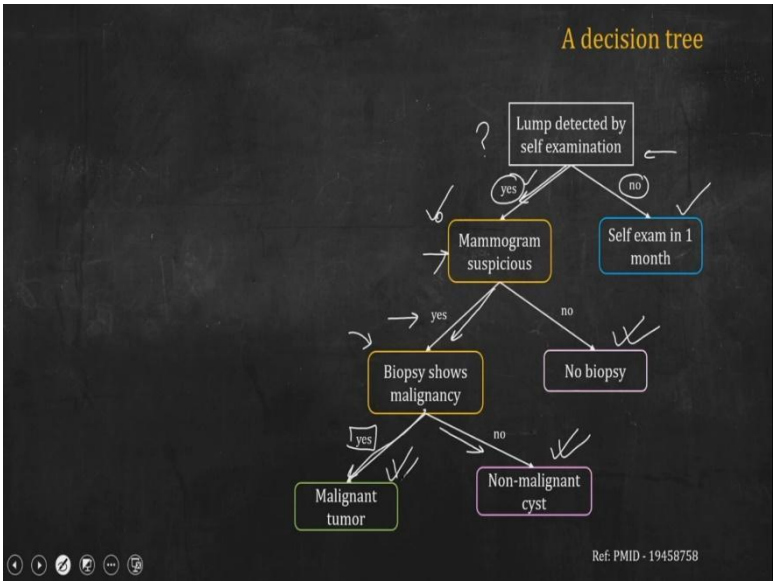
Data Analysis for Biologists
Professor Biplab Bose
Department of Biosciences and Bioengineering
Indian Institute of Technology, Guwahati
Lecture - 42
Decision Tree Classifier

Hello everyone. Welcome back. In this lecture we will discuss about Decision Tree Classifiers. Before we go into the issue of a classifier, let us first learn what is a decision tree. I have taken an example.

(Refer Slide Time: 00:48)



Suppose, this is a decision tree used by a clinician to decide upon breast cancer. Let us understand how this tree work for a clinician. In this tree, each node, suppose this is one node, it is get divided in two parts, two paths. So, it is a binary tree. And at every node, we are asking a question and that question has some yes no answer.

So, at the root of this tree, at the top most node, suppose, a person comes to clinic after doing self-examination and the clinician asks this question, the lump detected by self-examination, has the person detected the lump by self-examination or not. That is the question the clinician asks.

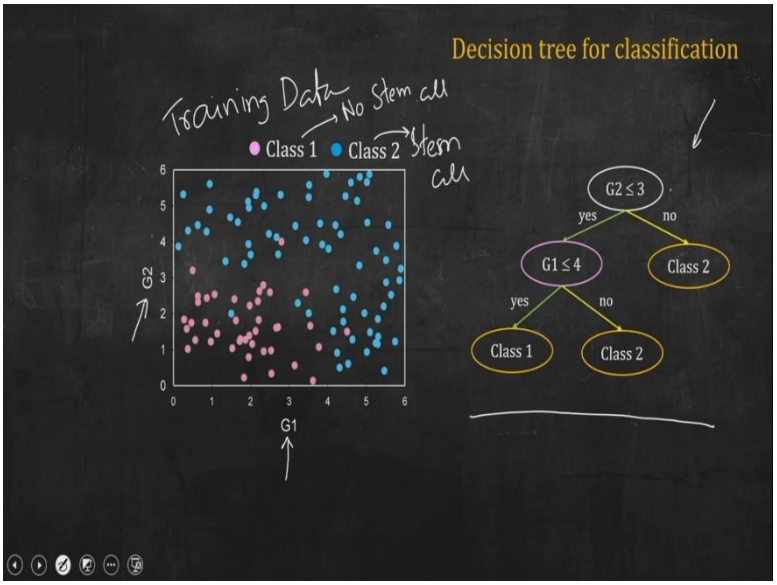
So, if the person has detected the lump in breast by self-examination then clinician follows this path. Because it is for yes. And the clinician decide to go for mammogram. If the lump was not detected by the person by self-examination then she will be recommended to do a self-examination may be after one month.

Now, look at this node, in this case, the clinician has recommended for mammogram and the mammogram has come, report has come. Now, the clinician asked the question, is the mammogram suspicious? If it is not suspicious then the clinician will decide, okay, we do not need any biopsy. The story ends there.

If the mammogram is suspicious that means we are following this yes arm of the tree then the clinician will go for biopsy. And the clinician now will ask does the biopsy show malignancy? If it does not show malignancy then the clinician will follow this path, this arm of the tree and will decide that this lump is nothing but a non-malignant cyst. Whereas, if the biopsy report shows malignancy that means this yes is correct then the clinician will follow this path and will decide that the lump that is detected in the breast is a malignant tumor.

So, this is a decision tree. It is a binary tree. And it is made of nodes and edges. At each every node, you are asking a question and that question has yes-no answer. In decision tree classifier, we will use this type of binary tree to classify data. So, let us learn how we do that. So, let us look into a data and the decision tree that I have built based on that data. Now, remember, this data is a training data set.

(Refer Slide Time: 03:40)

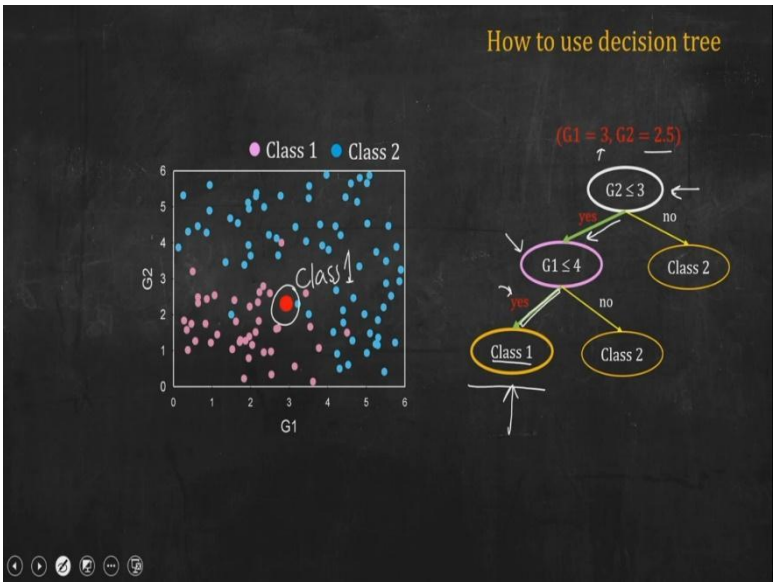


So, here is my decision tree that I have built using this training data. Here suppose, you can say, there are two predictor G 1 and G 2. You can imagine them. These are two genes which are marker for stem cell. And I may have two classes. Class 2 is stem cell. And class 1 may

be no stem cell. So, the data distribution you can see, there is some amount of patchy region but there is also diffused region also.

Now, based on this training set, I have created this decision tree. This decision you can see is a binary because at every node there is a question asked and there is a yes-no answer and we are following this yes or no path. Now, before I go into details that how I build this tree, the algorithm, let us see how I can use this tree classifier to classify a test data.

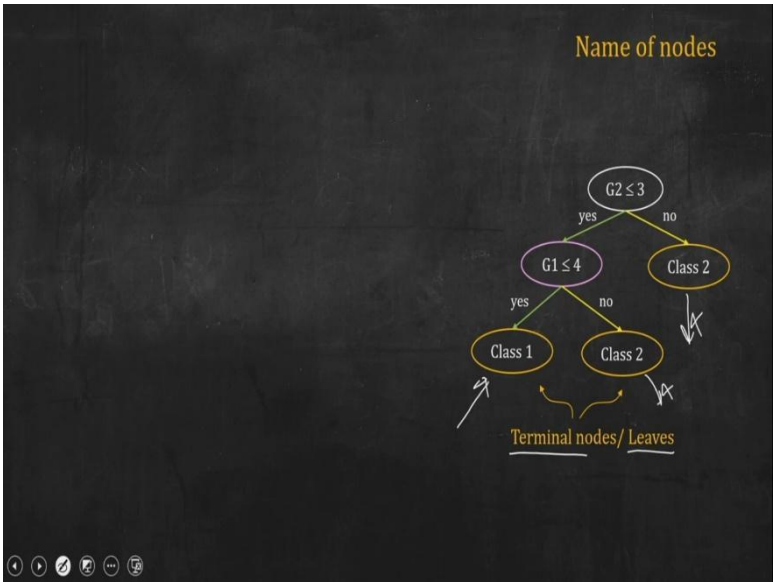
(Refer Slide Time: 04:52)



So, I have a test data, here, the red dot and its values are suppose G 1 is 3, G 2 is 2.5. So, how should we use the decision tree classifier? I will take this data at the initial node, the root. So, I have taken there. Now, I will check the first node, the first question. What is the first question? Is G2 less equal to 3? Okay. G2 is 2.5. So, it is less equal to 3. So, I will follow this yes path. I have marked that by red.

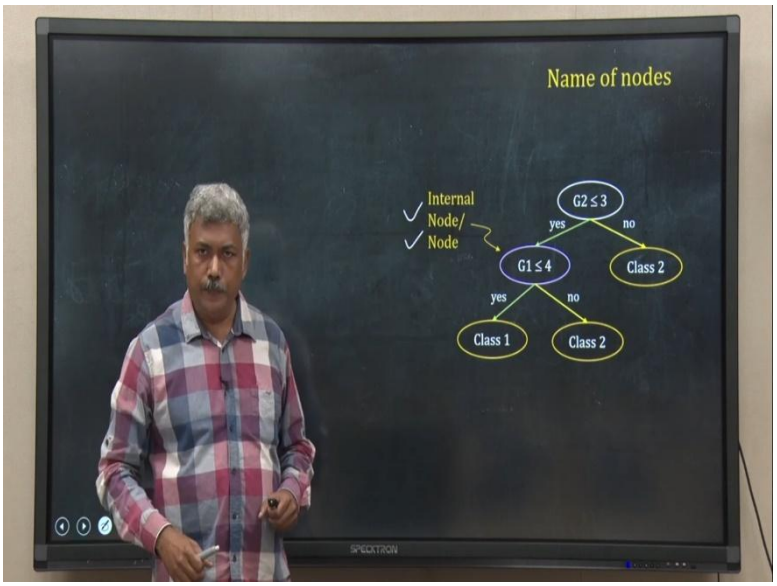
Now, I am in this node, the pink colored one. A new question is asked in this node. Is G 1 less equal to 4? Ok, let me check G 1. G 1 is 3. So, that means it is less equal to 4. So, again yes. So, I will follow this yes path, again I marked it by red. So, I land up in a node which tells that this node tells me that this test data point is class 1. So, this is class 1. So, I have classified the test data. In this way you can take hundreds of test data and classify them. Now, I will move into how actually I build this decision tree classifier. But before that some technical terms, let us get acquainted with.

(Refer Slide Time: 06:14)



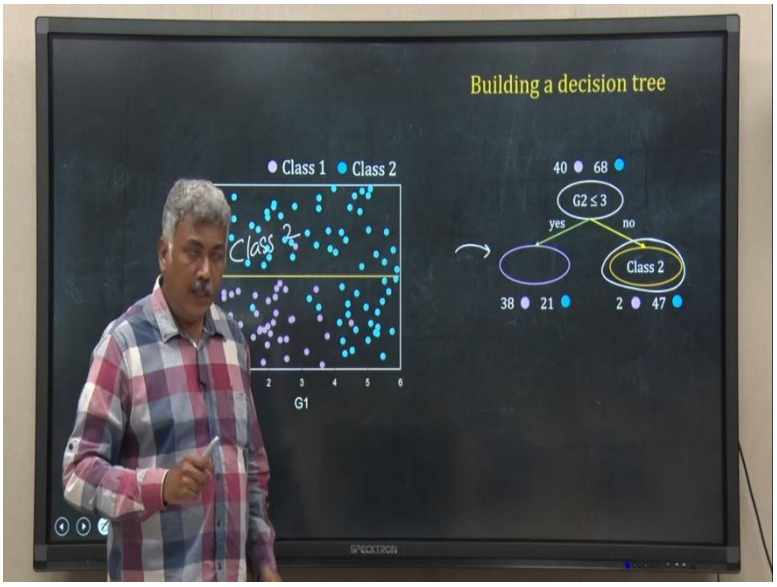
The topmost node is called the root, whereas the lowest one, where already we have made the decision about the class, will be called terminal nodes and leaves. And you can see from this terminal node, there is no more arrows. There is no more arrows because we are not asking any question there. We have already made the decision about the class.

(Refer Slide Time: 06:36)



So, those are the leaves and these are, the in between nodes where we ask questions to make decision, are called internal node or simply sometime people will call node. Now, let us look into how this decision tree classifier is built using a training data set.

(Refer Slide Time: 06:54)



So, I have the training data set shown here and I want to build this one. How should I proceed? Okay. The first step. Take all your data initially in the root node. So, what is my initial data? This is my initial training data. I have 40 data points, the pink color one, the class 1. And there are 68 data point for class 2. I have taken all those in the root node.

Now, I have to ask a question. What type of question? Like for example, is G1 equal to 1? Is G 1 bigger than 3. Something like that, a logic operation. And which will have yes-no answer. And I will ask the question in such a way that from a heterogeneous population, by giving this yes-no answer, I get more homogeneous subset of data. Let me explain again. I have a heterogeneous data here, 40 pink, 68 blue.

Now, I want to ask question in such a way, by giving yes-no answer, I should get more purer or homogeneous subset in these two places, in these two child nodes. So, the question that has been asked in my model, in my tree, is G2 less equal to 3? Then what does that mean? Actually, what I get I have shown it visually here. This question divide my data in two part.

So, the below one this one is yes. In this region, in the below region of the, in the below this yellow line, all data are satisfying that G2 is less equal to 3. So, all those data will come here, in this child node. So, that is why I have 38 pink 21 blue. Blue is for class 2. Now, look into the rest of the data. Rest of the data is from here.

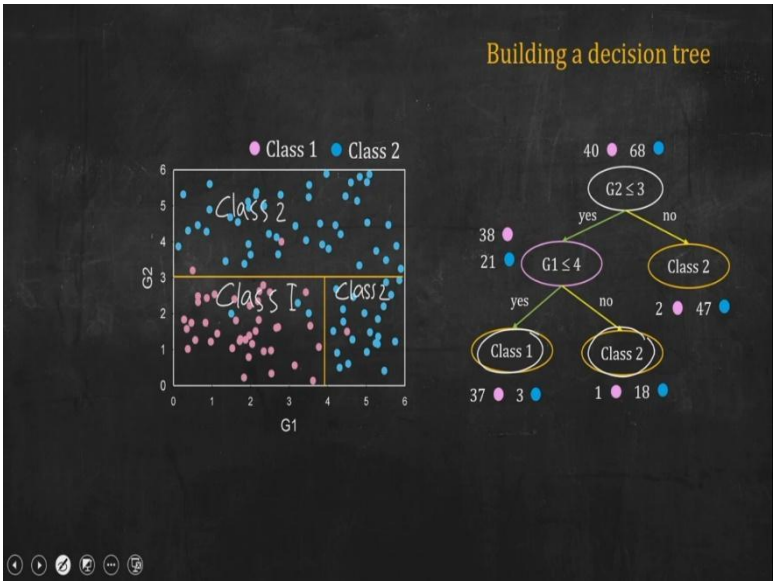
And you can easily see there is only two pink dots and rest of the data are class 2, blue one. So, that means by asking this question, asking this question I have separated the data in more

purer subset. Obviously, now I have a subset of data which is purer than the original one. But, how do I measure purity? I will come to that. There are certain mathematical measures of purity that is used to divide this data and to test which question I should ask. I will come to them later on.

But, now let us proceed. Suppose, we know that measure of purity. And if you look into this node, this child node here, you can easily see or you can focus on this part, you can easily see, it is largely homogeneous. It is mostly the blue colored dots, the class 2 thing. So, the classifier decides to stop there and say this is class 2. So, all this region is class 2.

Now, I will work this, on this, the algorithm will work on this child node. What it will do? Here, it will ask another new question, just like is G3, G2 greater than 4, G1 less than 2, something like that. And the algorithm will try to ask the question that will create more purer sub-population or subset of data.

(Refer Slide Time: 10:28)



And in this case, the algorithm has decided to ask the question, is G1 less equal to 4. Let us look and visualize that. So, here remember, we are not dividing the whole data set. We are dividing only the data set which is in this region. So, that means at that node, the algorithm is working locally, not on the whole data. That is very important; I will come back to that.

So, here in this data set, this box I have how many, 38 pink and 21 blue. So, now, the question G1 is less equal to 4 is dividing this data into two region. This one and this one shown by this

yellow line. And doing so, you can easily see, this part is largely blue, whereas this part is largely pink.

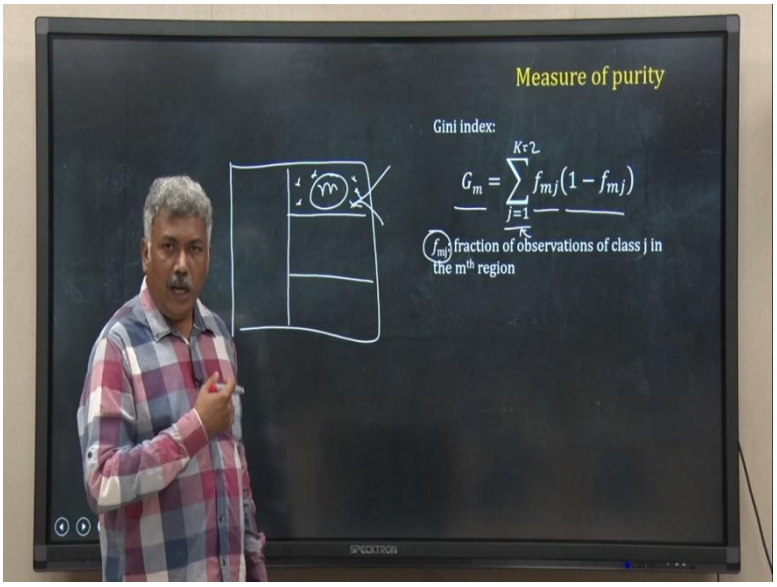
That means now, I have got two sub population, two subsets which are much purer than the previous situation. So, now, I have 1 pink and 18 class 2 data points on that child node, whereas this one has 37 pink or class 1 and 3 class 2. So, I have got relatively more purer sub population. And as they are largely pure, the algorithm decides to stops here, stop here.

And call this as class 2, this as class 1. That means this is class 2, this is class 1. This one earlier it has decided this is class 2. We are done. There is no more data left to classify. We have reached the leaves or the terminal nodes. So, the classifier stops there. Now, you can use a test data to classify the test data.

So, one crucial aspect of decision tree classifier is that at every node, it make a decision, it asks a question. Now, remember, it can ask large number of question, n number of question to divide this whole space of the data in multiple two, in two part, two segments. Which question it should ask? As I said it will ask a question which gives me more pure subset of samples.

Now, multiple questions, suppose three different question can give me a more pure data set with respect to the original data set. So, that means I have to compare between the outcomes of all these three questions. So, I need some measure, some measure for purity of data in a particular segment, particular subset, particular region. And there are two such measure that I will discuss in this lecture.

(Refer Slide Time: 13:26)

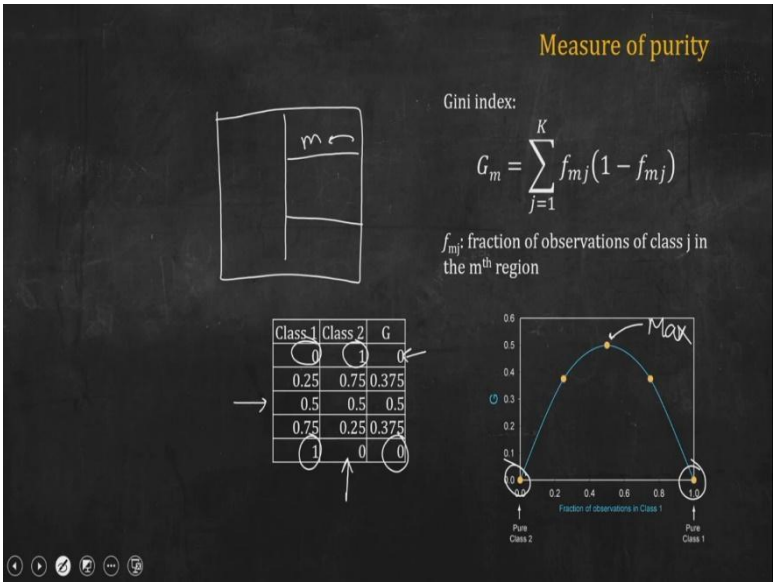


$$G_m = \Sigma\ f_{mj}\ (1 - f_{mj})$$

One is very common its called Gini index. What is Gini index? So, suppose let me draw, this is suppose the whole data space and I have divided in this, this way. Four division I have divided. And this is suppose the m'th part. And I have some data points here. Some of them are class 1, some of them are class 2, something like that. So, f_{mj}, f_{mj} is a fraction of observation of class j.

Suppose, we have two class. So, class 1 or class 2. j is one or two in this m'th segment, in this m'th region. So, that is the fraction of the data, whole data which is present in m'th segment and belong to the class j. So, it is f_{mj}. So, Gini index for this segment, Gini index of m is equal to f_{mj} into 1 minus f_{mj} and it is sum over all the classes. That is why j equal to 1 to k. Suppose I have k classes. In the previous example I had two classes. So, it will be k will be equal to 2. Now, let us see what is the Gini index for our classification tree.

(Refer Slide Time: 14:44)

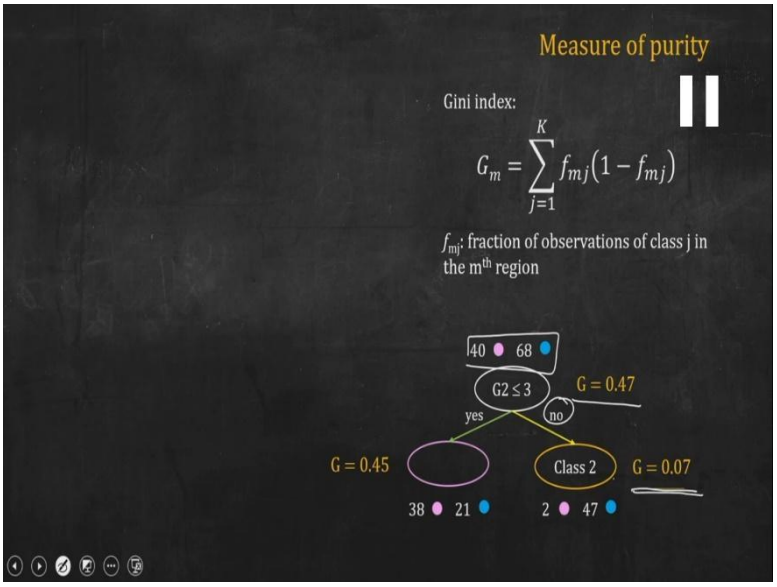


Before I go there, let me show some numerical way the properties of this Gini index. So, what I have done here, suppose, I have again, this one let me draw, I have four segment. This is the m'th segment. And there, suppose class 1, there is no class 1 data point. All the data points are class 2. So, the fraction f for class 2 is 1.

And if you calculate the Gini index for this, it will be 0. So, here. So, that means in this segment, in this region of the data we have pure class 2. Whereas, if all the data point there is of class 1, there is not a single class 2 data point then also Gini index will be zero. So, that is pure class 1. Let us see in between. Okay.

Suppose, in this segment or both class 1 and class 2 are present in equal amount, 0.5, 0.5 then I have the maximum value of Gini index. So, you can see as the population become more heterogeneous, Gini index increases, and then as it becomes more pure, it becomes low and low. So, both side of this maximum, you have more pure and pure sample. So, you have low Gini index. So, as I said we will do the calculation for our tree.

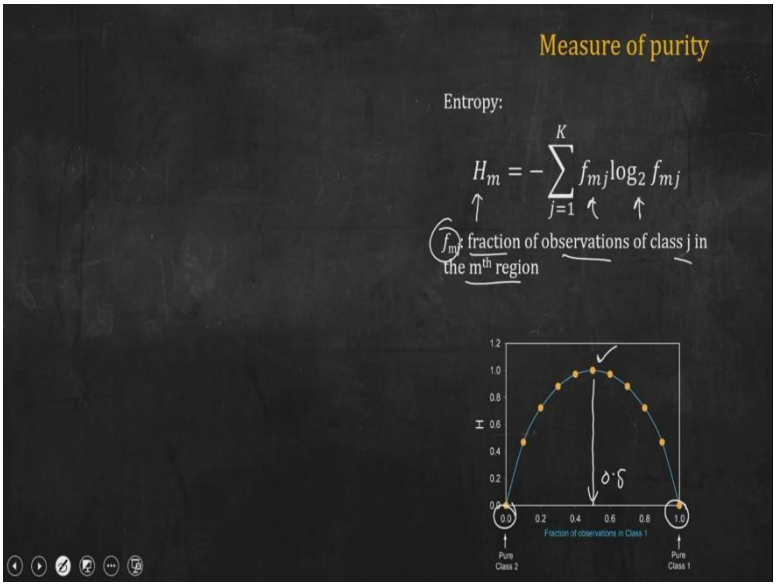
(Refer Slide Time: 16:08)



So, I have shown for one part, initial data. All the data taken here at the root node. Gini index is 0.47. Whereas, after I ask the first question and segment the data into part, in this, where we have got the no answer, you can see Gini index is so low, 0.07. So, that means that question, the first question of the root node has divided the data and created one subset which is very pure, that is why the Gini index is 0.07.

And that is why the classifier has decided to stop there and call the class. And it has called the class as class 2. Because a predominant majority of those data points are of class 2, the blue one. Now, there is another similar parameter which is used to measure purity of data set that is called entropy.

(Refer Slide Time: 16:58)

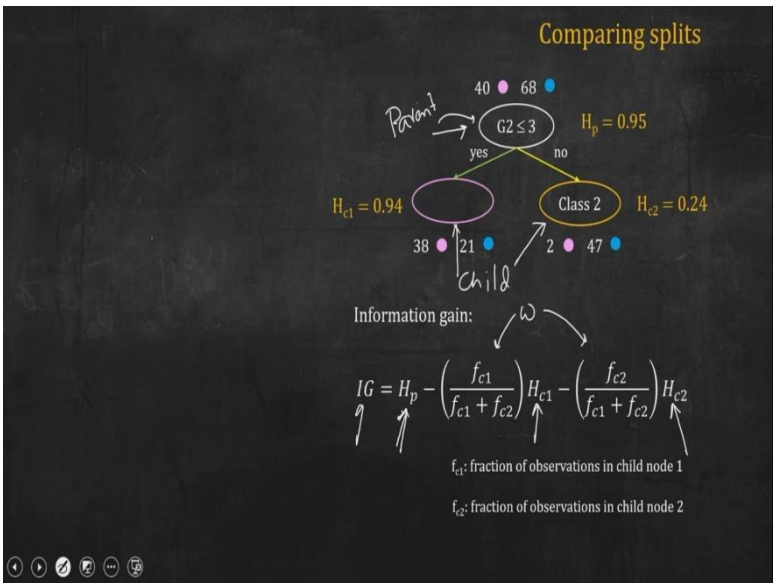


$$H_m = - \Sigma f_{mj} \log_2 f_{mj}$$

And again the same way, f_{mj} is the fraction of observation of class j in the m'th region, the same definition. The only thing is the formula is changed. So, the entropy, entropy of the m'th region, you have divided in multiple region the data set is equal to summation of f_{mj} $\log f_{mj}$. I have taken the base as 2. So, the entropy will be, unit will be bit.

And if you look into the behavior of entropy, it also has this behavior. When I have pure population, I have entropy 0, least entropy. As the population becomes more heterogeneous mixed type, the entropy increases and peaks at 0.5 when both the population of class 1 and class 2 are in equal amount. So, it is also behaving just like the Gini index.

(Refer Slide Time: 17:52)



$$IG = H_p - \left(\frac{fc1}{fc1+fc2}\right) H_{c1} - \left(\frac{fc2}{fc1+fc2}\right) H_{c2}$$

So, now, here I have shown the calculation for entropy in case of our first splitting. So, here entropy is very high because at the root node we have taken the whole heterogeneous population data. And in the class 2 which I have got by this no answer has a lower entropy 0.24.

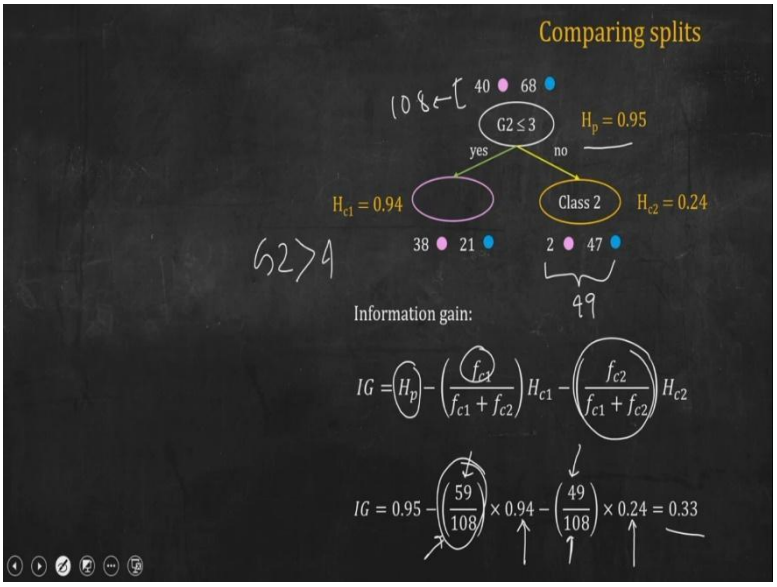
Now, let me remind you that what is happening, at each node, for example, this is a decision node though its a root node, here again this is decision node, in this decision node we are asking a question to get a purer subsets of data. Now, as I said earlier, I can ask n number of question. So, that means I have to compare between the outcomes of multiple questions. To compare these outcomes what usually we use is called information gain. Let me explain it.

So, in this case, we have asked the question whether G2 is less equal to 3 or not. So, I will check what is the information gained by this question. Ok. The formula of information gain is you take the entropy of the parent node. What is the parent node here? This one is the parent node and these are the child node. We have two childs, its a binary tree.

So, entropy of the parent node minus entropy of the first child node minus entropy of the second child node. But you do not take the raw value of the entropies of child nodes. You multiply them with some weights. These are the weights. What are the weights? For the child node 1, you take the ratio of fraction of data point in that node, in this child node divided by total number of data points. Similarly, you multiply the entropy of the second child node with

the fraction of the number of data point in that child node divided by total number of data points. So, let me do the calculation.

(Refer Slide Time: 20:07)



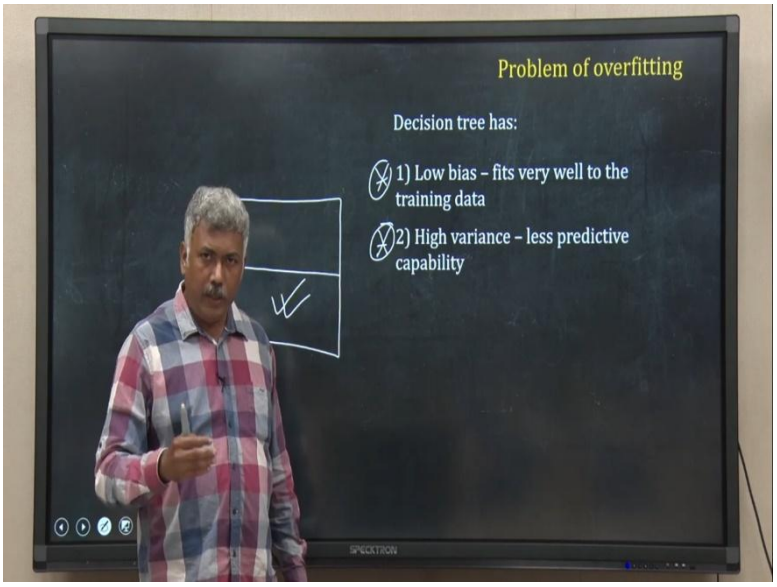
So, what we have? We have, rather than taking fraction what I have done here? I have taken the original number that does not make any change in calculation, only the numerical values are different. So, my parent node's entropy is 0.95, here it is written. The entropy of one child is 0.94, the other one is 0.24. And I have multiplied by this weight and the weight here.

For the second one weight is 49 divided by 108. Because I have 49 number data points here, 47 plus 2 is 49. And here I have 108 data point originally. So, this weight is nothing but the 49 divided by 108. Similarly, this weight is 59 divided by 180. So, simple, you are just multiplying with the relative weight of each of this child node depending upon how many data points are there in these two segments.

So, now, do the calculation. You will get 0.33. So, now, suppose, you has another question. In place of this G2 less equal to 3, you ask your alternate question. Suppose you say ok, G2 is bigger than 4. Then you do the same calculation again and check out what is the information gain.

And if the information gain for this question is less than the information gain of this question, I will keep the previous question, because in that case we have got more information. That means we have got better segregation of heterogeneous population into more homogeneous sub population.

(Refer Slide Time: 21:47)



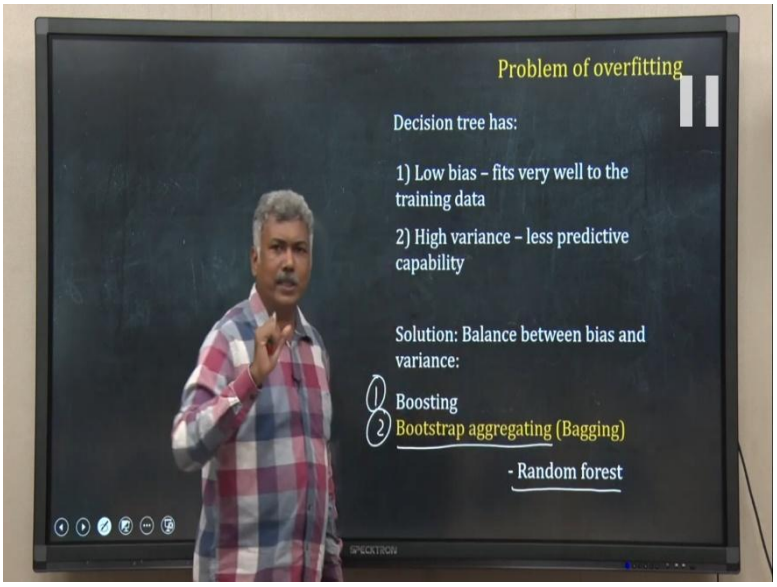
So, now, look into some lacunae or trouble with this approach that we are using. As I said earlier, what we are doing. In this method, this algorithm, we are working locally. So, suppose in our example, what we have done, this is my data space. We divided the data first in two parts.

And the second question, we work only on this segment, not on the whole data. So, essentially, the algorithm is deciding locally, on the one segment of the data and it is not deciding the fate based on the whole data throughout the algorithm, throughout the process, throughout the steps. So, that mean that is why this algorithm creates a tree which is actually over fitted.

So, as you remember, we have discussed the problem of over fitting earlier also. So, it has a low bias. That means it fits very well with the training data set that you have given. So, it has a low bias, whereas as it is low bias and over fitted, it will have high variance. That means if I have a completely new test set, it may not predict it properly in that case for those test samples.

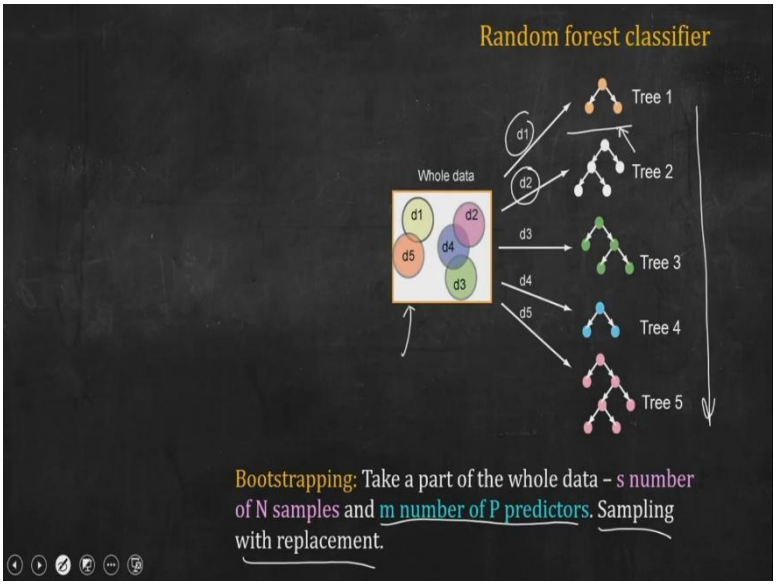
So, it will have less predictive capability. So, it has high variance. Now, this is not a unique problem for our decision tree classifier. This problem is there in most of the machine learning algorithms and there are ways to reduce this problem. For example, in case of decision tree classifier, we usually use two approach.

(Refer Slide Time: 23:24)



One is called boosting; another one is called bagging. Bagging is nothing but bootstrap aggregating. So, you will first bootstrap and then you will aggregate the data. And one of the example, one of the algorithm which use bagging in decision tree classifier is random forest and I will briefly discuss that. The principle of drawing the tree will be same but we will change it a bit how we build the tree.

(Refer Slide Time: 23:50)



So, what you do here? Suppose this whole square is a whole data set. So, the first step is bootstrapping. In bootstrapping, you do not take the whole training data set to train your

machine-learning algorithm, you do not do that. What you do? You divide this data into multiple training data.

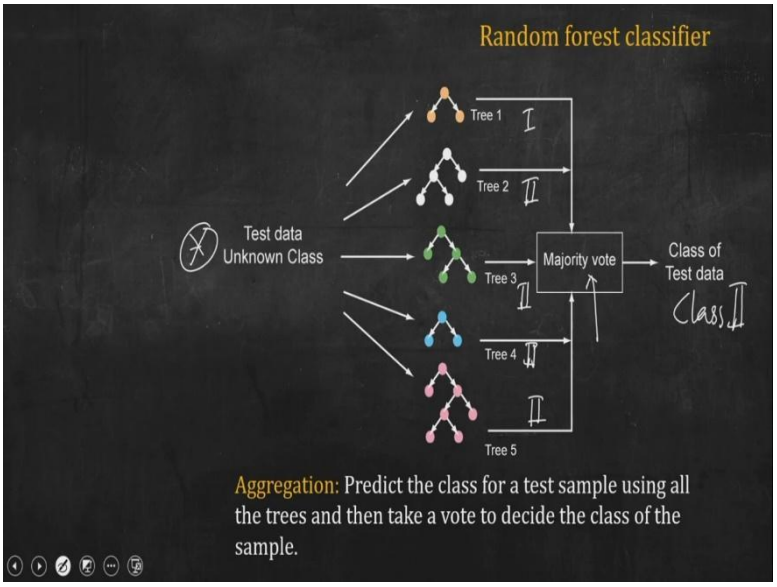
So, from one large training data set you create some subsets and you create those subset by sampling with replacement. So, that is what I have shown here d1, d2, d3, d4. These are the subsets of the training data set. Now, you take one of those subset data and create a decision tree.

Now, when you are creating this decision tree, again, you are not taking all the predictors. Suppose, in your training data, set there are 10 predictors or trained independent variable on which the dependent variable or the response variable depends. So, x1, x2, x3, something like that up to x10.

So, when you are building this first tree, you will not take all the predictor information for that data d1. You will select as part of them. Suppose, two third of those predictors. Similarly, you now take data subset 2 and create the tree 2. Now, again you do not take all the predictor, you take a subset of those predictors. So, in this way I have shown 5 trees. You can make hundreds of such trees and every case you are choosing, jumbling of the which parameters or which predictor you will choose to build this tree.

So, in this way what you have done, ok, maybe you have 20 or 30 predictor in your whole data but none of this tree use all those predictor together. So, in technical term, these trees are generated from bootstrap data because you have to use a subset of your data as well as they are decorrelated. Because you have not used all the predictor. You have used m number of predictor out of this total number of p, the total number of predictor that you have used. So, this is how you create the forest, forest of decision tree. So, now, you have to use this forest to classify a test data.

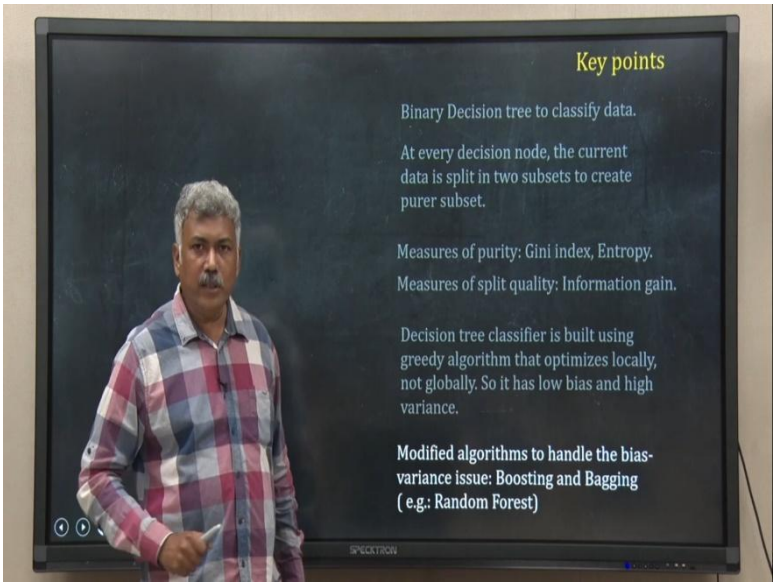
(Refer Slide Time: 26:10)



So, that is where the aggregation part come of bagging. So, you have a test data, unknown sample. You do not know the class of that. So, you feed that to each of this tree. And each of this tree will classify the data. So, suppose it say, the tree 1 has said it is class 1, the tree 2 has said is class 2, tree 3 has said is class 2, 4 has says also class 2, 5 has also said class 2.

Now, you will aggregate these decisions made by these multiple decision tree and you will use the principle of majority vote to decide the class of the test data. In this example, out of 5 these trees, 4 have said this data belong to class 2, whereas one tree has said it belongs to class 1, that is minority. So, I will say, my test data belong to class 2. So, that is how we build a random forest and we use it to classify our data.

(Refer Slide Time: 27:17)



That is all for this lecture. Let me jot down what we have learned in this lecture. We have learnt about decision trees which are used for classification and these trees are binary tree. What we are doing here? In this tree at every decision node, we are dividing the data, we are splitting it to two subset to create some purer subsets.

Now, when I say purer subset I need some measure of purity, that is why we have learned about two measures of purity. One is called Gini index, another one is called entropy. And at every decision node I can ask multiple question and I have to choose which question suits best for my tree. And so these questions and their outcomes are compared using what we call information gain and we have discussed that also.

Then we have discussed that this decision tree works locally. So, it is a greedy algorithm and it usually becomes over fitted. So, we have to be very careful that decision tree classifier can be over fitted. So, they may have less predictive capability. To solve this problem, there are method using boosting and bagging. One of the bagging technique that we have learnt is the random forest technique. That is all for this lecture. See you in the next one.