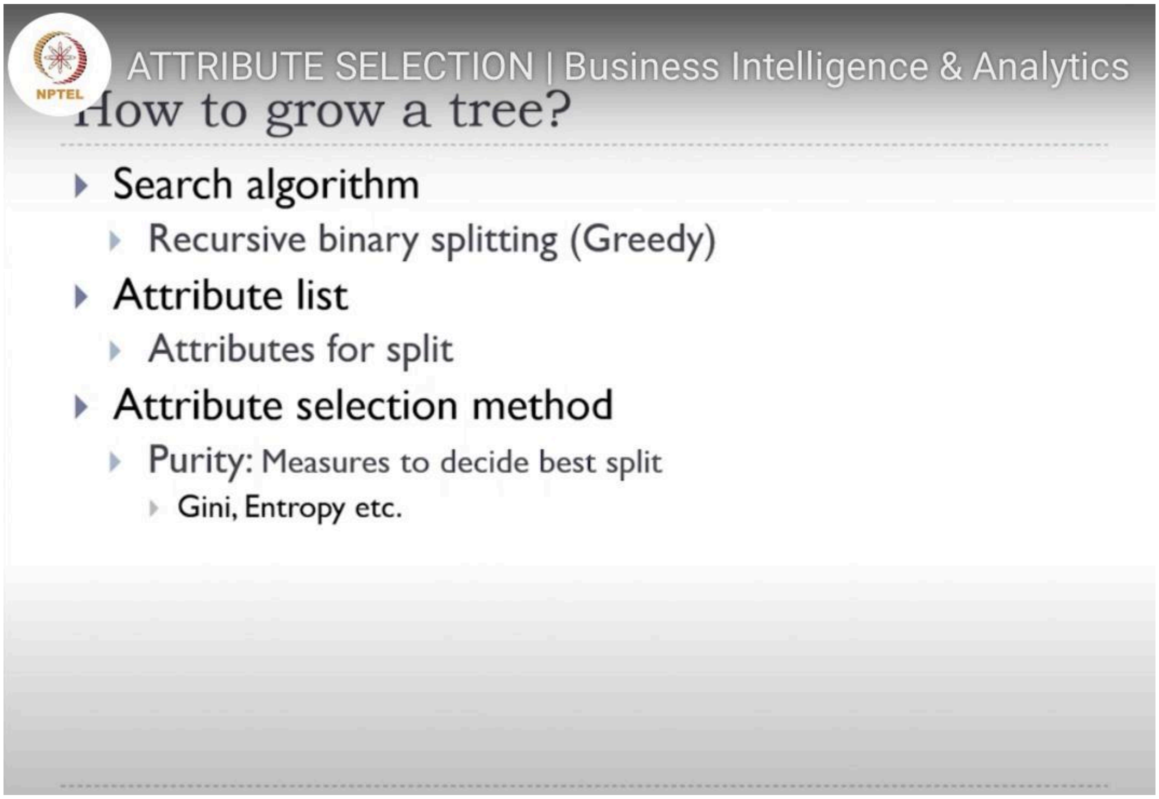
Course Name:Business Intelligence and Analytics
Professor Name:Prof. Saji.K.Mathew
Department Name:Department of Management Studies
Institute Name:Indian Institute of Technology Madras
Week:07
Lecture:26

ATTRIBUTE SELECTION | Business Intelligence & Analytics



ATTRIBUTE SELECTION | Business Intelligence & Analytics

How to grow a tree?

- Search algorithm
  - Recursive binary splitting (Greedy)
- Attribute list
  - Attributes for split
- Attribute selection method
  - Purity: Measures to decide best split
    - Gini, Entropy etc.

Now, let us slowly move on to understanding how a decision tree algorithm will work. Or what are the essential inputs and steps? What are the inputs or arguments that an algorithm will require? And what are in the control of the analyst or the researcher while specifying an algorithm for, sorry, decision tree induction? So, number one, functionally, a decision tree algorithm is a search algorithm. A decision tree algorithm is a search algorithm. What does it search for? It searches for homogeneity. We have seen that the purpose or the objective is to create more homogeneous subgroups than the parent node. And the decision tree would do this in multiple steps till you get reasonably sized, reasonably homogeneous leaf node.

So, the same search algorithm will be repeating from iteration, sorry, from step to step. And therefore, it is recursive in nature, because it is the same set of steps. And typically binary splitting, a binary tree will do binary splitting. So, it starts with a node, splits into two, and then you have two sub nodes. And that will also be split into two, if it can be split, then you have further sub nodes. And this goes on till the terminal nodes based on some rule for termination is reached.

So, there is also the word greedy added. Decision tree algorithms are greedy algorithms. What does greedy algorithms mean? When the node splits into two subgroups or sub nodes here, then the decision tree decides to split based on certain criteria, based on certain condition.

We will touch upon that in the next slide. That condition obviously is, that condition which will lead to maximum purity or maximum homogeneity for the subgroups that is formed. But it only looks at the homogeneity of the immediate subgroups or immediate children. It does not look at what could happen for, during further split. It does not look at the homogeneity of the potential grandchildren and grand grandchildren and so on.

It only looks at the immediate outcome. And therefore, this is known as a greedy algorithm. And this is obviously based on the complexity or the cost of optimization, if you go for considering a lot more steps before deciding on the split. So, I leave that part, but that is the way most decision tree algorithms work. They are greedy algorithms.

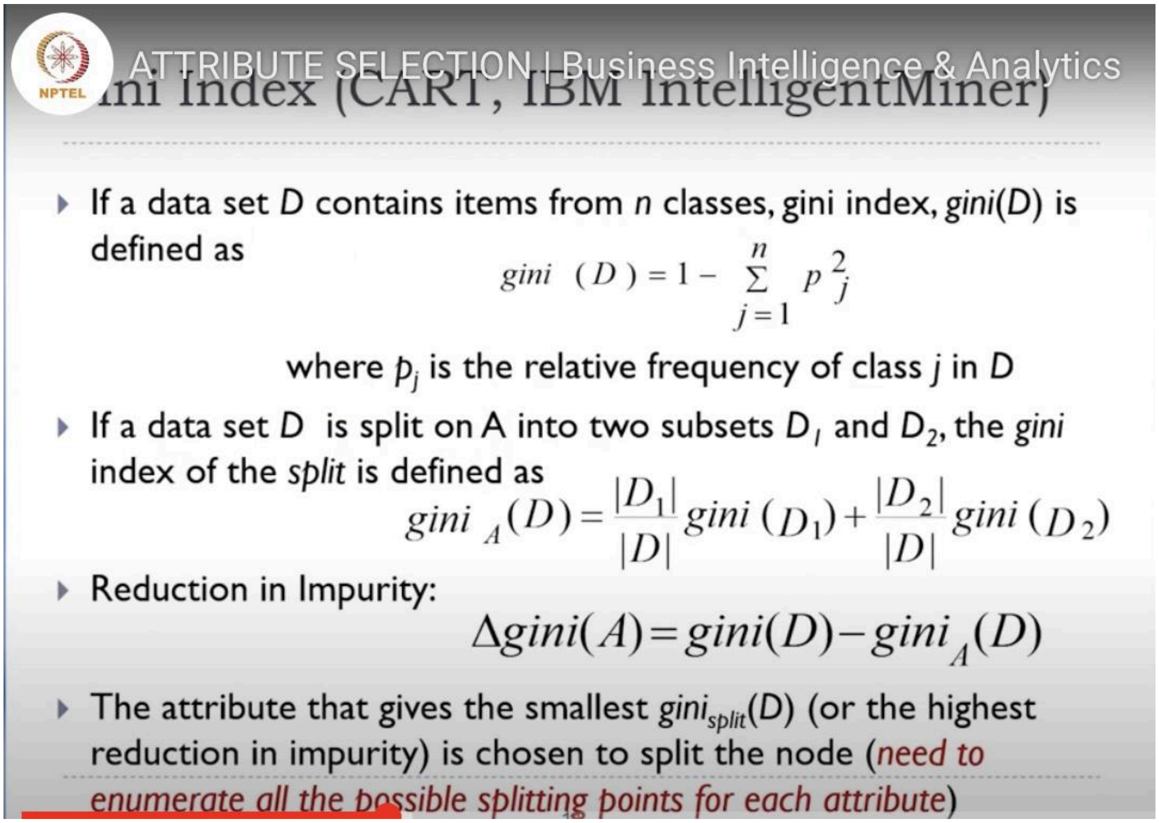
And there is, of course, research into it. There are algorithms which are not greedy as well, but I am talking about a general case and the rationale behind it. And then one is to provide the data set with a clear understanding of which are the attribute or the features and which is the target variable. So, there is the x data set and the y. y is your outcome, x is the feature set or the set of attributes and data is actually is prepared in that format and that is the input at the root node.

So, data and then along with that the attribute list as I said, if there is x then what is x1, x2, x3 to xp. There are p number of variables and their data. Now, this is the set of attributes. y is separate, that is the target variable. Now, this far is fine, data, attributes, target variable all defined when you specify a model.

And the third important step is to have a measure, measure for, measure for what? You see an algorithm searches for purity or homogeneity and there has to be some measure because all these algorithms can work only with objective measures. And therefore, there has to be some measure of homogeneity and what is generally used term in decision tree literature, is purity. It is synonymous with homogeneity. So, purity of a node, purity of a node and also purity of a split is something an algorithm will evaluate before deciding which attribute to use for split. And therefore, there has to be some measure, some measure of purity.

An algorithm has to search for a right conditioned split. Right conditioned is a condition based on a given attribute, one of the attributes, which attribute to select is the search. It is the algorithm will search for that and which attribute to select depends on which attribute based split leads to highest purity of the sub nodes. And that is quite intuitive, is not it? The purpose, the objective is to get results that are highly homogeneous or highly pure as compared to the previous nodes. Therefore, there has to be some measure for purity.

And I am going to deal with two measures, two widely used measures in decision trees, one is the Gini score and other is entropy. Both the measures are widely used and they are measures of purity, measures of purity when a decision tree node is split. And they operate both at the node level and also at the split level, meaning you can have a measure for purity for a given node, you can have a measure for purity for a given split. We will see those details as we go now. So the Gini index, so within brackets I have shown which are the algorithms that widely uses Gini index, it may not be very updated, I am just giving examples.



ATTRIBUTE SELECTION | Business Intelligence & Analytics

ini Index (CART, IBM IntelligentMiner)

- If a data set D contains items 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 data set D is split on A into two subsets D_1 and D_2, the *gini* index of the *split* 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 gives the smallest $gini_{split}(D)$ (or the highest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

CART stands for classification and regression trees. And the IBM intelligent minor is more proprietary. But leaving that aside, let us try to understand what is a Gini index? What is a Gini mode of measuring purity of nodes and split? So, here is the definition, if a data set D contains items from n classes, n is the number of class labels or the number of values of the outcome variable or target variable. So, data set D meaning that is the root node with a particular size, D has a particular size, D is the whole data set. And in the data set, there are attributes and target variable.

Target variable has n classes or n values, like the n classes in the case of our sunburn, can be sunburn, no sunburn. So therefore, two classes. So that is for understanding. So, if a data set D contains items from n classes, Gini index represented as Gini D is, Gini D = $1- \Sigma^{n}_{j=1} p_j^2$. And p stands for the proportion of the jth class in the node. p stands for the proportion of the jth class in the given node. You can easily imagine here that when you have a node and they are split, now that node is split. So, each node will consist of different classes of the target variable. For example, each node will have, sunburn and no sunburn. And what is the proportion of j= 1, suppose it is sunburn, what is the proportion of sunburn records in that node plus square of that, what is the proportion of no sunburn classes in the node.

So you find the sum of the squares of the proportion of the classes in the node and then there is a 1 minus. Correct, and that is a formula, that is a formula for Gini score of a node. That is a formula for Gini score of a node. And if it is a categorical variable and not a binary variable, you can extend this to n number of classes, that is a generic formula. And now, that is about one node, the purity of one node as measured by Gini score is this, given by this formula.

Now, what is more important for a decision tree split is to understand what is the overall purity of a split. Decision tree algorithm search seeks to maximize the purity of split. And therefore, there has to be an overall measure of a split, not the Gini score of a single node, but the combined purity. And that is given by the next formula. So this is to be understood as, Gini score of a node D when split based on an attribute a and here we assume that there are many attributes a, b, c, sorry to say p, p number of attributes.

Suppose there are p number of attributes, you first select attribute a. So Gini for node D, when split based on an attribute a is given by D1 / D. This is D1, this is D2 and this is D. So that solves the problem of understanding how it looks like.

This is D. Again not done well. Let me write it. D,D1,D2, Dis the parent node, D1 and D2 are the two child nodes. So D1 / D, absolute value of D1 by D, this stands for the size.

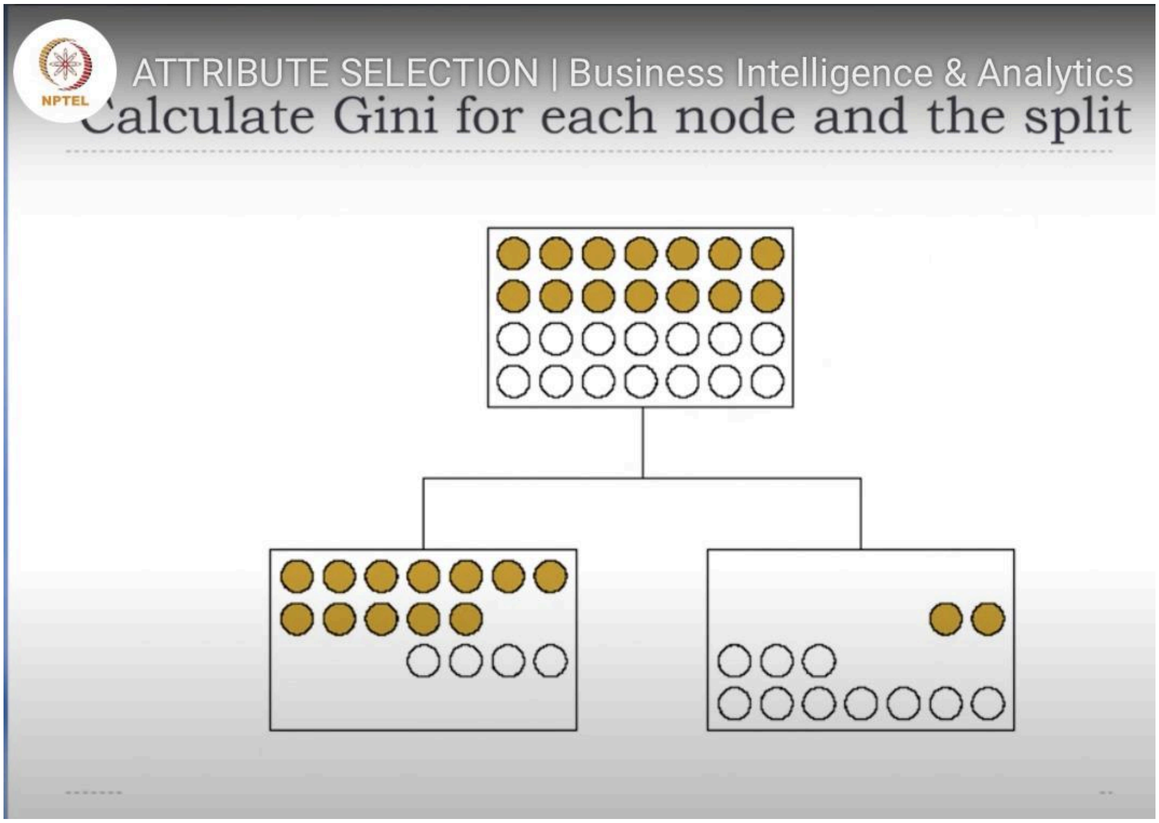
This stands for the size. This symbolizes the size of D1 upon D into Gini of (D1 + D2)/ D into Gini of D2. Obviously, when you look at this formula, you have to calculate Gini of D1 and D2 using this formula. And this is equation 1 and this is equation 2. And then you apply or substitute the values of Gini D1 and Gini D2 into equation 2, wherein you also need to calculate the weights or the ratios of each node. D1/ D and D2 / D is what? These are the relative sizes. These are the relative sizes of the nodes. Or in other words, we can say that a Gini score of a split based on a given attribute A is the weighted sum, weight with respect to the sizes. The weighted sum of the Gini scores of the individual nodes or the child nodes. That is the meaning of this formula.

Weights with respect to sizes. So it is the weighted sum of the Gini scores of the child nodes. That is the Gini score of the split. And then of course, you can calculate something known as a differential Gini, reduction in impurity when split based on A is a Gini of D minus Gini of the split based on A. That is the net improvement in purity that you achieved when you split the node into two, based on attribute A. Now, this is becoming quite obvious to us.

What is the search algorithm does? It searches for that particular attribute, which maximizes this value, which maximizes purity. Whichever attribute maximizes purity is the most desirable attribute to split a particular node. Now, you can imagine how the trees goes, the tree goes on to grow based on a measure like Gini score. And when Gini score is used as a basis to measure purity of a split, then the decision tree algorithm identifies the right attribute to split at each level and builds the tree.

Now, let me take an example to illustrate the concept of the purity measurement, which we just discussed. And probably that will help you understand and really imagine what goes on when a decision tree algorithm is inducing certain rules to create a classifier. So here, you can see,it is a binary split, there is a root node. And what is visualized is the two values of the target variables. So you have the brown class and the white class or the white circles or the brown circles. And it is split based on some attribute A.

Let us imagine it is some attribute A and some condition based on attribute A. And so let us say this is the D and this is D1 and this is D2. Now, our aim is to apply the formulae which we learned in the previous slide and calculate the Gini scores of the nodes and finally the Gini score of the split. And that is what we are going to do here.



ATTRIBUTE SELECTION | Business Intelligence & Analytics

Calculate Gini for each node and the split

Now, in order to do that, one of the first things that we can do is to look at the sizes. So what is the size of the root node? This is 7 × 4= 28. The size of the left node D1 is 7 + 5, 12, 12 + 4, 16. And the size here is D2 is of course, it is the remaining 28 - 16 or this is 12. So we got the sizes, a sense of sizes that will actually help us sort of compute the Gini score of the split.

So first let us calculate the Gini score of the root node. What is the Gini score of the root node? Gini D is equal to, of course, this is 1 minus proportion of the brown circles. Proportion of the brown circles is obviously 14/ 28. The 14 brown circles, of course squared plus the white circles are also 14 in numbers. So 14 / 28, the whole squared.

And we close the formula. And therefore, it is actually a case of $(1 - 0.5)^2$, which is 0.25 + 0.5^2, which is again 0.25 or in other words, this is 1 - 0.5 = 0.5. We have an interesting observation here. When you look at a node like D, we can see that this is for a binary outcome variable. This is a fully heterogeneous node, equal distribution of the two classes brown and white.

So in other words, when in terms of Gini score, if a node is completely impure, when a node is completely impure, the Gini score is equal to 0.5. Now let us see. So we can imagine again, if a node is fully pure, suppose there were only brown nodes and no white, brown circles, not nodes, only brown circles and no white circles and suppose it were completely homogeneous, then you know, there is no, one numerator will be 0 and the other will be same as numerator will be equal to denominator.

So therefore, it is 1 - 1^2. Therefore, the 1 -, 1 - 1, which would show that for a pure node, the value of, the value of Gini score will be 0. So Gini score will vary from 0 to 0.5. That is the range, in which.

So the lower the Gini score, the better the purity. Of course, I said the objective of decision tree algorithm is to maximize purity, but in terms of the Gini score, this will be to minimize the Gini score. So this is inverse, inversely coded or scaled. Alright, so that is one aspect. Now, let us calculate the Gini score of D1. Gini of D1 is equal to 7 + 5= 12, $(12 / 16)^2 + (4 /16)^2$.

I am not, put it well, let me correct it. 1 - Gini D1, D1 is equal to 1 -. Sorry, 1-{ $(12/16)^2$ +($4 /16)^2$}. You will get some value, I am not calculating that value. And obviously that value will be less than 0.5, because this is more pure, you can see that it is a reasonable split, the child node is more pure, so it will be a value less than 0.5. And now look at the other one, what is the Gini score of D2 is equal to 1 -, you can see{ (2 / $12)^2$ + $(10 /12)^2$}. This is some other value. Now suppose this value is say, let me say, I am not calculating, I am not spending time to calculate this. So suppose this value is A1 and suppose this value is A2.

Now our goal is to calculate the Gini score of the split. And therefore, ultimately we need to find out what is Gini of when you split D with respect to attribute A is given by the weighted sum of the Gini scores. What is the weight that should be assigned to node D1? The weight is its relative size and size is, the size of D1 is 16. Therefore 16 divided by what? What is the size of D? That is 28. 16/28 × Gini score of D1, which we calculated as A1, A1 +, this appears to be cluttered and I will write it again.

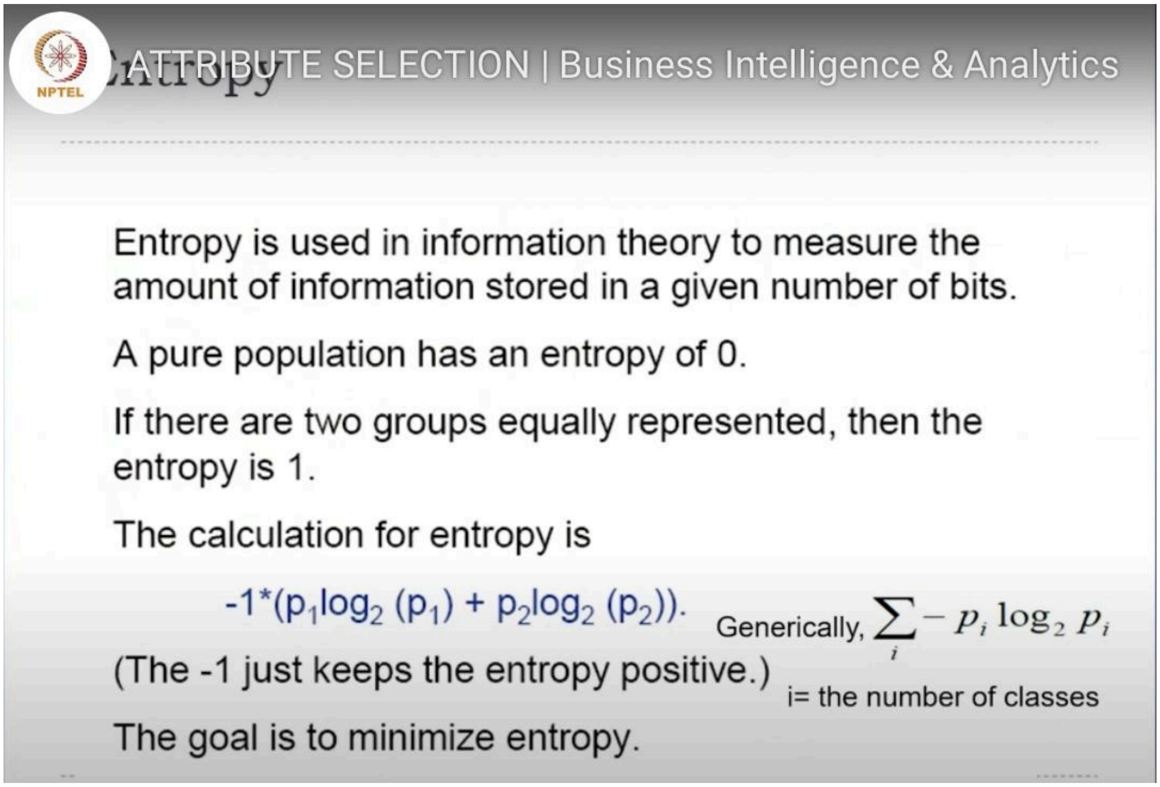
You should not get confused. Gini score, Gini of the split is given by weighted sum. So weight is 16/ 28, that is the size of the root node. The left node has 16 members. Sorry, there is no squaring there. That is the weight of it into Gini score of D1, which is A1 +.

Now we come to the right side, what is the relative size or the weight that we should give here? Size is 12. So 12 /28 × the Gini score of the right node is A2. This will be the value of the split. Correct? And suppose this overall value is say, let us say this is A1.

You get a value,sorry, I should not call it as A1. Suppose this is, let us use the term G. So I will call, because it is Gini score. So suppose this is G1. The objective is to choose that attribute, which would maximize purity or minimize Gini score. In this case, we have already seen it is inverse, inverse relationship.

So therefore, suppose you have in your project, three attributes, three candidate attributes to split or A, B and C. You split based on attribute A, the Gini score of the split is G1. Suppose you split based on B, you again come to recompute because the subgroups formed will be different in terms of their structure and constitution. So you get a Gini score, which is G2 here. And when you split based on so A, B and subsequently you go for C, you get G3.

Now algorithm searches for highest purity or lowest Gini score. So whichever is the lowest value in terms of Gini score of the split, that particular attribute would be used to split the node. In our first example, when we found that hair color was used to split the tree first and we asked this question, why hair color, why not height, why not weight? Now you know what is the basis for deciding which attribute to use, to split a given node, it is based on the value of split purity. And that is what we have seen here.



ATTRIBUTE SELECTION | Business Intelligence & Analytics

Entropy

Entropy is used in information theory to measure the amount of information stored in a given number of bits.

A pure population has an entropy of 0.

If there are two groups equally represented, then the entropy is 1.

The calculation for entropy is

$-1*(p_1\log_2 (p_1) + p_2\log_2 (p_2))$. Generically, $\sum_i - p_i \log_2 p_i$

(The -1 just keeps the entropy positive.) i= the number of classes

The goal is to minimize entropy.

Let us move on. Let us talk about another measure called entropy. Entropy is a measure that has been adopted into information science from thermodynamics, from a field of natural science to information science and of course, data mining and analytics actually draws upon information science as a science, a scientific basis. And a scholar of

MIT, Claude Shannon actually developed, is also known as the father of information science. So he developed the science for doing data mining and one of them is of course, one of his contribution is this formula, to calculate a measure similar to the Gini score when you actually use decision trees. You can see Claude Shannon's formula uses the same variables.

The final measure is entropy. Entropy is the final measure of purity, but it is given by $- 1\times\{p_1 \log_2 (p_1)+ p_2 \log_2(p2)\}$. What is p_1 and p_2? It is the same as the proportions, the proportion of a class within A0. We discussed what is that proportion in the previous example. And this particular formula is given for a binary target variable, where there are only two proportions p_1 and p_2 or two classes, 1 and 2. But if there are more number of classes, you can use a more generic formula like this.

And why is a -1 at the outside of the bracket? That is because when you find algorithms for numbers less than 1, you know, this is all proportion. So, the value will be 1 or less, you will get negative values. And therefore, in order to make entropy positive, he added a -1. And so, therefore, entropy is also a measure of purity and a pure population has a entropy, which is 0. Entropy by connotation means disorder in thermodynamics, or you know, the extent of useful energy.

So, the more the entropy, the less useful or the more the disorder. And therefore, extending the same concept, if entropy is high, it only means that the node is more impure. So, low entropy is more desirable, as in the case of Gini, a low Gini score is more desirable, because that means more purity. So, in the case of a binary node, the range of values is likely to be 1 or not likely to be, it is the value will be 1. The highest entropy value for purity, highest purity, when only one class is represented in a node, then it will be 1. I am sorry, it is opposite is the case, sorry for making a wrong statement, when a node is completely pure, entropy will be 0, when a node is completely impure, meaning, if it is a binary case, if both the classes are equally represented, then the entropy will be 1, showing that the purity is most, you know, this is most impure or it is most heterogeneous, equal, equal mix, that shows that there is no purity.

And that is the case of entropy equals 1, high entropy is not desirable. So, an entropy value should again tend towards 0. And that is the lowest value of the entropy would be used by the decision tree algorithm to make the split. So, I hope this is clear. So, essentially to sum up what have we done now, we know that you need an algorithm to grow a decision tree, an algorithm searches for purity of child nodes and therefore, we need measures for, measures for purity, to calculate purity of split based on each attribute. So, we have discussed two such measures Gini score and entropy, which are widely used for calculating impurity or purity.

They are actually scales of impurity. And therefore, as I said, if you have to look at purity, you have to look at it inversely and therefore, I made a confusion here. Sorry about it.