So till now. I hope you are able to understand all the previous algorithm in depth maths. Intuition. Now from this particular video I'm going to start something called as decision tree algorithm. It is a very important algorithm altogether because in the upcoming videos we'll be seeing ensemble techniques, bagging and boosting. Uh, over there. Some of the algorithms like Random Forest Boost all uses Decision Tree internally. Okay. So let's go ahead and let's try to understand the maths intuition behind decision tree. So to begin with guys our decision tree classifier. And right now see decision tree can be used for solving both the kinds of problem statement that is classification and regression problem statement. In this video we'll focus on how to solve a classification problem statement with the help of decision tree. And with respect to this, we are going to discuss some of the important topics like Entropy and Gini index, which is specifically used for to check priority split. And then we'll go and understand about information gain. Uh, information gain and all. I'll try to just explain, but let's start with some simple understanding. Usually in Decision Tree there are two types. One is ID3 and one is CART. Currently the sklearn library that you probably use for, uh. Creating the decision tree algorithm through Python code uses this cart technique. Okay. So we are going to focus on both the techniques. But in short see the basic difference between ID3 and CART is that in CART you just create. Whenever you're creating a decision tree, you create only binary splits okay. With respect to each and every node, you basically create a binary splits, whereas in ID3 it does not happen like that. You can also create more than a binary split like you may also have three splits from one specific root node. Okay. So this is the basic difference. And yes, some of the mathematical uh formulas will be changing with respect to this. Okay. But let's start and let's understand like what exactly Decision Tree is and why it is basically used now I hope you probably may know many, many programming language. So let me just write a very simple programming language. Let's say I'm using Python. So let's say I'm going to write age is equal to 14. And I'm going to say if age is less than or equal to 14 I'm just going to say that print okay. The the person is in school. Okay. So if I'm just saying that if, uh, he or she is basically he or she's age is basically less than 15. I'm just saying that a person is in school. Okay. So this is one condition, okay. And let's say after that I write an if block. So if I say if the person age is greater than 15 and less than or equal to. Less than or equal to 21. I will probably say that the person may be in college, okay. The person may be in college. Okay. I'm just giving you an example, guys, everything will make sense why I'm writing the specific code. Okay. And finally else I'm just saying that the person has passed, okay? The person has passed the college. Okay. So I'm just giving this specific statement. Now understand if I really want to compare decision tree right decision tree will also be working similarly like this. If else if condition. Right. So let's say I'm going to probably take a root node okay. And here let's say my root node over here will be first condition. Age is less than or equal to 15 okay. So obviously I will be having two outputs. Either it can be yes or it can be no. Right now in this particular case, if the age is less than or equal to 15, what we are going to do, we are going to basically create another root node. And here it is basically going to print. The person is in school. Right. So here obviously it is going to basically print whether the person is in school. Right. Very simple. Now, similarly, in the case of no I may have a different root node. Okay. And over here what it will be that, uh, this no basically means we are basically going to, uh, move towards the left block. Okay. Or instead of writing yes or no in this way, I will just change this. I will make sure that I will write. Yes over here and no over here. Okay. So left hand side will just try to write like this. So here I can definitely say that the person uh, when it is. Yes. In this particular case it is going to print. The person is in school. Okay, so person is in school. It is going to print. In the case of no, what will happen is that we're just going to go to the Elif block. Now in the Elif block I will be writing another condition here. The condition will be age is greater than 15 and age is less than or equal to 21. So with respect to this, again I can have two more divisions and this two specific division will be again, yes. Let's say this is yes and this is no. So this can either be true or false. If it is no it will probably print something. Otherwise it will come over here. And again we can do this specific split from this particular. Yes. Uh or sorry from this particular. No. Which will be my else block. Okay. So here you can basically see that like this the same if else condition. I have actually constructed a simple decision tree. And this is how a decision tree will get constructed uh, from this kind of conditions itself. So suppose if I have a specific data set, our main aim is that if I have some independent and dependent features, we are going to make sure that we try to take that particular data set, and we'll try to construct a decision tree in this specific way. But just to construct this, you really need to know some of the technical terms like entropy, Gini index and all information gain and all. Okay. So uh, in short decision tree will try to create in this bottom to top approach okay. So let's go ahead and let's talk about this and how a decision tree will basically get constructed. What does entropy Gini index information gain actually mean. So here I'm going to take a very simple example. So this I have a specific data set. Let's say this particular data set I have already spoken about this particular data set in a bias. This data set is very simple. Here you have independent features like outlook, temperature, humidity and wind. And you really need to predict whether the person is going to play tennis or not based on this independent feature. So obviously I'll be using decision tree classifier. This is a classification problem, specifically a binary classification problem. I will try to solve it okay. Binary classification problem. Now let's start okay. Let's start. And let's see how we are going to split this all nodes and how we are going to construct the decision tree. So in the first instance I am just going to take outlook as my node okay. So let's say this is my outlook I'm taking outlook as my first feature okay. Or the root node. Now you may be thinking Chris why you did not take temperature humidity, wind or anything as well. Don't worry guys, I'll explain you about that also. But right now let's consider that I have selected outlook. Now over here with respect to outlook, how many unique categories are there? So if you go and see Sony's one unit category, overcast is one unit category and rain is one unit category. So based on the outlook I am going to divide this into three splits. Okay. So the first split over here is basically my sunny okay. The second is basically overcast. And the third is something called as rainy. Okay. So based on the outlook, you can definitely see that I'm getting this many number of uh, this number number of sub categories from this particular root node. Now with respect to outlook, first of all try to find out how many yes and nos are there. So if I start counting one. Yes two years three years, four years five years, six years 789. So total number of yes that I have over here is nine. Yes. And total number of nos that I basically have 12345 okay. So here I have nine years and five nos okay. Now with respect to outlook when I split it with respect to sunny. Considering sunny as a category. Now go and find out like how many number of yes and nos are with respect to sunny. So sunny over here two nos. I can see that this is three. No. And obviously if I probably try to see one. Yes. Over here and the other is over here. So definitely I can, uh, you know, just by seeing this, uh, you can just go and count it ahead and you will basically be finding I will be getting around two yes and three nos. Right. Because, see, this is one no, two no. And one more no is somewhere here with respect to sunny. So somewhere here. So I have three nos and two. Yes. So I have one. Yes. Over here and the other I see over here. So you just have to count and just try to find out that with respect to sunny, how many. Yes. And nos are there. Similarly with respect to overcast, if you go ahead and find out with respect to overcast, I will all be getting. Yes. So one yes two years, three years and four years. So here I'm going to get four yes and zero nos. Similarly with respect to rain you just go and calculate I'm getting three yes and two nos okay. So here you can clearly see that I've, I've been able to find out like how many number of categories are left with respect to this particular split. Now, now the next thing is that with respect to sunny you can see that I'm having two yes and three nos. So this is basically a impure split. Why impure split. Because I have both yes and nos. Right. So this is going to be an impure split. So what I'm actually going to do I'm going to further split this down into some more categories. Be considering other other, let's say considering other features. Suppose I may consider, uh, temperature. So I may again split it. Why I'm splitting it because this is a impure split. But if I see in this particular case, this is a complete pure split. Why I'm saying pure split. Because here either I have yes or no. So here I see that I have four yes, but zero nos. So all my output over here is going to be yes itself. Right. So suppose if I traverse this, suppose if my outlook is overcast, I'm definitely going to say that I'm going to play right. So we should. This pure split is always good because we will be able to get the leaf node. So this node is basically called as leaf node okay. Leaf node basically means either you have yes or nos right. Similarly in this particular in case you are basically having an uh, impure split. So further you really need to split this up. Okay. Now this kind of split will keep on happening in Decision tree unless and until we don't get a leaf node. Okay. We don't get a leaf node. Now, obviously, just by seeing you can definitely find out whether this is a pure and impure split. But mathematically, how do we find out whether this split is pure split or impure split. So for that particular case we will be using a technique which is called as entropy and information gain. So let's go ahead. And here I'm going to just uh write down two important points. First of all first of all purity. Right. If I really want to check whether it is a pure or impure split, impure split. So how do I check it out? Okay. For this, mathematically we have to use two techniques. One is entropy and the other one is something called as Gini index. Okay. Gini index or Gini impurity okay. So I can also say it as Gini impurity okay. So these are the two techniques that I'm specifically going to use. Uh let me just write it down. This as Gini impurity okay. Gini impurity. So we'll try to understand mathematically how this specific things will happen. Now coming to the second thing is that I just asked you a question. Why did you may probably be thinking, Krish, why did you just select outlook? So what feature you probably need to select for the splitting purpose okay. So second question is this what feature. You need to select for splitting. Because you can you can select any feature. Right. So how do I understand that. Which is my best feature to select for a splitting purpose. Because if I select this particular feature and if I'm able to get the leaf node quickly, that basically means this particular feature should be selected. And it is correct. Right? But if I select a feature that may not give you a leaf node, it may take time to give you a leaf node, then that is bad, right? So for this particular case we use something called as information gain. So now this two specific thing we are going to understand. First we'll start with entropy and Gini impurity. And then we'll start with information gain. Okay. So this is with respect to the introduction of decision tree classifier. So in the next video we are going to discuss about entropy and information.

Let's continue our discussion with respect to decision tree classifier. And uh, in this video we are going to discuss about this two important points. That is purity. How to check whether a split is pure or not. And for that we are going to use entropy and Gini impurity okay. So let's quickly write down the formula and then I'll start explaining the formula itself. So the first thing on the left hand side we'll try to understand about entropy. And the second thing that we are going to see is something about Guinea impurity okay. So this is Guinea impurity. Now with respect to entropy, if I really want to see the formula, the formula is basically given by h of s r minus p plus log base £0.02 plus minus. P minus. Log base £0.02 minus. Now what is this p plus and p minus. So let's consider that we have a binary classification problem. Now in this particular case what will be p plus p plus is nothing but the probability of being positive like probability of one category like or the positive category. Or it can be a negative category. So here if I say one and zero. So if I really want to talk about if my outputs are basically having ones and zeros, I may say that p plus is nothing, but the probability of being one p minus can be the probability of being, um, you know, zero. So like something like that. Suppose if I have multi-class classification, then instead of writing minus p plus, I can write minus. What is the probability of being category one? What is the probability of being category two. What is the probability of being category three. So it will be like it will keep on going like minus minus minus sign. So that is the basic difference between the formula okay. And similarly if I really want to find out the Gini impurity then I will probably be using one minus summation of I is equal to one to n and this will be p square. I'll talk about what exactly is p square. It is the combination of both the positive probability and negative probability. So like it's like p plus and p minus I'll just expand this equation in some time okay. Now let's go ahead and understand first of all entropy. And then we really need to understand many things in entropy also. So I hope uh with respect to binary classification you have understood the formula P plus basically means it is just the probability of being one category. And this p minus is the probability of being the other category. Okay. Now let's say that, uh, I have a split, let's say with respect to this feature, what I'm doing, this particular feature split. And let's say initially I had three. Yes. And uh, three yes. And three. Sorry. Six yes. And three nos. So when I made the split, let's say I'm going to get two categories. It is nothing but this first category C one okay. And the second category C two. Now with respect to this split let's say I am getting three yes and three nos over here okay. And similarly I'm actually getting three yes and zero nos over here okay. I'm just saying that after this particular split in the category one, uh, I can see that total number of yes and nos are three yes and three nos. Like how we did this particular split over here the same way we are doing this particular split. Okay. Now what we are going to do is that we are going to check, we are going to check the purity split for this. And here you can definitely see that, okay, this is an impure split because I have 50 percentage which is saying like yes and 50 percentage as no. So this is obviously a very impure split right. Very impure split. So and this in this particular case, you can say that this is a pure split. Just by seeing it we are able to understand. But mathematically let's go ahead and apply it. So I'm just going to write H of S. Now here I'm just going to see h of c one and c of c one basically means what is the entropy of this specific uh node okay. So h of c one the entropy will be I'll just write the formula minus p plus log base £0.02 plus minus p minus log base £0.02 minus. Right. So this is the formula that we are going to use. Now what exactly is p plus. Now p plus is nothing. But here you can clearly see that. What is the probability of being. Yes. Right. So obviously when you see this the probability of being yes is three by six. Right. Three by six. Because our total number of yes and nos if I combine it is six. So here I can basically write minus three by six log base two three by six okay. And similarly what is the probability of being negative over here. Again the same thing. Uh so here I can definitely say that okay. This is uh nothing but three by six log base two three by six. Right. Because the probability of no is also that many. So if I go ahead and compute this, uh, log base two one by two, and if I start computing it, and if I try to apply the log base two formula here, I'm going to get one. Okay. Now see guys, since this is a very impure split because the number of yes and nos are equal. Right. So obviously my maximum maximum impure split value will have one. So this is the reason why

I say that this is a very impure split. And with the help of entropy we are able to find out that okay, fine. It is one. Now in this particular case here you can directly see that this is a pure split, because here I have only yes and zero nos like three yes and zero no. So let's go ahead and apply this specific formula for that too. So here I have a SC of two. So is nothing. But uh what is probability of plus over here. So it is nothing but three divided by three right. Because the total number of uh nodes are zero. So if I combine both it is going to be three only. So here I'm going to have minus three by three log base two three by three. Then it will be minus what is the probability of being. No it is zero right. Because zero by three is nothing but zero right. So here I'm going to basically use zero log base to zero. So this in turn uh will be uh if I try to just compute this, this will be zero itself because zero multiplied by anything and this will be minus one log base two one. So this in turn will be giving you a. Zero value. So here, since I'm getting a zero, this is basically a pure split. And obviously you can see over here since I have only yes I'm actually getting a pure split. Okay. Now with respect to this, let me create a very good draft graph so that you'll be able to understand entropy. So out of this I can definitely come to a conclusion. Let's say this is my conclusion that I'm coming up with uh, over here in the x axis I can have probability of plus or minus. And obviously this value will be between 0 to 1. So if I say probability of plus is 0.5 that basically means probability of minus is point five. Now whenever I have probability of plus is point five you know that. See in this particular case probability of yes or probability of no is point five. So the maximum value that we can get is one. So in the y axis I will try to plot h off that specific category. It can be. Yes okay. Now in this particular case whenever my probability of plus or probability of minus is point five, I am always going to get my value as one okay. That is a complete impure split okay. And with respect to zero to point five it will differ 0.5 to point one it will differ. So finally I will be able to get this. Sorry, I will be able to get this kind of curve. Okay. When we are trying to plot probability of plus with h of S okay. So maximum value will be one. And when it is point five I'm going to get this. So this is the curve that we usually get with respect to entropy. Your values will be always between 0 to 1 that you always need to remember. Now if you are able to get this kind of graph, I hope you are able to understand it, because why I'm teaching you this graph is that because in the interview, you know people, the interviewer may ask you what is the differences between entropy and your Gini impurity? Okay. So this kind of questions may be definitely there. Okay. So you will be able to say this that your guinea your you know, your entropy basically ranges between 0 to 1. Okay. Now let's go ahead and try to see for Gini impurity. So for in the case of Gini impurity how we will be able to see. And I'm just going to repeat that same formula what I had actually written okay. So Gini impurity is nothing but one minus summation of I is equal to one to n, n is the number of data points. And this is p square. I had written this p square. Now let me expand this. So it is nothing but. P plus whole square plus P minus whole square. Now what is this? This is probability of being. Yes. This is probability of being no. Right. So in this particular case whenever I probably if I'm applying for this particular split uh gini impurity. So for if I have three years and three no's how do I calculate what is the probability of yes. Over here it is nothing but three by six which is nothing but one by two. So let's go over here. I will write here one one by two whole square plus one by two whole square, because three by six is nothing but one by two. So when I try to compute this, this is nothing but one minus half. It is 0.5. So for an impure split which is a complete impure split, complete impure split the maximum value with respect to Gini impurity, I will be getting somewhere around 0.5. That is what from this particular formula we are able to see right. And we will be able to see that how we will be getting a graph which looks like this for Gini impurity. So your in the in the case of Gini impurity your value will be ranging between zero to point five. In the case of entropy it will be ranging between 0 to 1. If you don't believe me, just try to apply for this particular split. So for this particular split let's say if I have three yes and zero no's. So if I try to apply this, this is nothing but one minus three by three. Uh, whole square okay. And this is nothing but one minus one is zero. I will not have probability of minus because it is zero. Right. So zero by anything is zero. So I'm not writing it over here. So here you can see the for a for a pure split I'm getting it as zero for a very impure split I'm getting as one. So this will again be ranging between 0 to 1. Right. So this is the basic difference with respect to understanding about entropy and Gini impurity. Now my next question is that Krish uh why why why did you suppose if you have suppose if you have three features f1, f2, f3, why did you first of all select F1 and probably make a split to f2 and F3? Why? I could not make a selection of F3 and make a split for F2 and F1, and probably in the future I can probably make again a split with F2 and F3. How do I decide which features to basically select to make this decision? Tree split okay. Decision tree split. So for this we will try to understand about information gain right. Information gain. So uh and with this help of information gain we'll be able to understand that which features should I select and probably do the decision tree split with the help of Gini index, Gini impurity and entropy. You are definitely able to find out like whether your this node split is pure split or not split uh, pure on impure split. If it is a impure split, you further do this particular split right. In this particular case you don't do more splits. You may take another feature and make, uh, do this particular split. But now and I hope you have understood what is the difference between entropy and Gini impurity. But now my question is that which features to select how we should select each and every feature and start doing the splits? Uh, after we understand whether it is a pure or impure split. So for that we'll be using information gain. So in the next session we are going to discuss about information gain okay. So I hope just revise. This particular session is super important with respect to different different interview questions. You may be getting different different questions right. So yes, I'll see you all in the next video. Thank you.

Now let's continue the discussion with respect to Decision Tree. In our previous video, we have already discussed about Gini index and Gini impurity and entropy. And we understood that okay, to check the purity split we basically use this. But now let's discuss about information gain. Let's say in my data set I have features like f one, f two, f three. And probably I have my output feature. I may take f one and start splitting as a decision tree. Right. Or I may take probably f two and start splitting in terms of the decision tree. Right. So how do I determine from which feature should I select in order to start the split. And for that specific case we will be using information gain. Now let me quickly write down the formula for information gain. So here we basically write gain as comma feature one. That basically means whether I need to start with this feature or this feature is equal to. This is basically the entropy of the root node. Okay. Minus summation of v belongs to value. I will explain each and every term. Don't worry about it. What exactly it is. First of all, let me just write down the formula. And this is entropy with respect to all the categories okay. Now what exactly this is. This is nothing but entropy. Entropy of the root node. Okay. Entropy of the root node. Now, let me, uh, quickly talk about this. And obviously you know that. What is h of S? H of S for a binary classification, I can basically write the formula as probability of log uh, probability of plus to multiply by log base two plus. Similarly probability of minus log base £0.02 minus. So this is the formula that we specifically use. Now let's say if I am trying to do a split okay. Let's say I'm trying to do a split over here okay. And uh we'll try to compare this two specific split okay. So let's say my first split is with F1 okay. Let's say I have this F1 split and here I'm just making into two categories C1 and C2. Okay. And let's say initially in F1 I had nine yes and five nos. And here I actually have six years and two nos. And here I actually have three yes and three nos. Okay. So let's say that this is what is the outcome that I probably get. Okay. And similarly uh, let's say there will be another split which is with F2. And here I am getting, getting category C3 and C4. Now I really need to compare out of this two, which is the best split? Should I start with F2 or should I start with F1? Now let's consider this one and try to put up and try to find out the formula what game we are actually getting. Okay, so here you can definitely see that it is nine years and five. No six years and two nos. And three years and three nos. So first of all we'll try to calculate the entropy of the root node. So here this entropy of the root node will try to calculate. So here we have already minus p plus minus p plus is nothing. But over here in this particular case it will become nine by 14 log base two nine by 14. And here I basically have probability of minus. So in short I will be getting somewhere around uh uh, you know, with respect to this. No, I will be getting five by 14 log base two five by 14. Okay. So with respect to in order to calculate and this is the entropy for the root node okay. So if I try to calculate it I have already done the calculation. This will approximately be equal to .94. So this is the first step I have actually found out what is the entropy of the root node. And I've got somewhere around .94. Now let's go ahead and let's try to find out this specific thing. Now here in this category this is my first category C1I have six yes and two nos. This is my second category C two. Uh I have three yes and three no's. Now whenever I say h of s v. In short I am trying to find out the entropy of both these categories. Okay. So let's quickly go ahead and calculate the entropy of both these categories. So here I'm going to write h of c one. And again I'll try to apply the same formula. In this particular case what is the probability of. Plus it is nothing but six by eight right. So here I'm going to get six by eight because there are total six. Yes. And uh two nos. Then I here I'm going to write log base two six by eight. Similarly here minus when I'm doing with respect to the probability of negative, what is negative or probability of no. In this particular case I'll get two by eight log base two two by eight. Right. So finally you'll be able to see that what is the output that I'm going to get. It is nothing but .81. So this is what the uh entropy of category one. Similarly I'll go ahead and calculate the entropy of category two. So in this particular case entropy of category two is basically having three yes and three nos. And obviously I didn't I don't think I need to solve this because this is a complete impure split. Right. So if it is a impure split I know and this is a very, very impure split, the worst impure split, I know with respect to this I'm always going to get one. So I have actually found out the entropy of category two, uh, as one and this as .81. Okay. Now, fine. I've got this information for both the categories. Now let's understand what is this s of and s and I will start uh putting up all these values okay. So finally let's go ahead and write the gain. So gain with respect to s comma s one. First thing is h of s h of s is nothing but .94 minus. Now here it is something called as summation. Now when I say summation, that basically means I need to find out all the summation. So here I'm just going to use this bracket. What is s of v s of v. See, whenever I write s as basically means the entire sample from feature one, then whenever I say s of v that basically means we are trying to check this category and this category. So S of V will just try to find out that after the split, how many number of yes and nos are there. So here you can see eight total total. If I try try to calculate it is six yes and two no. So total eight is there. And over here total 14 was there. So here 14 is the total here eight is the total here. How much. Six is the total right. So S of V will be eight divided by 14 okay. For the first case. So here I have eight multiplier divided by 14 and multiplied by h of s v h of c one is how much h of c one is nothing but .81. So I'm going to multiply this with .81. Then here I'm going to have. Plus since it is summation. And here you can basically see with respect to the next category. What is this of V. It is nothing but six six divided by 14. So six by 14 multiplied by one. Okay. So here I will be probably getting a value. If I compute it it is somewhere around .049. So this is my guess. Uh gain of s comma s one okay. So here I'm just going to highlight it. So this is my total gain. Okay. With respect to s comma F1, now I have my next split. Right now let's say this is my next split with respect to f2 f2 feature I will be having C3 and I'll be having C4. And similarly we can go with more further splits as many as we want. And now if I go and compute the gain of s comma f2 and let's say if I get a greater value than this. So in this particular case, let's say after computing all this thing, you know, and I'm not restricting it to just one depth, guys, it can be any number of depth. So that many number of categories automatically this formula will get extended sometimes using summation right summation of this one. Right now in this particular case here you can see that gain of S of F2 is greater than gain of s comma f1. So here this is .049. Right now in this particular case this gain is more. So this indicates that I have to start my splitting from feature F2 right. So that entire combination can be checked upon. Decision tree internal checks. All this particular condition if I'm specifically using entropy or Guinea impurity, and it will then try to find out which feature is best for splitting. So this is how the entire information gain is basically calculated. This basically calculated okay. Calculated okay. Now one very important thing that we have probably missed in our previous class is that when should we use entropy and when should we use Gini index. So this is a very important question. Uh, when should we use entropy? And when should we use guinea impurity? Okay, so this part I will try to complete in my next video. So I hope you have understood what exactly is information gain. Okay. And you have got an idea about information gain completely uh, and how it is basically calculated. But at the end of the day it is just helping you to decide or it is just helping the decision tree to decide which features we should select and we should start splitting. Okay. Now in the next video, we'll discuss about when to use entropy and when to use information gain. So yes, I'll see you all in the next video. Thank you.

Now let's understand when should we use entropy and when should we use Gini impurity? Now this is super important because we discussed about two different techniques to check the purity split. Right. And that is entropy and Gini impurity. And obviously you know the formula H of S is nothing but probability of plus log two probability of plus minus this. This is standard formula itself. Now let's say if my categories are three categories as an output okay. Now in this particular case my this equation will get expanded. And here I'm basically writing minus probability of category one log base two probability of category one minus probability of category two log base two probability of category two. And similarly minus probability of category three log base two probability of category three. So as the number of output categories will keep on increasing, this will be increasing. So this uh is what. So that you should not get confused with respect to this particular formula. So similarly this Gini impurity will also get expanded based on as the number of output categories get expanded. Now let's go to the question when should we use Gini impurity and when should we use entropy. Now see guys with respect to the entropy the formula uses log okay. So here a simple simple easy way to talk about entropy is that whenever your data set. As small. Okay, you can definitely use entropy. Now people will say how small? Okay. So I'm just saying okay, let's say if you just have 10,000 records you can definitely go with entropy. You know, because if you are trying to implement with the entropy, right, it will obviously get take time because here there is log okay. With respect to small data sets. Uh, the time difference will be pretty, uh, the difference between the time, uh, that is going to take to train the model. It will be almost minimal, right. For between entropy and Gini impurity. So when the data set is large definitely try to use Gini impurity okay. So this is just the solution that I really want to give. Uh but again how much large how much small? Again, that is a question. Uh, you should try to think over it. Uh, in short, uh, my suggestion would be that by default, whenever you use any decision tree classifier, they use Gini impurity. Okay. So this is just a suggestion to make in front of you with respect to when to use entropy versus when to use Gini impurity. But by default, if you are using decision tree classifier, Gini impurity is basically used. And I think for most of the scenario or problem statement we use Gini impurity. Okay. So now let's go ahead and understand uh in our next video is that when our features right. Let's say in this particular thing the features were just categorical variables. But what if our feature is over here a continuous variable variable okay. Continuous values are there. So how do we make decision tree to split this specific values. That is what we are going to see okay. So yes uh I will see you all in the next video. Thank.

We are going to continue the discussion with respect to decision trees. Uh, in our previous when we are discussing about, uh, decision tree classifier, in our previous video, we saw this particular problem statement. And here one thing that you could note that all the features were basically categorical features. Right. And this is specifically by output feature. Now here you can basically see categorical features. We could easily divide those categories based on each and every split. What if I get a feature that is specifically continuous like this. Now in this particular scenario how should we go ahead and probably do the decision split. So that is the reason the topic name is decision tree split for numerical features. Right. So let's go ahead and understand now in this first step is that first of all we will sort the feature values okay. This is the first step. We basically say feature values. Now over here by default I made sure that all this particular values are sorted already. So you can see this is sorted in the ascending order okay. Specifically sort the feature value based on ascending order. Now what we do is that we create many decision trees okay. So in every decision tree let's say this is my first decision tree I am going to pick up my threshold. Now let's say this threshold over here is my first value. So I am going to do the first split from here. Right. So I'm going to make 2.3. So here this basically means my root node will be less than or equal to 2.3. Any value that is less than or equal to 2.3, there will be one instance like this. And the other instance will go like this. So if it is less than or 2.2 uh, less than or equal to 2.3, let's say this is yes. And this is no. Now in this particular scenario, you know that I will be getting what output I will be getting one. Yes. Output. So in this particular case how many. Yes I have I basically have one yes. And zero nos because I don't have any nos over here, whichever values are less than 2.3. Now similarly I will go and create this specific node over here. Now in this node how many yeses are there. 123. So three. Yes. And how many nos? 1233.

Nos. Right. So here you can definitely see that initially I had four yes. And three nos. After the split with respect to less than or equal to 2.3, I made a binary split. And in the left hand side you can see how many yes I got and how many no I got. So this is one decision tree we initially construct. And let's say if there are many features we further split this into many, many, uh, splits. Uh, that is not a problem. Right. And based on the entropy and all, uh, we can basically do the further splitting. Now, let's say after this, the second decision tree will be made sure that how now the threshold will be changed. Now we are going to keep the threshold as 3.6. Now what will happen is that now my threshold will get updated. It is 3.6. And now I will create a node which will basically be having less than or equal to 3.6. Now when I say less than or equal to 3.6, again there will be two splits okay. Yes or no now, whichever is less than 2.3.6. That basically means this two records are less than 3.6. So here obviously I'm going to get two yes and zero nos. In this particular case how many yes and zero nos I'm going to get 1 to 2 years and three nos I'm going to get right now like this. I will keep on creating multiple decision trees with different. If I'll just move this to the next step and then this will be in one side, and then this will be in next side. Like that I'll create one multiple decision tree. But now to understand whether this split is better and this split is better, how do I actually get it? It is very simple. We basically use the information gain. So whichever has the highest information gain we will be selecting that specific split. Let's say if I have ten different splits with different different values, whichever will be the best, whichever split will be having the best information gain from there, let's say from uh, from this particular value, if I'm doing the splitting right and I get the best information gain out of this specific split. So I am going to basically split my node by taking four as my root node. So less than or equal to four as my root node I'm actually going to consider. So this is how actually it happens in case of a decision tree split for numerical feature. Now there is one major disadvantage. Guys just understand this particular split when we are doing the comparing with so many decision tree. What happens if we have in our data set millions of records. So the time complexity is usually high okay. The time complexity is usually high for this particular process okay. So this is the disadvantage uh with respect to this. But again since we are using Decision tree, uh, we really need to uh, use this technique or internally, it uses this particular technique with respect to all the numerical features. Right. So I hope you have understood this particular video. Uh, yes. Uh, keep on. If you are not understanding anything, please revise this particular video and try to gain that momentum of understanding the in-depth maths intuition. Thank you. I'll see you all in the next video.

So guys, in this video we are going to discuss about a very important topic which is called as Post-pruning and Pre-pruning in Decision Trees. Now, just by the meaning of pruning, I think you will be able to understand something at least. Okay, so let's discuss what exactly is this post pruning and pre pruning and let us understand why do we do post pruning and pre pruning also. So guys suppose if I have a training data set and if I probably use a decision tree for this particular training data set. So and if I use the by default parameters that are available in decision tree, it will start splitting it. And it will make sure that it will keep on splitting unless and until we don't get the leaf node. Okay. So let's say here I'm actually getting the leaf node. Uh, unless and until we don't get the entire leaf node it'll keep on splitting. Right. So this is also a leaf node. But here we quickly got that specific leaf node here. Till till this particular part. When we went here we got the leaf node. Now let's consider that at this particular instance if I have nine years and two nodes as my output category, that is remaining. And what did I do? I did a further split and made sure that I get nine years and zero nodes. And since this is an impure split and here I probably got, uh, you know, zero. Yes. And two no's. Right. So here this also became a pure split and this also became a pure split. And finally I was able to get a leaf nodes like this. Right. So this is also a leaf node and this is also a leaf node. Now understand something about pruning okay. Pruning whenever we talk about right. You have seen gardener. You know whenever some plants you know they grow right. They just try to prune it so that the shape is maintained and it grows in a proper way. Right. So similarly in decision trees, also whenever I try to do this kind of split completely to its depth with respect to the training data set, usually we face a scenario which is called as overfitting. Now many times I've discussed about overfitting. Overfitting basically means your train accuracy. It gives very high because we have done the splitting completely till the end. But with respect to the test accuracy, my accuracy will be very, very low. Okay, so this scenario, we basically say it as overfitting. And with respect to overfitting. Again I'm repeating it. You usually get a scenario of low bias and high variance biases for training data. Variance is for test data right. Since over here the test accuracy is low. So I'm saying it as high variance. Okay. Now if I'm splitting my decision tree completely to its depth I usually face an overfitting scenario. Okay. Now how can I prevent this overfitting scenario. Because if I'm using the default parameters, you'll be able to see that I'll get my all decision tree will get splitted completely to its depth, right? So in order to reduce the overfitting, in order to reduce overfitting, we can use two different techniques. In order to reduce overfitting. We basically have two different techniques. One is, uh, specifically if I talk about this overfitting in order to reduce it, one technique is called as post pruning and the second one is pre pruning. Now what does post pruning mean? Post pruning basically means first of all I'll construct or I'll construct my complete decision tree. And then after that I'll start pruning it or cutting it. Okay. Now in this particular scenario let's say at this particular depth, you know, I was having nine. Yes. And two nos. And still I made this further split. Okay. But just understand over here in this scenario my number of output categories is nine years and two nos. And obviously the ratio of yes and no. In this particular case the ratio of yes is very high right over here because you have nine. Yes. And two nos very small number of nos. And yes. So I need not have to do the further split because whenever over here, if it comes right, whenever with respect to all this decision it comes over here, I should have selected. Okay. My output is basically yes. But still I did this particular further split. So in the post pruning what we can do is that we can basically cut the branch from here itself, because I don't want to further do the split till here, only I should be able to get since your maximum number of yes is there when compared to no. So I will say that. Okay, my my. I'll consider this as my leaf node and it will basically give me the output as yes. Right. So this is what happens in post pruning okay. In post pruning what we do in post pruning we first construct the decision tree. We first construct the decision tree. Construct the decision tree, and then in the second, we prune it. Prune it with respect to depth. With respect to depth. Okay. And why do we prune it so that it does not, uh, you know, lead to overfitting scenario? So in this scenario, the overfitting will not happen then. Then for the test data also it will be able to provide you the generic output okay. So this is the one. What about pre pruning guys okay. And one more thing about post pruning. This should post pruning should be applied for smaller data sets. Okay, because just imagine if I have millions of records. If I start creating Decision Tree till the end, it is going to take time, right? Whenever we have smaller data sets and like once we construct the entire decision tree and then we prune it, then the time complexity will be high, right? So we should definitely use post pruning for smaller data set. Now what about pre pruning. Now what is basically pre pruning mean right now pre pruning. Basically say that while you are constructing the decision tree okay. You don't have to construct a decision tree entirely like how we did it in post pruning. While you're constructing the decision tree you can play with some of the parameters. And yes, in decision tree you have some of the parameters like max features, right. You have some of the parameters like depth, max depth, you know. So these all parameters are there. Max depth okay. And there are also some more parameters with respect to uh, the split. How should the split happen. You know, all this features are basically some of the hyper parameters. Okay. So here in short in pre pruning. What we do, we play with the hyper parameter. We play with or we tune with hyper parameter while constructing the tree. We tune with hyper parameter. Or I can also write this as hyper parameter tuning while constructing decision tree. Hyperparameter tuning. While constructing Decision Tree. While constructing decision tree. This is super, super important. So that basically means when you are constructing a decision tree, you can basically, uh, play with those all these parameters. And through the hyper parameter tuning, you will probably be able to get what should be the max depth. What should, let's say max depth is 3 or 4. What should be a max number of features that you should probably take the different different parameters and if we refer to a sklearn, will be able to see all the hyper parameters that are actually used. Okay, max depth is there, max features is there. And there is also something called as split ratio and all and all. Okay, so let me do one thing. Let me quickly open the decision tree classifier algorithm and probably show you that specific part in uh, in this specific, uh, you know, uh, basically we're just going to see the sklearn documentation, right? So if I go to a sklearn decision tree classifier, just search for it here. You will be able to see that I'm getting an sklearn. And here you will be having many features. Right. So what all features you can play with one is criterion. You can use Guinea entropy. Log loss okay. Log loss was a part of uh logistic regression if you remember. Then you can use splitter. Like what splitter should be there best or random Maxdepth is one of the feature. You can play with that as an hyper parameter. And we use Gridsearchcv for doing it. You have minimum sample splits. You have minimum samples leaf. You have minimum weight fraction maximum features. So all these kind of hyper parameter you can play with it and you get hyper parameter tune it. So when the construction of the tree will automatically happen, this parameter can be set initially and your decision tree can be constructed. Right. So yes, uh, this was it to make you understand the differences between Post-pruning and pre-pruning, uh, or most of the things will be shown in the practical part. You don't have to worry about it. Uh, here we are. Just understood the maths in depth. Intuition. Okay, I'll see you all in the next video. Thank you.

In this video we are going to discuss about Decision Tree Regressor. In our previous videos we have completed Decision Tree Classifier. We have got to see that how classification problem statement can be solved using decision tree classifier. We have learnt about entropy information gain. We have understood about post post pruning, pre pruning and many more things now. Similarly, decision tree can also solve a regression problem statement. Now what do a regression problem statement basically mean here you can definitely see that our output feature is basically a continuous feature. Right now. Whenever our output feature is a continuous feature, at that point of time, I usually use a decision tree regressor. Obviously in decision tree classifier, you know, we used to do some kind of splits. Whenever we do some kind of binary splits, we actually say it as cart, you know, cart technique or off decision tree. And similarly we will be making this kind of splits now during the splits in decision Tree classifier. What we used to do is that we used to use techniques such as entropy, Gini impurity. Right. Gini impurity. Um, and we used to also use something called as information gain entropy. And Gini impurity is basically for the purity split. Information gain is basically to find out which feature should I use for making the split. Now in Decision Tree Regressor, I cannot use this because, uh, whenever we are using information gain or entropy or Guinea entropy, our output feature was actually, uh, you know, it was a fixed number of categories. It can be a binary classification or multi-class classification. So with respect to decision tree regressor, we are going to see that how we are going to make a split for this. So let's take this particular example I have a data set which has experience gap and salary. Now based on this experience and gap, my model needs to predict what should be the salary. And this gap is nothing but career gap. Suppose if the person has career gap, obviously the salary may reduce a bit, you know. So this kind of model will try to create a decision tree model and will try to solve this regression problem with the help of decision tree regressor. And then we are going to understand how does decision tree regression work okay. So let's start with I'm just going to take this continuous feature and probably make two splits okay. First split I want to make sure that I keep threshold value this one. And the second split I probably keep a threshold value this one. So what I'll do is that two splits initially will start with less than or equal to two. So here I'm going to get one split other split okay. Wherever it is less than or equal to two. If this is true I'm going to basically say yes no it will become new here. Now with respect to this, how many different outputs I have over here you can see 40 K is my output that I'm getting over here right now. Similarly, if it is no then what? What is the remaining output that I'm getting. I'm just going to write it down okay. And this is what we did with respect to the classification also. Right. So here is my 42 K uh 5260 and 50 okay. Now this is the first split. And similarly let's say that uh uh, this is my first split. And uh, now I'm going to change my threshold to 2.5. So let's say now I'm getting a new threshold with respect to less than or equal to 2.5. And I have similar two kind of splits over here, which is this one and which is this one right now let's say in this split, if I'm saying less than or equal to 2.5, that basically means I have this two values. So I'm going to say 4042. And this will be 5260, 56. Now remember our output feature is continuous. Now obviously, uh, if I really wanted to compare between this two splits, which one to select, whether I should select this value in the feature or this value in the feature. We used to use information gain in classifier, but here we cannot use it because finally our output is a continuous value. So how should I basically check whether this split should be selected or this split should be selected. So we are going to use something called as variance reduction okay. Variance reduction. Specifically in case of regression problem statement. Now I'll talk about it. What is this variance reduction and all. And we'll try to discuss it with formula. So you don't have to worry about anything. I'll try to cover everything. So in variance reduction first of all we need to compute the variance okay. And I hope everybody knows the formula for variance. So I will just provide a definition. Uh summation of I is equal to one to n y minus y hat whole square. This y hat that I'm basically writing it is nothing but average right? Average. And uh, if you have seen this in linear regression and also we basically say this as mean squared error. Right. So we are basically going to compute the mean squared error for each and every node. And then we are going to combine them to actually uh find out the variance reduction. And then finally out of this two splits whichever has the maximum variance reduction, we are going to take that specific split. And that is how it is basically used. So in. Shot. If you're using Decision Tree Regressor, we use a instead of using entropy and information entropy or Guinea guinea impurity, we use something called a mean squared error. Okay. So this is how it basically happens now. Now what I am actually going to do. First of all I'm just going to go and compute y hat how to compute y hat. Just go over here and try to find out the average. So here if you take 4042 5060, 56 you will be getting 50 k okay. So I have already done the calculation. So I'm directly writing it over there. If you probably find a mistake you can let me know in the comment section. Okay so here it is. Now what I'm actually going to do. Let's go ahead and calculate the variance for root for all the child nodes also. Now in this particular case my root will have all my output values like 40 k 42 k and I will be having 52 k, 60 k and 56 k. Okay. Similarly over here also I'll be having all the values like 40 k, 42 k, 52 k, 60 k and 56 k. Okay, so I have all my values over here with respect to the variance okay. Now let's go ahead and calculate the variance of root okay. So first step we are going to calculate the variance of root. And then we'll calculate the variance of child node. Now in order to calculate the variance of root what I'm actually going to do. First of all in this particular root node how many values are there. So 12345 right. So five values are there. Now obviously what I'll write one by n as one by five. And then here you have summation of 1 to 5, one to all these particular values five. So y minus y hat. What does this basically mean. Y minus y hat basically means I'm going to basically consider each and every feature. Like for t minus y hat is nothing but 50. So that is what I'm actually going to do. So here let me write it down Y -40 whole square okay. This is my first one. And then we'll go and do it for 42 okay. So I will just go ahead and write 42 -50 whole square. Then my third number is basically 52 okay. So it is nothing but 52 -50 whole square. Plus after 52 I have 60 and 56. So 60 -50 whole square. And after 62. Oh so this is my 52 right. Okay. 5260 and 60. And then I have 5656 -50 whole square. So I'm just going to compute this. Now in order to compute it I can basically write one by five. I will be having 100 over here. This will be 64 over here. And um this will be four over here. I hope I am doing the calculation right. If I'm getting it wrong, you can let me know. Okay. So these are all the values that I am probably getting it. And if you probably use a calculator you will be able to see the specific output over there. And we'll be able to see the value. Uh, just go ahead and quickly calculate it. Use the calculator. But according to my calculation I have got 60.8. So here in short for variance of the root in this particular case is 60.8. In this case also it will be 60.8 because both are having this number of outputs. Now let's go and compute the variance for this child one okay. So child one for the first one. So here I'm just going to compute variance of child one that is c one. Then again I will basically have formula one by m summation of I is equal to one to n y minus y hat. This is nothing but y mean. In short, uh, here I'll be having one by how many numbers I have. Actually, with respect to n over here, only one number. Right. So only one number is basically here. So I'm going to just use one. And then this will be summation of I is equal to one to n. No need to write this again and again. So this will basically be 40 -50 whole square. Because my y meaning 50 so ten square is nothing but 100. So variance of child one over here is specifically 100 okay. So variance of child one is nothing but 100 okay. Now what about variance of child to variance of child two. So variance of child two. Let's go ahead and compute it. So here I will be having variance of child two and this will be nothing but one. By n summation of I is equal to one to n y minus y hat whole square. So how many is the end value over here I have 1234. Right. So 1234. So this will be one by four okay. And here you can start. First number is what. First number is 4242. -50. Whole square 52. -50 whole square. So I'm just going to write 42 -50 whole square plus 52 -50 whole square plus 60 -50 whole square plus 56. -

50 whole square. So you just go ahead and compute this. And obviously, uh, if I am just directly seeing my, uh, notes over here so that I don't want to spend much time with respect to quickly doing all those things. Right. So let's quickly see how much did I get. Um, and, uh, then we'll just go ahead and write the variance of, uh, child too. So if you do this particular calculation, it will be nothing but one by four here it will be 64 plus four plus 100 plus uh, 36. So in short, you are basically going to get somewhere around, um, let's see the calculation. Um. All 64 plus 468, 168 plus 36 two naught four divided by four, which is nothing but 51. Okay, so 51 I'm actually getting over here. You can do the calculation with respect to this. I just want to be accurate. So I have done this calculation. So this is what is my variance with respect to C one and variance with respect to c two. Now what I'm actually going to do is that once I compute all the variance with respect to root node with child, I'm going to apply my variance reduction formula. So my variance reduction formula will be something like this. It will be nothing but variance off route. Minus summation of weights of I, and then this will be variance of chai. Okay, now let me talk about this one also just in some time. So what is the variance of root? In this particular case it is nothing but 60.8. So here I have basically 60.8 minus W of I. What is w of I. Now you need to understand what is w of I. First of all we'll go with respect to the left node. Now in this particular case you can see that after the split how many are there in the left node only one output is basically there in the left node right 40 K. And before we had how much 12345. So if I try to find out the ratio that will be basically be giving me the weights. Now in this particular case, since I have just have one element on the left node, I'm just going to basically write the ratio as one by five. Okay. So that will be the ratio with respect to my variance root. So one by five. And then I will be multiplying it with my variance of child one on the left hand side. So it is nothing but 100. So here I'm just going to use it as 100. And then I'm going to do the summation. Similarly what will be the right hand side. You can see four elements are there. So I can definitely write four by five. So this is basically my four by five multiplied by what is the variance of child with respect to the this child. So it is nothing but 51 right. So this is my entire calculation. Now I can go ahead and just calculate it. And finally you'll be seeing 60.8. 60.8 -20. And probably I'll just see my notes again so that I can get the calculation quickly, because I don't want to do the calculation again and probably, uh, make a mistake on the same. Right. So let's see. Um. So, um, quickly, let me just do this. I'm just going to open my calculator. It is going to be 60.8 -20. Uh, ten, 10.2. So 40.2. So it is also going to be almost similar thing. Right. So let's go ahead and calculate it okay. I'm just going to use my calculator. It's good. So let's open the calculator quickly. So I am going to open the calculator. And this is nothing but four divided by five multiplied by 51 okay. So this is nothing but 40.8. So this is -40.8 which is nothing but 60.8. So I'm here I'm going to get zero. So in short the variance reduction with respect to the first split I'm just going to write it over here. The variance reduction with respect to the left split. Variance reduction with respect to the left split is zero okay. So this is for the left one. Left split that we made okay. Similarly let's go and compute for this particular split also. Now for this particular split I can do the same thing already I found out this. Let's find out the variance of child one. So child one over here I'm just going to use one by two because there are two nodes on the left. And then 40 -50 whole square plus 42 -50 whole square okay. So this is there. So one by two 100 plus eight is a 64. So here you will be able to see 164 divided by two which is nothing but 82 okay. So 82 uh, is what I am actually getting with respect to the first leaf node. So let me just compute it. Yes, it is perfectly fine. And similarly, uh, if I really want to calculate with respect to this node also what is the variance. So I will go ahead and calculate variance of child two okay. And this will be one by three. Since I have three nodes. And here 52 -50 is nothing but two square. So for this 60 -50 will be 100 because 1010 square and this will be 36 okay. So finally you can see that I'm getting 140 divided by three. So let me just open the calculator and let me write it down 140 divided by

three. So this is nothing but 46.66. Okay. Now, finally, if I go and compute the variance reduction. Okay, it is nothing but. Or let me just compute the variance reduction again. I'm just going to make a split over here so that I don't intermix multiple things. Okay. So let's go ahead and compute the variance. Uh variance reduction of the right node of this. Right. Right. Split. Right. So here I can basically write nothing. But uh what is the root node. It is nothing but 60.8 minus summation of how many elements are there. In the left hand side there are two elements 40 comma 42. So what I'm actually going to do two divided by five. It is going to be since I'm going to use that specific weight multiplied by what is the, uh, variance of this particular node. It is nothing but 82. And uh, finally you'll be able to see plus, uh, finally here is three nodes. So I'm going to basically use three by five multiplied by multiplied by 46.66 46.66. Now if I go ahead and compute it, it is nothing. But, uh, let's open the calculator quickly. So here you'll be able to see 82 multiplied by two divided by five. Okay. So I'm going to get 32.8 over here. So this is 32.8 plus. Uh, what I'm actually going to do, I'm going to basically say three multiplied by 46.66 divided by five. And that is nothing but 27.996. Okay, so if I add this plus 32.8 and if I try to subtract it with 60.8, you will be able to get something this much right. So I hope the calculation is right. And this will be .004. Sin 60.8 is bigger. So here I'm getting some variance reduction. And here my variance reduction was zero. So I can definitely conclude that variance reduction. With respect to the left. Left. Split. Left split, which was basically using that is less than variance reduction with respect to the right split, which we made it right. So obviously if I try to compare out of this two, which is the better split, I will definitely select this split okay. So this split will basically be selected for solving the decision tree regressor. And we basically use this particular step to find out with respect to as we go ahead again, we can do further splits and we can probably try to find it out. And similarly, you know, even I have continuous variables, I have to make different, different splits and then try to find out which is the best split going ahead. Right. So variance reduction is a very important property with respect to decision tree whichever variance reduction. Now when I say left split this was the left tree right left tree. This tree right this tree. Again let me mention which tree. So this was the tree right. Less than or equal to two. This was the tree that we are actually checking with. Less than or equal to two. And if I talk about the variance reduction which I got some value because over here this value that I got was zero right. This value I got was something like 00.004. Right. So obviously here you can see variance reduction on this side is greater. And this split that we made was less than or equal to 2.5. Further we can make any number of splits. So finally you'll be able to see that we will try to find out the variance reduction. And that is how we'll be able to select at which splits should be selected when compared to the decision tree classifier where we use information gate. So I hope you have understood this. Uh, and again, uh, you need to understand one more thing. Finally, let's say if this was my split, okay, let's say if this was my split, that was selected. Okay. And let's say this was my root node over here, what is there in the root node that I'm getting. Let's say this is my root node and I'm here I'm getting as 40 comma 42. So here I'm getting as 40 comma 42. And here I am probably getting what all values are 5260 and 56, 52, 60 and 56. Let's say I get my new test data. And this test data is basically less than two point less than or equal to 2.5. So obviously I'm going to go over here. So how I will be able to find out my final output. Let's consider this is my leaf node. Let's consider I'm saying okay after probably all the splits. So mine. Finally you'll be able to see the output will be the average of both these values okay. Now in this particular case my output will be 41. And this usually happens in regression. Any number that is greater than 2.5 will come over here. And here we will try to find out the average 52 plus 50 plus 60 plus 56 divided by three. So if I probably go and compute the average it is going to be 52. Plus 60. Plus 56 divided by three. Right. So once I get this, you'll be able to see that I'm getting 56 as my output. So if it comes over here the average output is basically the output of the decision tree regressor okay. So this is super important to understand. Apart from that we have understood what exactly is variance reduction and all. And how do I calculate variance reduction with respect to both the different types of splits. Right. So yes this was it. I'll see you all in the next video. Thank you.