $a_2 = 0$ $\{1\}$. So when $a_2 = 1$, we went forward by keeping c=1 or it is seen first.

As the decision tree is completed.

Discriminant rules:

Rule 1: IF $a_1(x_i = 0)$ THEN $c(x_i = 0)$

Rule 2: IF $a_1(x_i = 1)$ THEN $c(x_i = 1)$

We'll try to resubstitute as the part of stage 2.

train = $\{O_5, O_6, O_7, O_8\}$

The rule accuracy can be calculated as : 3/4 : 75%

O_5 is well classified using Rule 2

O_6 is misclassified from Rule 2.

O_7 is well classified using Rule 1.

O_8 is well classified using Rule 2.

So stage 3:

test : $\{O_1, O_2, O_3, O_4\}$.

O_1 is classified well using Rule 2.

O_2, O_3, O_4 are classified well using Rule 1.

so the predictive accuracy would be 100%.

So our classifier would be the union of these rules. F_1.

Rule 1 : IF $a_1(x_1=0)$ THEN $c(x_1=0)$

Rule 2 : IF $a_1(x_1=1)$ THEN $c(x_1=1)$

Rule 3 : IF $a_1(x_1=0)$ THEN $c(x_1=0)$

Rule 4 : IF $a_1(x_1=1)$ THEN $c(x_1=1)$

After removing the repetitions:

It would be

Rule 1 : IF $a_1(x_1=0)$ THEN $c(x_1=0)$

Rule 2: IF $a_1(x_1=1)$ THEN $c(x_1=1)$

The classifier's rule accuracy is $\frac{100+75}{2} = 87.5\%$.

& prediction accuracy is $\frac{100+75}{2} = 87.5\%$.

Repetition 2 :

If we clearly check the given fold is

test = $\{O_5, O_6, O_7, O_8\}$, test = $\{O_1, O_2, O_3, O_4\}$

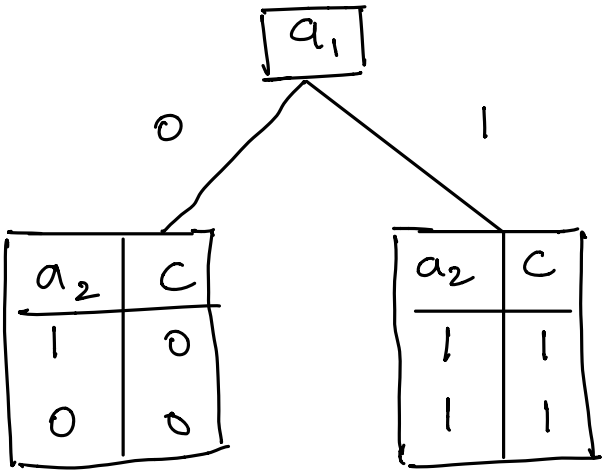
is exactly same as the 2nd fold in the previous repetition. So, similarly the 2nd fold in this's case would be the same as the 1st fold in previous scenario. so the classifier would be the same.

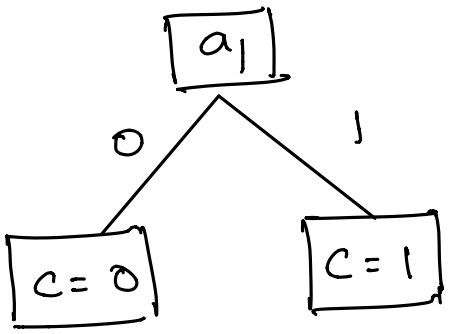
So $F_2 = F_1$

Repetition 3 :

(1) training = $\{O_1, O_3, O_5, O_7\}$ test = $\{O_2, O_4, O_6, O_8\}$.

Training: Stage 1 :





Discriminant rules:

Rule 1 : IF $a_1(x_i = 0)$ THEN $C(x_i = 0)$

Rule 2 : IF $a_1(x_i = 1)$ THEN $C(x_i = 1)$

So the rule accuracy is obtained by substituting the training data.

O_1 & O_5 are well classified using Rule 2.

O_3 & O_7 are well classified using Rule 1.

So the rule accuracy is 100%.

so during stage 3 for calculating predictive accuracy

O_2 & O_4 are well classified using Rule 1.

O_8 is well classified using Rule 2.

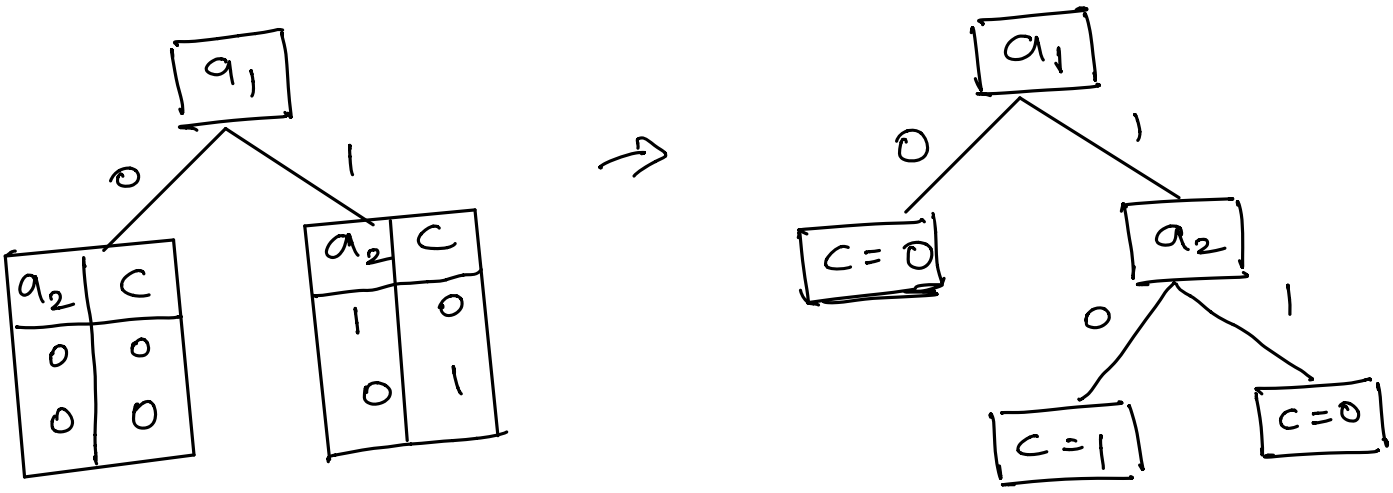
O_6 is misclassified from Rule 2.

so prediction accuracy = 75%.

(2)

training = $\{O_2, O_4, O_6, O_8\}$ test = $\{O_1, O_3, O_5, O_7\}$.

Training: Stage 1:



Discriminant Rules:

Rule 1: IF $a_1(x_1=0)$ THEN $c(x_1=0)$

Rule 2: IF $a_1(x_1=1)$ AND $a_2(x_2=0)$ THEN $c(x_1=1)$

Rule 3: IF $a_1(x_1=1)$ AND $a_2(x_1=1)$ THEN $c(x_1=0)$.

For stage 2 & resubstituting, the rule accuracy is

O_2 & O_4 are well classified using Rule 1.

O_6 is well classified using Rule 3.

O_8 is well classified using Rule 2.

So Rule Accuracy is 100%.

prediction Accuracy during stage 3 is

O_3 & O_7 are well classified using Rule 1.

O_1 & O_5 are misclassified from Rule 3.

So predictive accuracy is 50%.

So the combined rules of our classifier F_3 would be without repetitions.

Rule 1: IF $a_1(x_1=0)$ THEN $c(x_1=0)$

Rule 2: IF $a_1(x_1=1)$ AND $a_2(x_2=0)$ THEN $c(x_1=1)$

Rule 3: IF $a_1(x_1=1)$ AND $a_2(x_1=1)$ THEN $c(x_1=0)$.

Final Rule accuracy = $\frac{100+100}{2}$ = 100%.

Predictive accuracy would be = $\frac{50+75}{2}$

= 62.5%.

Repetition 4: The folds same as repetition 3 & hence the classifier would be the same too

$F_4 = F_3$

Step 2:

After calculating the predictive & rule accuracy in the previous step.

We know that $F_1 = F_2$ & $F_3 = F_4$.

So let's compare F_1 & F_3.

	Rules Accuracy	Predictive Accuracy
F_1	87.5%	87.5%
F_3	100%	62.5%

Based on these details, Rule Accuracy is almost similar for both these classifiers but the predictive accuracy of F_3 is very low compared to F_1. So I would pick F_1 or F_2 as my classifier F. Let it be F_1

So $F_1 = F$.

Step 3 :

We need to construct the bagged ensemble classifier F^*.

F^* contains F_1, F_2, F_3 & F_4. Since $F_1 = F_2$ & $F_3 = F_4$. The majority voting is calculated just between F_1 & F_3.

Given data.

O	a_1	a_2
o_1	0	1
o_2	0	0
o_3	1	0
o_4	1	1

Since our F is F_1

the rules of this classifier are :

F :

Rule 1 : IF $a_1(x_1 = 0)$ THEN $c(x_1 = 0)$

Rule 2 : IF $a_1(x_1 = 1)$ THEN $c(x_1 = 1)$

And our F^* consists of F_1, F_2 $(F_1 = F_2)$ & F_3, F_4 $(F_3 = F_4)$

f^* : F_1

Rule 1 : IF $a_1(x_1 = 0)$ THEN $c(x_1 = 0)$

Rule 2 : IF $a_1(x_1 = 1)$ THEN $c(x_1 = 1)$

F_3

Rule 1: IF $a_1(x_1=0)$ THEN $c(x_1=0)$

Rule 2: IF $a_1(x_1=1)$ AND $a_2(x_4=0)$ THEN $c(x_1=1)$

Rule 3: IF $a_1(x_1=1)$ AND $a_2(x_1=1)$ THEN $c(x_1=0)$.

Let's first classify using F.

O_1 is classified 0 using Rule 1.

O_2 is classified 0 using Rule 1.

O_3 is classified 1 using Rule 2.

O_4' is classified 1 using Rule 2.

for F^*

F_1 is classified similar to F as above.

where as F_3.

O_1 is classified 0 using Rule 1.

O_2 is classified 0 using Rule 1.

O_3 is classified 1 using Rule 2.

O_4 is classified 0 using Rule 3.

Since we need to take majority vote.

of F_1 & F_3 for F^{**}. The final result would be.

O_1 & O_2 are classified as 0.

O_3 & O_4 are classified as 1.

(O_4 doesn't have a majority vote, so we went with 1 as it's seen first).

	F	F^{**}
O_1	0	0
O_2	0	0
O_3	1	1
O_4	1	1

So based on our results, both these classifiers provided the same results.