

**Data Analytics with Python**  
**Prof. Ramesh Anbanandam**  
**Department of Management Studies**  
**Indian Institute of Technology - Roorkee**

**Lecture – 57**  
**Classification and Regression Trees(CART) - I**

In our previous lecture, we studied about different cluster and techniques, in this class we will start a new topic that is a classification and regression trees, shortly this is called CART models.

**(Refer Slide Time: 00:41)**

### Agenda

- Introduction to Classification and Regression Trees
- Attribute selection measures – Introduction

The agenda for this lecture is introduction to classification, regression trees, attribute selection measures and introduction. There are different measures for selecting attributes; attributes means variables that we will study about different attribute selection measures in this class.

**(Refer Slide Time: 00:55)**

---

## Introduction

- Classification is one form of data analysis that can be used to extract models describing important data classes or to predict future data trends
- Classification predicts categorical (discrete, unordered) labels whereas Regression analysis is a statistical methodology that is most often used for numeric (continuous) prediction
- For example, we can build a classification model to categorize bank loan applications as either safe or risky
- Regression model is used to predict expenditures in dollars of potential customers on computer equipment given their income and occupation

---

The introduction about this topic, CART model; classification is one form of data analysis that can be used to extract models describing important data classes or to predict future data trends. Classification predicts categorical that is discrete, unordered labels, whereas regression analysis is a statistical methodology that is most often used for numeric prediction. There is a difference between classification techniques and regressions.

In a classification techniques; the dependent variable is categorical variable most of the time but in regression analysis, most of the time the regression analysis, the continuous variable is the dependent variable for regression analysis. For example, we can build a classification model to categorize the bank loan applications as either safe or risky, see this is categorical. The regression model is used to protect expenditures in dollars of potential customers and computed equipment given their income and occupations. Most of the time the regression analysis used to predict a continuous variable but the classification analysis is used to predict the categorical variable.

**(Refer Slide Time: 02:16)**

## Problem Description for Illustration

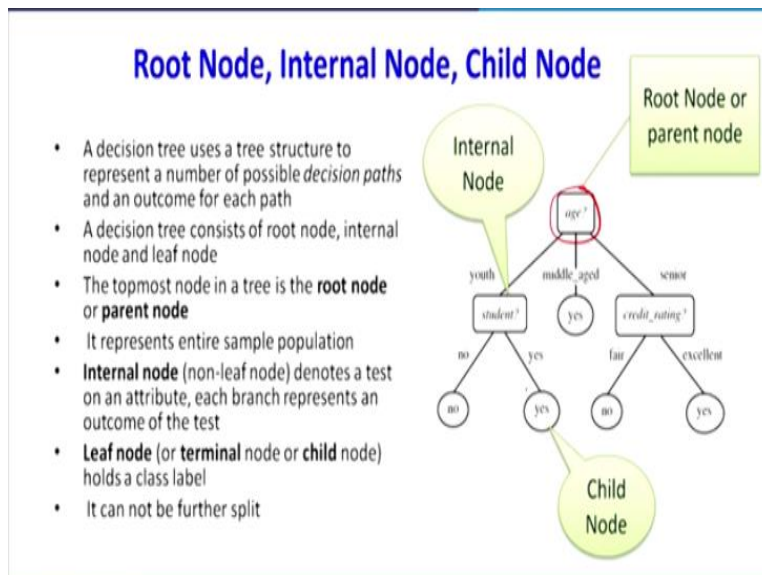
RID	age	income	student	credit_rating	Class: buys.computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	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

Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.

We are going to take one problem, this problem is taken from this book, Han, Pei and Kamber, data mining; book title is data mining concept and techniques. The problem says there are 1, 2, 3, 4, 5 columns, in the 5 columns there is age is there, income is there, student, credit rating, this dependent variable is buys computer; buys underscore computer, so this is a database, one portion of data base is shown.

So, the dependent variable is buys computer, there are 4 independent variables like age, income, student and credit rating. So by taking this example, we are going to explain how to use CART model, in coming lecture also we will use this data.

(Refer Slide Time: 03:04)



Now, let us understand certain terminology in the CART model, for example root node, internal node and child node, when you look at this picture, there is age; age there are 3 levels; youth, middle aged, senior, then it is a student there are 2 levels; no or yes. Credit rating; there are 2 levels; fair and excellent. So, a decision tree uses a tree structure to represent a number of possible decision paths and an outcome for each path.

The decision tree consist of root node, internal node and leaf node, you look at this the first one the age because the whole problem we are going to start with a variable age, so this age is called root node or parent node that is in the rectangular box. A decision tree consist of root node, internal node and leaf node, the top most node in a tree is called root node or parent node, this one for example age, this is a root node.

It represents entire sample population, the next term is internal node, for example here student is the internal node or non-leaf node, denotes a test on an attribute, each branch represents outcome of the test. The next node is leaf node or child node, see this yes or no that is which is in the elliptical shape that is called leaf node, it cannot be further split.

**(Refer Slide Time: 04:46)**

## Decision Tree Introduction

- A decision tree for the concept *buys\_computer*, indicating whether a customer at All Electronics is likely to purchase a computer
- Each internal (non-leaf) node represents a test on an attribute
- Each leaf node represents a class (either *buys\_computer = yes* or *buys computer = no*).

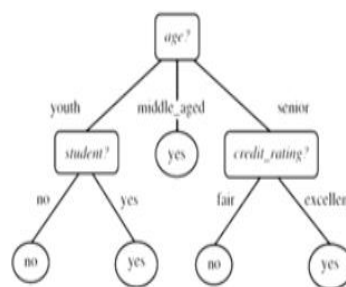


Figure 1.1 : Decision Tree

A decision tree for the concept buys computer is a variable indicating whether a customer at all electronics that is a database, where it was taken, a customer at all electronics is likely to purchase a computer, each internal node represents a test on attribute, each leaf node represents a

class, a class means either a person buys the computer yes or no, actually this yes or no is nothing but this column that we will go to into the child node.

**(Refer Slide Time: 05:21)**

---

## CART Introduction

- CART comes under supervised learning technique
- CART adopt a greedy(i.e., non backtracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner
- It is very interpretable model

---

Now, the CART comes under supervised learning techniques, we know that the machine learning techniques are classified into 2 categories; one is supervised, another one is unsupervised. What is the meaning of supervised learning is that there is a label; label in the sense we know in advance what is going to be independent variable, what is going to be dependent variable, in this problem also, it is supervised learning.

Because we know in advance what is the buys underscore computer is going to be our dependent variable, then CART adopts a greedy that is a non-backtracking approach in which decision trees are constructed in a top down recursive divide and conquer manner, it is very interpretable model. A person who was not having any statistical analysis also can easily interpret the CART model.

**(Refer Slide Time: 06:14)**

## Decision Tree Algorithm

Input:

- Data partition, D, which is a set of training tuples and their associated class labels;
- Attribute list, the set of candidate attributes;
- Attribute selection method, a procedure to determine the splitting criterion that “best” partitions the data tuples into individual classes. This criterion consists of a splitting attribute and, possibly, either a split point or splitting subset.

RID	age	income	student	credit rating	Class: buys computer
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

Output: A decision tree

Now, I will explain the decision tree algorithm, what are the inputs data partition D, this whole data set is a set of training tuples and their associated class labels, this is the class labels. Attribute list is the set of candidate attribute, for example in this problem these are the attributes; age, income, student, credit rating. Now, what you have to do before starting the problem, out of these 4 variables, we have to decide from which variable we have to start for classification.

So, for that we need a attribute selection method, so attribute selection method a procedure to determine the splitting criterion that best partitions the data tuples into individual classes, this criterion consist of a splitting the attribute and possibly, either a split point on splitting subset. So, output of this model will be the decision tree.

**(Refer Slide Time: 07:16)**

---

## Decision Tree Algorithm

- The algorithm is called with three parameters: D, attribute list, and Attribute selection method
- D is defined as a data partition. Initially, it is the complete set of training tuples and their associated class labels
- The parameter attribute list is a list of attributes or independent variables which are describing the tuples
- Attribute selection method specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class

---

Decision tree algorithm; the algorithm is called with the 3 parameters; one is D that is a data set, then is attribute list independent variable and attribute selection methods. D is defined as data partition; initially it is the complete set of training tuples and their associated class labels. In our problem, this whole table represents the D, initially what is that it covers all the independent variables and dependent variables.

The parameter attribute list is a list of attributes or independent variables which are describing the tuples. Here the parameter attributes; attributes nothing but all the independent variables, so attribute selection method specify a heuristic procedure for selecting the attribute that best discriminates the given tuples according to class.

**(Refer Slide Time: 08:12)**

## Decision Tree Algorithm



- This procedure employs an attribute selection measure, such as information gain, gain ratio or the Gini index.
- Whether the tree is strictly binary is generally driven by the attribute selection measure
- Some attribute selection measures, such as the Gini index, enforce the resulting tree to be binary. Others, like information gain, do not, therein allowing multiway splits (i.e., two or more branches to be grown from a node).

---

This procedure employs an attribute selection measures such as information gain, that is a one method for selecting the attribute, second one method is called gain ratio, third method is called Gini index, whether the tree is strictly binary is generally driven by the attribute selection measures. For example, we can go for binary selection suppose, this is one variable, sometime we can go for more than 2 classifications also.

For example, if you use Gini method that will cover in coming class that always you need to go for binary selection, some attribute selection measures such as Gini index enforce the resulting tree to be binary, others like information gain do not, therein allowing multiway splits. So, if you follow Gini index, there is only binary split, if you follow other than Gini index for example, information gain, you can have more than 2 split also.

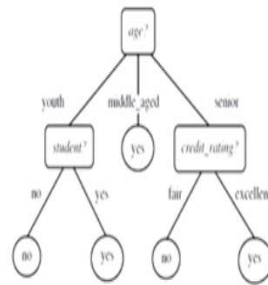
**(Refer Slide Time: 09:11)**



## Decision Tree Method

Method:

- (1) create a node  $N$ ;
- (2) if tuples in  $D$  are all of the same class,  $C$  then
- (3) return  $N$  as a leaf node labeled with the class  $C$ ;
- (4) if  $attribute\_list$  is empty then
- (5) return  $N$  as a leaf node labeled with the majority class in  $D$ ; // majority voting
- (6) apply  $Attribute\_selection\_method(D, attribute\_list)$  to find the "best"  $splitting\_criterion$ ;
- (7) label node  $N$  with  $splitting\_criterion$ ;
- (8) if  $splitting\_attribute$  is discrete-valued and  
multiway splits allowed then // not restricted to binary trees
- (9)  $attribute\_list \leftarrow attribute\_list - splitting\_attribute$ ; // remove  $splitting\_attribute$
- (10) for each outcome  $j$  of  $splitting\_criterion$   
// partition the tuples and grow subtrees for each partition
- (11) let  $D_j$  be the set of data tuples in  $D$  satisfying outcome  $j$ ; // a partition
- (12) if  $D_j$  is empty then
- (13) attach a leaf labeled with the majority class in  $D$  to node  $N$ ;
- (14) else attach the node returned by  $Generate\_decision\_tree(D_j, attribute\_list)$  to node  $N$ ;
- endfor
- (15) return  $N$ ;



N-Node

C- Class

D- tuples in training data set

Decision tree method; I am going to explain different steps in the decision tree method, there are 15 steps in coming slides, I will explained in each steps. So, first we will start the overview of all 15 steps. Create a node  $N$ , if tuples in  $D$  all are of the same class  $C$ , then return  $N$  as the leaf node labelled with the class  $C$ , if attribute list is empty then return  $N$  as a leaf node labelled with majority class in  $D$  by using the concept called majority voting.

Then apply attribute selection method  $D$  to find the best splitting criterion, label node  $N$  with the splitting criterion, if splitting attribute is discrete valued and multiway splits is allowed, then attribute list, there are different attribute list that we can choose for example, splitting attribute is one method. The step 8 is if splitting attribute is discrete valued and multiway split is allowed, then the attribute list which already have occurred that has to be removed from the our  $D$ .

For each outcome  $j$  for splitting criterion, so what is a splitting criterion is partition of the tuples and grow sub tree for each partition. Let  $D_j$  be the set of data tuples in  $D$  satisfying the outcome  $j$ , if  $D_j$  is empty then attach a leaf labelled with the majority class in  $D$  to node  $N$  else attach node returned by generate decision tree to node  $N$ , then  $N$  for return  $N$ . So, I am going to explain that each steps in detail in coming slides.

**(Refer Slide Time: 11:05)**

## Decision Tree Method step 1 to 6

- The tree starts as a single node, N, representing the training tuples in D (step 1).
- If the tuples in D are all of the same class, then node N becomes a leaf and is labelled with that class (steps 2 and 3)
- Steps 4 and 5 are terminating conditions
- Otherwise, the algorithm calls Attribute selection method to determine the splitting criterion
- The splitting criterion (like Gini) tells us which attribute to test at node N by determining the "best" way to separate or partition the tuples in D into individual classes (step 6)

### Method:

- (1) create a node N;
- (2) if tuples in D are all of the same class, C then
- (3) return N as a leaf node labeled with the class C;
- (4) if attribute\_list is empty then
- (5) return N as a leaf node labeled with the majority class in D; // majority voting
- (6) apply Attribute\_selection\_method(D, attribute\_list) to find the "best" splitting criterion;
- (7) label node N with splitting\_criterion;
- (8) if splitting\_attribute is discrete-valued and  
multisway splits allowed then // not restricted to binary trees
- (9) attribute\_list ← attribute\_list - splitting\_attribute; // remove splitting attribute
- (10) for each outcome j of splitting\_criterion
- // partition the tuples and grow subtrees for each partition
- (11) let D<sub>j</sub> be the set of data tuples in D satisfying outcome j; // a partition
- (12) if D<sub>j</sub> is empty then
- (13) attach a leaf labeled with the majority class in D to node N;
- (14) else attach the node returned by Generate\_decision\_tree(D<sub>j</sub>, attribute\_list) to node N;
- endif
- (15) return N;

The tree starts as a single node N representing training tuples in D that was our step 1, if the tuples in D are all of the same class, then N becomes a leaf and is labelled with that class that is step 2 and 3. If the tuples in D, all of the same class C, then return N as a leaf node labelled with the class C, the meaning is, suppose the age is taken as variable, if there are 3 split; one is youth, middle-aged and senior.

For example, the middle-aged with respect to our dependent variable all are answered yes, so if it all are answered yes, we need not go for further classification, then the age attribute has to be dropped from our model, then we have continue with the remaining attributes like student, credit rating and income. So, step 4 and 5 are terminating conditions, if attribute list is empty that means, you have to go for each attributes otherwise, the algorithm calls attribute selection method to determine splitting criterion.

Suppose, only 1 attribute is there, if there are remaining attributes, to choose that attribute, you have to use attribute selection method to determine splitting criterion. The splitting criterion like Gini tells us which attribute to test at node N by determining the best way to separate or partition the tuples in D into individual classes.

**(Refer Slide Time: 12:44)**

## Decision Tree Method - Step 7 - 11

- The splitting criterion indicates the splitting attribute and may also indicate either a split-point or a splitting subset
- The splitting criterion is determined so that, ideally, the resulting partitions at each branch are as "pure" as possible. A partition is pure if all of the tuples in it belong to the same class.
- The node N is labelled with the splitting criterion, which serves as a test at the node (step 7).
- A branch is grown from node N for each of the outcomes of the splitting criterion.
- The tuples in D are partitioned accordingly (steps 10 to 11)

### Method

```

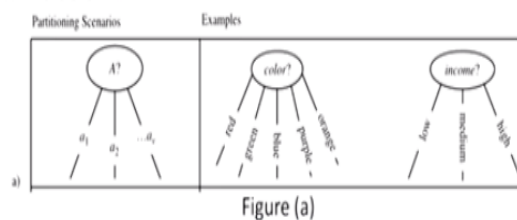
(1) create a node N;
(2) if tuples in D are all of the same class, C then
(3)   return N as a leaf node labeled with the class C;
(4) if attribute list is empty then
(5)   return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply Attribute selection method(D, attribute list) to find the "best" splitting criterion;
(7) label node N with splitting criterion;
(8) if splitting attribute is discrete valued and
    multway splits allowed then // not restricted to binary trees
(9)   attribute list ← attribute list - splitting attribute; // remove splitting attribute
(10) for each outcome j of splitting criterion
    - // partition the tuples and grow subtrees for each partition
(11)   let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)   if Dj is empty then
(13)     attach a leaf labeled with the majority class in D to node N;
(14)   else attach the node returned by Generate decision tree(Dj, attribute list) to node N;
endfor
(15) return N;
  
```

The splitting criterion indicates the splitting attributes and may also indicate either a split pointer or splitting subset; I will explain, what is the meaning of split point and splitting subset in next slide. The splitting criterion is determined, so that ideally the resulting partitions at each branch are as a pure as possible. A partition pure if all of the tuples in it belongs to the same class, the node N is labelled with the splitting criterion which serve as a test at the node that is a step 7. A branch is grown from node N for each of the outcomes for splitting criterion, the tuples in D are partitions accordingly that is our step in 11.

(Refer Slide Time: 13:30)

### Three possibilities for partitioning tuples based on the splitting criterion

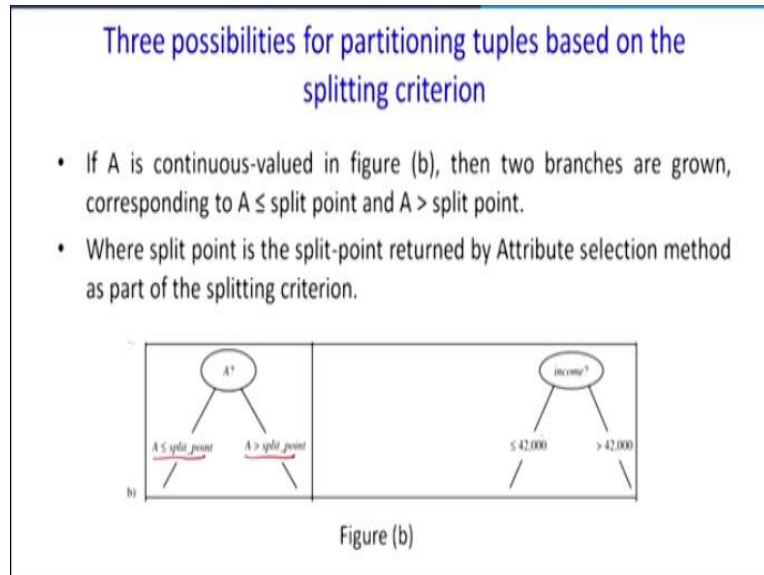
- There are three possible scenarios, as illustrated in Figure (a), (b) and (c).
- Let A be the splitting attribute. A has 'v' distinct values, {a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>v</sub>}, based on the training data
- If A is discrete-valued in figure (a), then one branch is grown for each known value of A.



So, 3 possibilities for partitioning tuples based on the splitting criterion, there are 3 possible scenarios as illustrated in figure a, b and c. Let A be the splitting attributes, A has v distinct

values  $a_1$ ; see  $a_2$ ,  $a_2$  and  $a_v$  based on training data. If  $A$  is discrete valued in figure a, then one branch is grown for each known value of  $A$ . See for example colour; may be red, green, blue, purple, orange, if it is income, there are 3 split; low, medium, high.

**(Refer Slide Time: 14:13)**



If  $A$  is a continuous valued in figure b, then 2 branches are grown corresponding to  $A$  less than or equal to split point and  $A$  greater than or equal to split point. So, what will happen; the  $A$  less than or equal to split point is the one split,  $A$  greater than split point is another branch, where the split point is the split point returned by attribute selection method as part of the splitting criterion.

For example, income is there, we can group that into 2 categories, those who have incomes are below 42,000, those who have incomes are above 42,000; this 42,000 generally is nothing but the average value.

**(Refer Slide Time: 14:54)**

### Three possibilities for partitioning tuples based on the splitting criterion

- If A is discrete-valued and a binary tree must be produced, then the test is of the form  $A \in S_A$ , where  $S_A$  is the splitting subset for A.

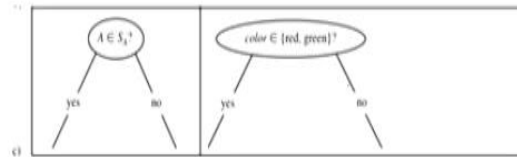


Figure (c)

If A is a discrete valued and binary tree must be produced, then the test is of the form A belongs to  $S_A$ , where  $S_A$  is the splitting subset of A, so is it that A belongs to  $S_A$ , if it is yes is one group, no if it is another group. In the; if A, for example colour it may be red or green, then that time also, it should be yes or no.

**(Refer Slide Time: 15:18)**

### Decision Tree Method – termination condition

- The algorithm uses the same process recursively to form a decision tree for the tuples at each resulting partition,  $D_j$ , of D (step 14).
- The recursive partitioning stops only when anyone of the following terminating conditions is true:
  1. All of the tuples in partition D (represented at node N) belong to the same class (steps 2 and 3), or

Then, we will go for termination condition; the algorithm uses the same process recursively to form a decision tree for the tuples at each resulting partition  $D_j$  of D, what is the recursion means; if one attribute is over that is repeated for the second attribute and third attribute up to all the attributes are exhausted. The recursive partitioning stops only when any one of the following

terminating condition is true. The first condition is all of the tuples in partition  $D$  representing at node  $N$  belong to the same class that was our same step 2 and 3.

**(Refer Slide Time: 15:58)**

### Decision Tree Method – termination condition

2. There are no remaining attributes on which the tuples may be further partitioned (step 4).
  - In this case, majority voting is employed (step 5).
  - This involves converting node  $N$  into a leaf and labelling it with the most common class in  $D$ .
  - Alternatively, the class distribution of the node tuples may be stored.
3. There are no tuples for a given branch, that is, a partition  $D_j$  is empty (step 12).
  - In this case, a leaf is created with the majority class in  $D$  (step 13).
  - The resulting decision tree is returned (step 15).

Or there are no remaining attribute on which the tuples maybe further partitioned that is in step 4, in this case majority voting is employed, this involves converting a node into a leaf and labelling it with the most common class in  $D$ , alternatively the class distribution of the node tuples may be stored. The third condition is there are no tuples for a given branch that is a partition  $D_j$  is empty that is explained in step 12. In this case, a leaf is created with the majority class in  $D$  that is your step 13, the resulting decision tree is returned that is our step 15.

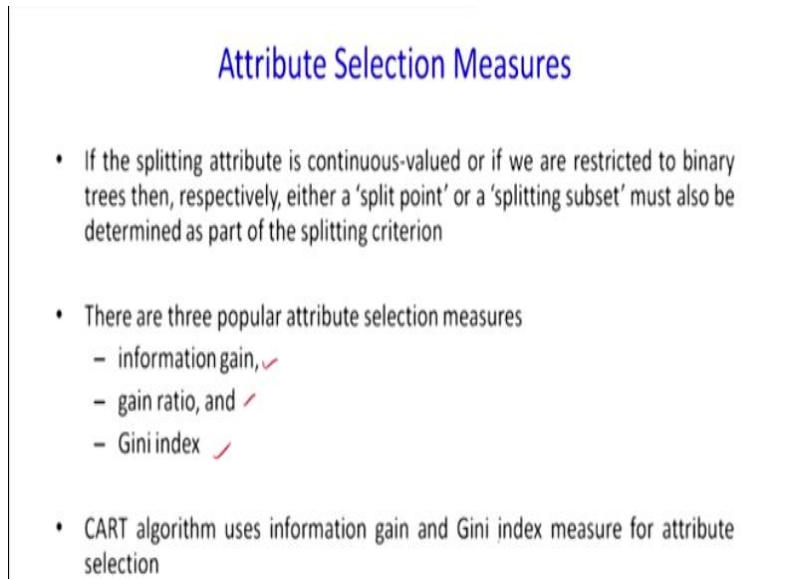
**(Refer Slide Time: 16:42)**

### Attribute Selection Measures

- Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split
  - It is a heuristic approach for selecting the splitting criterion that “best” separates a given data partition,  $D$ , of class-labeled training tuples into individual classes
  - The attribute selection measure provides a ranking for each attribute describing the given training tuples
  - The attribute having the best score for the measure is chosen as the splitting attribute for the given tuples
-

Now, the second part of the selection is different attribute selection measures; attribute selection measures are also known as splitting rules because they determine how the tuples at a given node to be split, it is a heuristic approach for selecting the splitting criterion that best separates a given data partition D of class labelled training tuples into individual classes. The attribute selection measures provide a ranking for each attributes describing the given training tuples. The attributes having the best score for the measure is chosen as the splitting attribute for the given tuples.

**(Refer Slide Time: 17:29)**



The slide is titled "Attribute Selection Measures" in blue text. It contains four bullet points:

- If the splitting attribute is continuous-valued or if we are restricted to binary trees then, respectively, either a 'split point' or a 'splitting subset' must also be determined as part of the splitting criterion
- There are three popular attribute selection measures
  - information gain, ✓
  - gain ratio, and ✓
  - Gini index ✓
- CART algorithm uses information gain and Gini index measure for attribute selection

If the splitting attribute is continuous valued or if we are restricted to binary trees, then respectively, either a split point or split subset must also be determined as part of the splitting criterion. There are 3 popular attribute selection measures; one is information gain, gain ratio, Gini index. In this class, I am going to explain the theory about this 3 attribute measures, in coming classes by using the same examples which I have discussed, I am going to find out the value of information gain.

I am going to explain the selection procedures using the criteria information gain, gain ratio and Gini index. So, in this lecture this we are going to see the theoretical point of all these 3 selection methods. So, CART algorithm uses information gain and Gini index measures for attribute selection.

**(Refer Slide Time: 18:30)**



## Attribute Selection Measures

The notation used herein is as follows. Let  $D$ , the data partition, be a training set of class-labeled tuples. Suppose the class label attribute has  $m$  distinct values defining  $m$  distinct classes,  $C_i$  (for  $i = 1, \dots, m$ ). Let  $C_{i,D}$  be the set of tuples of class  $C_i$  in  $D$ . Let  $|D|$  and  $|C_{i,D}|$  denote the number of tuples in  $D$  and  $C_{i,D}$ , respectively.

RID	age	income	student	credit rating	Class	boys	computer
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		

$m = 2$

Attribute selection measures let us find out certain notations, the notation used herein is as follows. Let  $D$ , the data partitions, be a training set of class labelled tuples, for example this dataset suppose, the class label attribute as  $m$  distinct values, here  $m$  is there are; this is a class 1, here the value of  $m$  is 2 because yes is 1 category, no is another category, distinct class in  $C_i$ . The  $C_i$  is it is 1 to  $m$ , it may be 1 and another may be it is 2.

Let  $C_{i,D}$  be the set of tuples of class  $C_{i,D}$  for example, what is the  $C_{i,D}$  means, for example if it is a high for this income variable, in high how many no is there; 1, 2, high, it is a 2, for income one level is called high for that it is no, no, so that is our  $C_{i,D}$ ;  $C_{i,D}$ , set of tuples of class  $C_i$  in  $D$ . So, this modulus of  $D$  represents that the 14  $C_{i,D}$  represents how many number of values if it is high and what is no. If it is for example, if you say low, this variable; income variable low, how many yes is there; 1, low, 2, low, 3, so 3, the modulus of  $C_{i,D}$  is 3, so this values I have explained in coming lectures with an example.

**(Refer Slide Time: 20:11)**



## Information Gain

- This measure studied the value or "information content" of messages
- The attribute with the highest information gain is chosen as the splitting attribute for node
- This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions
- This approach minimizes the expected number of tests needed to classify a given tuple

Then, we will go to the first criteria for selecting the attributes information gain, this measure studied the value or information content of messages, the attribute with the highest information gain is chosen as the splitting attribute for node, this attribute minimises the information needed to classify the tuples in the resulting partitions and reflect the least randomness or impurity in these partitions.

(Refer Slide Time: 20:49)

### Information Gain-Entropy Measure

- The expected information needed to classify a tuple in  $D$  is given by

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

- Where  $p_i$  is the probability that an arbitrary tuple in  $D$  belongs to class  $C_i$  and is estimated by  $|C_{i,D}|/|D|$ .
- A log function to the base 2 is used, because the information is encoded in bits
- Info(D) (or **Entropy** of  $D$ ) is just the **average amount of information needed** to identify the class label of a tuple in  $D$

RID	age	income	student	credit rating	Class: buys computer
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	no
14	senior	medium	no	excellent	no

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

So, this approach minimises the expected number of test needed to classify a given tuple, so information gain that is nothing but entropy measure, the expected information needed to classify a tuple in  $D$  is given by Info  $D$  equal to minus summation  $I$  equal to 1 to  $m$   $p_i \log p_i$  to the base 2,

where  $p_i$  is a probability that an arbitrary tuple in  $D$  belongs to class  $C_i$  and is estimated by modulus of  $C_i D$  divided by modulus of  $D$ .

For example, in this for this dataset, what is a  $p_i$ ;  $p_i$  is the number of yes, how many number of yes is there? 1, 2, 3, 4, 5, 6, 7, 8, 9, so it is 9, that 9 is nothing but your  $C_i D$ , modulus of  $D$  is total 14 that is for level 1. For the level 2, how many no is there? 1, 2, 3, 4, 5, so 5 divided by 14  $\log_2 5$  divided by 14 to the base 2 equal to 0.940 bits, so this is the meaning of our Info  $D$ . A log function to the base 2 is used because the information is encoded in bits.

So, Info  $D$  or entropy, another name for entropy is just the average amount of information needed to identify the class label of a tuple in  $D$ . Generally, the lesser the value of entropy that means we need very less informations to identify the class label of a tuple  $D$ , so generally the value of entropy should be less, so that attribute will be chosen for classification.

**(Refer Slide Time: 22:44)**

### Attribute Selection Measures

- It is quite likely that the partitions will be impure (e.g., where a partition may contain a collection of tuples from different classes rather than from a single class).
- How much more information would we still need (after the partitioning) in order to arrive at an exact classification?
- This amount is measured by 
$$\text{Info}_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$
- The term  $|D_j| / |D|$  acts as the weight of the  $j_{th}$  partition.  $\text{Info}_A(D)$  is the expected information required to classify a tuple from  $D$  based on the partitioning by  $A$ .

It is quite likely that the partitions will be impure, where a partition may contain a collection of tuples from different classes rather than from the single class, so how much more information would still need, after the partition in order to arrive an exact classification. So, this amount is measured by Info  $D$  for one attribute that is for summation  $j$  equal to 1 to  $v$  for all the splits modulus of  $D_j$  divided by modulus of  $D$  multiplied by Info  $D_j$ , this is nothing but your entropy.

The term  $D_j$ ; modulus  $D_j$  divided by modulus  $D$  act as a weight of  $j$ th operation,  $Info\ D$  for attribute  $A$  is expected information required this one, expected information required to classify a tuple from  $D$  based on the partitioning by  $A$ .

**(Refer Slide Time: 23:51)**

## Information Gain

- The smaller the expected information (still) required, the greater the purity of the partitions
- Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on  $A$ ). That is,

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

- The attribute  $A$  with the **highest information gain**, ( $Gain(A)$ ), is chosen as the splitting attribute at node  $N$ .

So, the information gain; the smaller the expected information required the greater the purity of the partitions, so information gain is defined as the difference between the original information requirement that is done by based on just proportion of classes and the new requirement that is obtained after partitioning on  $A$ . So, the gain  $A$  is  $Info\ D$  minus  $Info\ D$  for attribute  $A$ , I have used this example in my coming classes with the help of numerical example, I have explain how to find out the gain  $A$ .

The attribute  $A$  with the highest information gain is chosen as a splitting attribute at node  $N$ , you see that the entropy should be very smaller but the information gain should be higher for choosing an attribute.

**(Refer Slide Time: 24:46)**

## Gini Index

- Gini index is used to measure the impurity of D, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2$$

- Where  $p_i$  is the probability that a tuple in D belongs to class  $C_i$  and is estimated by  $|C_{i,D}|/|D|$ .
- The sum is computed over 'm' classes.
- The Gini index considers a binary split for each attribute

RID	age	income	student	credit rating	Class: buys computer
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

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

Next we will go to the next concept Gini index; Gini index is used to measure the impurity of D, the data partition or set of training tuples, the formula for Gini D is  $1 - \sum_{i=1}^m p_i^2$ , where  $p_i$  is the probability that tuples in D belongs to a class  $C_i$  and is estimated by modulus of  $C_i/D$  divided by D, for example I will explain for this dataset, how to find out Gini index.

So, 1 minus; so how many yes is there here, it is 9 yes is there, so 9 by 14 whole square minus, how many no is there; 5 no is there; 5 divided by 14 whole square, so  $1 - \frac{9}{14} - \frac{5}{14}$  whole square equal to 0.459, this is Gini index because the sum is computed over m classes, we are doing for all the for m equal to 1, m equal to 2, the Gini index considers a binary split for each attribute. So, here we are going to get only binary split if you use Gini index.

**(Refer Slide Time: 26:02)**

## Gini Index

- When considering a binary split, we compute a weighted sum of the impurity of each resulting partition
- For example, if a binary split on A partitions D into  $D_1$  and  $D_2$ , the Gini index of D given that partitioning is-

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

- For each attribute, each of the possible binary splits is considered
- For a discrete-valued attribute, the subset that gives the minimum Gini index for that attribute is selected as its splitting subset

When considering a binary split, we compute a weighted sum of impurity of each resulting partitions for example, if a binary split on a partitions D into 2 category; one is D1, D2, then the Gini index of D given that partitioning is; so Gini D for each attribute modulus of D1 divided by D Gini of D1 + modulus of D2 divided by modulus of D Gini of D2. So, in my coming lectures I have used this formula also with the help of a numerical example.

There you can have very clear understanding how we are finding this, for each attribute each of the possible binary split is considered, for a discrete valued attribute the subset that gives the minimum Gini index, you have to remember this, minimum Gini index for that attribute is selected as its splitting subset.

**(Refer Slide Time: 27:02)**

## Gini Index

- For continuous-valued attributes, each possible split-point must be considered
- The strategy is similar where the midpoint between each pair of (sorted) adjacent values is taken as a possible split-point.
- For a possible split-point of A,  $D_1$  is the set of tuples in D satisfying  $A \leq$  split point, and  $D_2$  is the set of tuples in D satisfying  $A >$  split point.
- The reduction in impurity that would be incurred by a binary split on a discrete- or continuous-valued attribute A is

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

- The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute

---

For continuous valued attributes, each possible split point must be considered; the strategy is similar where the midpoint between each of the pair, adjacent value is taken as a possible split point. If there is a continuous variable, the midpoint in a sorted dataset, the midpoint should be taken as the splitting criteria. For a possible split of A,  $D_1$  is the set of tuples in D satisfying A less than or equal to split point.

And  $D_2$  is the set of tuples in D satisfying A greater than split point, the reduction in impurity that would be incurred by a binary split on a discrete or continuous valued attribute A is delta of Gini A equal to Gini D, this is for our class variable minus Gini D for a particular attributes. The attribute that maximises the reduction in impurity has the otherwise, which is having minimum Gini index is selected for splitting attribute. So, this value should be maximum otherwise, this will be maximum only if the Gini index is minimum.

**(Refer Slide Time: 28:16)**

## Which attribute selection measure is the best?

- All measures have some bias.
- The time complexity of decision tree generally increases exponentially with tree height
- Hence, measures that tend to produce shallower trees (e.g., with multiway rather than binary splits, and that favour more balanced splits) may be preferred.
- However, some studies have found that shallow trees tend to have a large number of leaves and higher error rates
- Several comparative studies suggests no one attribute selection measure has been found to be significantly superior to others.

So, we have seen 2 methods; one method is information gain, another method is Gini index, there is one more method is called gain ratio that I will take in my coming classes. So, how to choose which attribute method has to be chosen, all measures have some bias for example, this technic information gain also having some bias that I will explain in coming class. The time complexity of decision tree generally increases exponentially with the tree height.

Hence, measures that tend to produce shallower trees that is with multiway rather than binary split and that favour more balanced split may be preferred that is why most of the time Gini index is chosen because that is giving a balance split however, some studies have found that shallow trees tend to have a large number of leaves and higher error rates, several comparative studies suggest no one attribute selection measures has been found to be significantly superior to others.

**(Refer Slide Time: 29:24)**

## Tree Pruning

- When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers
- Tree pruning use statistical measures to remove the least reliable branches
- Pruned trees tend to be smaller and less complex and, thus, easier to comprehend
- They are usually faster and better at correctly classifying independent test data than unpruned trees

Next, we will go the concept called tree pruning, when a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. So, tree pruning use statistical measures to remove the least branches, pruned trees tend to be smaller and less complex and thus easier to comprehend, they are usually faster and better at correctly classifying independent test data than unpruned trees.

**(Refer Slide Time: 30:01)**

## How does Tree Pruning Work?

- There are two common approaches to tree pruning: **pre-pruning** and **post-pruning**.
- In the **pre-pruning** approach, a tree is “pruned” by halting its construction early (e.g., by deciding not to further split or partition the subset of training tuples at a given node).
- When constructing a tree, measures such as statistical significance, information gain, Gini index can be used to assess the goodness of a split.

How does tree pruning work; there are 2 common approaches to tree pruning; one is pre-pruning, another one is post-pruning. In the pre-pruning approach, the tree is pruned by halting its construction early that is by deciding not to further split or partition the subset of training



tuples at a given node, when constructing a tree measures such as statistical significance, information gain, Gini index can be used to assess the goodness of a split.

**(Refer Slide Time: 30:34)**

### How does Tree Pruning Work?

- The **post-pruning** approach removes sub\_trees from a “fully grown” tree
- A subtree at a given node is pruned by removing its branches and replacing it with a leaf
- The leaf is labelled with the most frequent class among the subtree being replaced
- For example, the subtree at node “A3?” in the unpruned tree of Figure 1.2
- The most common class within this subtree is “class B”
- In the pruned version of the tree, the subtree in question is pruned by replacing it with the leaf “class B”

Now, let us talk about the post-pruning; the post pruning approach remove the sub tree from a fully grown tree, a sub tree at a given node is pruned by removing its branches and replacing it with a leaf, so the leaf is labelled with the most frequent class among the sub tree being replaced.

**(Refer Slide Time: 30:55)**

### How does Tree Pruning Work?

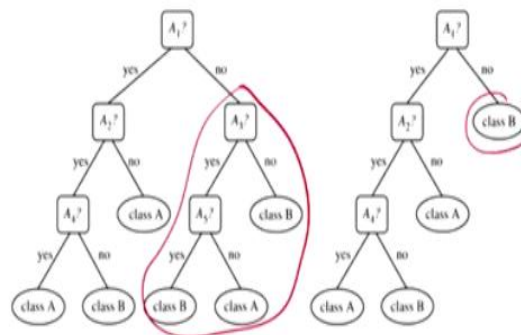


Figure: 1.2 An unpruned decision tree and a post-pruned decision tree

Look at this picture which is given in the next slide, assume that we are going to remove this portions, what is happening; in this when you look at this one, if you are removing this portion of the tree, the class B is frequently occurring, so we have to bring as a leaf node, in that leaf node,

the class B has to be retained. For example, the subtree at the node A3 in the unpruned tree is shown in figure 1.2.

The most common class within the subtree is class B, look at this there are class B is there, class B is there, class A is there, 1 class A is there, 2 class B is there, so the most common class with this subclass is class B. The pruned version of the tree, the subtree in the question in is pruned by replacing with the leaf class B, you see that this is the pruned version, in that we have retain class B. This figure explains unpruned decision tree and the post pruned decision tree.

What you have done in this lecture; I have introduced what is classification regression tree CART model, then I have explained different terminology, which are frequently used in the CART model, then I have explained the theory behind different attribute selection measures like information gain, Gini index. At the end, I have explained how to do the pruning of the tree. In the next class with the help of a numerical example, I will explain how to do or how to select different attributes. For example, with the help of information how to choose attributes; correct attributes, thank you very much.