In this video, we are going to discuss about a new machine learning algorithm, which is called as XGBoost. That is nothing but extreme gradient boosting machine learning algorithm. And with the help of this algorithm, we will be able to solve both classification and regression. Problem statement again. Uh, as usual, we'll take up a classification data set and we'll try to solve it. We'll try to see how the decision tree is basically constructed inside the XGBoost, and how sequentially we can actually construct it. What are the important parameters that we should look at it, and what is the function equation altogether in order to calculate the final output. So all these things will be discussing before that. Let's take a very simple example over here. So here is my entire data set. The data set has two input features. One is salary and credit score. And based on that whether you'll get a credit card approval or not okay. So this is the, uh, entire data set with respect to this. And here are some steps that I've actually written in the right hand side. And we will probably follow this entire steps in order to find out the XGBoost itself, or how to create a decision tree inside XGBoost. So to begin with, first step again, is it to basically construct a base model? So let me just go ahead and write over here. So first step is basically to create a base model. So let's say this is my base model. Now obviously with respect to different different base model right. Whenever we are creating a base model with respect to classification as my classification is a binary classification over here, what I'm actually going to do, I'm going to keep a probability of 0.5. So anytime whatever output this particular model will be giving, it will basically be giving the probability of 0.5 okay. So this is super, super important step. Uh, let's say that this base model is not biased towards anything okay. Not towards anything. So we are just going to take it as a probability of 0.5 later on. Uh, what output it is going to give. We'll discuss about it. Now let's go to the second step. Construct a decision tree with the root. Now we are going to construct our first decision tree, which will probably begin as after my base model. So the next that I'm actually going to construct is basically a decision tree over here. Now before I construct this specific decision tree, you really need to understand what data set I'm actually going to use. We are going to use salary credit over here and approval as my output feature. Now before that, whenever let's say after creating a base model, I know the probability that I'm actually going to get is 0.5. Okay. So with respect to any records over here, what is the residual that I'm actually get going to get. Because my residual one in this particular case let let me just write it down as r one. The residual one will be the difference between this y and my y hat which is my base model as 0.5. Right. Because my base model basically gives me the output as 0.5. So I'm just going to find out the difference between this and this. Now in this particular case, my difference over here will be minus point five. It will be point five. It will be point five. It will be minus point five. It will be point five. It will be point five and it will be point five. So I hope everybody is able to understand over here after constructing a base model, what I do is that I basically calculate my residual. Now before I go ahead and construct my next decision tree with this route, the input feature that I'm going to consider is salary and credit. And my output feature that I'm going to consider is R one. Okay. So let's go ahead and let's try to construct a decision tree. Now uh, this third and fourth important step that you will be able to see guys here I have basically written calculate similarity rate and calculate gain. This both step will actually help us to find out which feature should we select, whether salary is the best feature to select and split my decision tree, or whether credit is the best feature, uh, to basically select, uh, you know, to construct my decision tree. Let's say that, for instance, uh, I have anyhow found out that salary is my feature, that I really want to use it and use it for the splitting purpose later on will try to compare with with credit. Okay. So let's go ahead and let's consider that here. I'm actually going to take salary. So this will be my feature with respect to salary I'm going to use salary in order to construct my decision tree. Now with respect to salary you know that how many different types of output that I have with respect to salary, two different types, right? One is less than or equal to 50 K okay. And one is greater than 50 k. Now with less than or equal to 50 k. How many different types of values I have and with with greater than or equal to 50 k, how many different values that we have, or leaf nodes that I can say how many different values we have. Now let's go and see with respect to this particular data set with less than or equal to 50 k, I have -1.5 here. Then this is again less than or equal to 50 k. Then again I have 0.5 k over here. So and uh again one more less than or equal to 50 k I have 0.5. So I'll I'll keep on writing this. Minus .5.5.5 okay. So first of all I will go ahead and write it. Minus .5.5. And point five. And I can also see one last record is basically having less than or equal to 50 K. Again I have the value as point five. Okay. So here I'm basically going to write it as point five. Similarly greater than 50 K. I have three records. It is minus .5.5 and point five. So here I'm going to basically use minus .5.5 and point five okay. And finally here you can basically see that. And this is the next decision tree one that is getting constructed sequentially. And if I probably take up this particular root node I'm going to combine both these values over here. So it will be minus .5.5.5. .5.5. So this many point fives are there. And oh, last one I guess it is minus point five. So lets me compute it over here. So this is basically minus point five okay. So minus point five. And similarly over here I have minus .5.5 and point five. So perfect. Uh one computation because when I was trying to subtract the zero by minus point five I used, I had to get minus point five over here. Now this is what we have done with respect to the first split. Let's consider the salary is the best feature to do it. Okay I will now show you that. How can I compare two different splits with the help of similarity rate and calculate, uh, with the help of gain? So let's go ahead and try to calculate the gain uh sorry similarity rate for each and every node. So here you can see that I'm having this left side, uh split with respect to this. So obviously uh, let's go ahead and calculate the similarity rate for the left node. So here I'm going to basically write similarity rate. For the left child. So here I'm just going to write it as LC. And you know my formula. What it is. Summation of residual square divided by summation of probability one minus probability. This probability is nothing. But what is it? Whatever is the probability with respect to my base model, that is what I'm actually going to use. Now, if I probably do the summation of residual square for this node. So this is my left child. So how I'll do I will basically write 0.5 okay. And I will go ahead and write plus 0.5. And then I will probably go ahead and uh here you can basically see residual square. Okay. So uh, don't get confused with respect to this because this is super, super important. Uh, and I know many people will uh, basically do this confusion when I say residual all the summation of the residual, I have to probably take up. Okay. So -0.5 plus 0.5. So here also I have again another point five. And again minus point five whole square okay. Divided by uh if I say summation of probability multiplied by one minus probability, I have to probably take my base learner probability that is 0.5. And I have to make sure that how many number of residuals I have, that many number of time I have to do the summation. Okay. So here I'm going to first of all use probability. This will be 0.5 multiplied by one -0.5. So this will be my first for the first residual for this residual. Then again the same step I'll construct it for the next residual okay. Then again same thing I will do it for the next residual. Then again the same thing I will do it for next residual. So this is what I have to probably do for all calculating all this probabilities for all the residuals itself. Now you can go ahead and do the calculation forward. So here you can see this. And this will get deducted this and this will get deducted. So this entire value will be equal to zero. So if I probably calculate the similarity weight for this it will be equal to zero. Uh before that uh if you really want to do the calculation here you can do it. It will be 0.5 multiplied by .5.5 multiplied by .5.5 multiplied by .5.5 multiplied by point five. So this entire value will be equal to one okay. So zero by one. Anyhow it is going to be zero. So here I can definitely write that my similarity rate is equal to zero right. So this is the first step. Now similarly uh with respect to the right node also you can basically do it. So here I'm going to basically consider the similarity weight for the right node okay. So right child here I'm going to use right side. So again summation of residual square. So here I'll be writing 0.5 plus 0.5 plus 0.5 whole square. And here obviously you know that as many number of times if I write probability multiplied by one minus probability. Suppose if there are three. So here I'm definitely going to get it as 0.75. So this and this will get deducted. So it will be 0.25 by 0.75. So if you probably do this particular calculation it will be one by three. So one by three is here. So the similarity rate. With respect to this is. Similarity weight with respect to this is 0.33. Okay. Now similarly with respect to this root node, if you really want to calculate with respect to the root node, also we can do it. Now be a little bit smart. This and this will get deducted this and this will be get deducted this and this will get deducted. So I'll be having point to five divided by 12345677 basically means that basically means seven multiplied by 0.5 into point five. So it will be .25. So if I probably see this, it will be somewhere around 3514 1.75. So this is going to be 1.75 okay. Now just do the calculation. I've quickly done it. I know you'll be thinking how did we get this. Uh don't worry uh, just try to calculate it. You will be able to get it. Okay. So this that that is how we basically calculate, uh, uh, the similarity weight for the root node. Okay. Or root. And this is nothing but, uh, this value that I'm actually going to get, uh, you can go ahead and do the division with respect to this, but I'm directly going to write the value. So it is .14. So here the similarity weight I'm going to get it as .14 okay. Now whenever we have this kind of similarity weight with respect to the split, the next thing that I really need to find out is something called as gain. So in order to compute the gain it is very much simple. All we do is that we take the left child, we take the right child, and we probably subtract it with the root node. So left child obviously you know that how much the similarity weight is zero plus .33 minus uh minus .14. So if you probably do this here I'm going to get .21 okay. So this is the gain with respect to this split. Let's say I take up any feature and do the split over here. We will be comparing this gain. Whichever split will be having the highest gain will take that particular feature and will start splitting. Okay. So this is super super important super super easy. So let me rub this now what we have to do is that further going ahead. Now this is my split right now. Further I can also do more splits for this. Let's say I'm going to probably consider the further split from here. Right. And the next feature that I can consider is credit. Right. So if I probably consider credit, let me do one thing. Let me just make this equation go a little bit down so that you will be able to see this okay. So I'm just going to paste it. And let me put it more down so that it will be looking good though. Okay, just a second. Oops. Something happened. Okay. So this is fine. So here you can probably see all the similarity weight and all. So here I'm going to just rub it okay. So what? I did all the calculation. I moved it to the next page because after this step I really need to show what exactly the thing will happen. So here is my entire calculation with respect to the gain with respect to the left child, right child and with respect to the root. Okay, now let's consider we are going to take the next feature called as credit and make the further split. Now again, if I probably take credit over here okay. If I probably take credit a credit usually has two types. Okay. One side I can basically take bad and one side I can basically take good and normal. The another kind of split that can happen from the same node is that if I take credit over here, okay, one side I can have bad and good and one side I can have normal. So what is that is what I was trying to talk about. You know out of this two split, which split should I take. Right. That is the main thing. So that is the reason what we do. We again go ahead and calculate our. We go, we go ahead and calculate our similarity rate. And we go go ahead and calculate our our gain, you know, in short. So here you can see when the credit is bad, when the credit is bad. Here you can see that what value it is going to come. So first of all you can see the when the salary is less than or equal to 50 k. So the next step here you'll be able to see credit uh if it is less than or equal to 50 k. And the credit is bad here I'm going to get one -0.5. So here I obviously one -0.5 I'm going to get over here out of all this four records. Right. And uh the other scenario, whenever my salary is greater than or equal to 50 less than or equal to 50 K, I don't have any other records that has bad in this side. Okay. Because everything I have either good or normal. Right. So remaining all these three records will be coming over here. So .5.5.5 right. Point five I am -0.5. So I have all the records over here okay. So let me again write it down minus point five. Now what I will do is that again for this split I will go ahead and calculate the similarity rate. So go let's go ahead and calculate the similarity rate R for this left child. So obviously here you know it will be minus point five whole square. So it is obviously going to be 1.25.25 and divided by you know that. What what do we divide by. We basically divide by probability of more summation of probability multiplied by one minus probability. Right. So this will in turn be point five. So here obviously I'm going to get it as one. Because the reason why we get it as one over here. Because you can clearly understand that I just have one simple root node right. So here this is my similarity rate is equal to one because I just have one leaf node okay. So similarity rate uh. Similarity weight with respect to right node right child. Here also you can basically calculate this and this will get deducted. So again this will be .25 divided by .75. So if I probably go ahead and calculate this this will also be .33 okay. So here I'm going to get the similarity weight off .33 right now. Similarly with respect to credit I already know what is my similarity weight on top of it it is zero. So in short, uh, you will be seeing that finally I will be getting a gain which can again be calculated by one minus point. Sorry, one plus 0.33 minus. Uh, here you can see it is zero right. So overall here what I'm actually getting is 1.33 okay. 1.33. Now this is with respect to this split. Let's say again we calculate with respect to this split. And let's say if you are getting less than 1.33 then we'll try to select this. But if you go ahead and do uh or calculate the similarity weight and the gain with respect to this split, also we'll be getting 1.33. So out of this two, let's say that we have selected this now with respect to any new data set. How will the prediction basically happen? We are going to see and here I have actually constructed the entire decision tree, uh, of this by using this specific data set. Now let's say I pass this specific record. Now when I pass this specific record, the first thing is that it passes through this particular base model, right. So when it passes through this particular base model, it gives the value with respect to probability. But if we are getting the value with respect to probability, one thing that we really need to do is that we really need to, uh, calculate the log of odds. Okay. We basically say log of odds. We cannot directly use the probability because in logistic logistic regression also you saw that right? Whenever we used to get a probability, we used to pass it through a log of odds function. Right. Which is which was my cost function. So in short my first formula whenever for a new test data. It should basically. Pass through, log off odds and log off odds is basically given by a formula P log of p. Let me write it down much more clearly. So here you have log of p divided by one minus p now. And uh you know this is basically the log of odds formula. So in the first instance let's say if my first record is basically passing over here, uh, through the base model, okay. If I probably calculate, okay, what will be this probability if I place it over here to find out the log of odds. So log of odds. If I try to calculate over here it will be log of point five divided by point five. So log of one is nothing but zero. So whenever it passes through this base model as an output I will be getting my output as zero. Okay this is super super important okay. Now let's say if my first record passes through this, let's say if this is my data that is getting passed okay, to my XGBoost, then obviously the first value that I'm going to get from my base model is zero. So I will just write it down over here as final output. Let's say the predicted output that we are going to see when I give my new test data. In this particular case, I'm giving my first record. My first record is going to get zero as the output okay. And top of it, what I'm actually going to do. I'm also going to apply a sigmoid activation function okay. So first base model it is basically going to get zero okay. So here you can see like uh or let me do let me just create this once again so that you, you will not have any confusion.

So this is my base model. And then it will probably pass it through my decision tree one okay. Which we have actually constructed. Okay. And this decision tree will be up till some new depth okay. So first thing my base model I have already shown you that my output over here is zero. Okay. So for any new data. New data. Let's say in this particular case, I'm passing this first data point. So less than or equal to 50 K and credit is bad. It passes through the base model here. The output that I'm going to get is zero. So here I will be getting zero. And on top of that I'll apply a sigmoid activation function. Let's say that I'm applying an activation function over here. And this is basically my sigmoid activation function. Okay. First model that we pass is basically zero. Now when it goes to this decision tree that is this decision tree, you know, what is the salary. Salary less than or equal to 50 okay. So it is going to go over here and over here it is saying credit is bad. Then again it is going to go from here to here. And then it is going to go from here to here. So finally you can see that I'm getting minus point five. But here you can see that my similarity rate that I got initially was one. Right. So when I apply this in short that will basically be my output. But before that we will be adding a learning rate and my similarity rate will be 0.13 will be one because that is what I got my similarity similarity rate over here. Right. The similarity rate in this particular case was one. So this is basically my similarity rate. Right. So after this, what happens is that this sigmoid activation function I'm just going to expand. Let's consider my alpha is 0.1. Learning rate is usually between 0 to 1. In this particular case I'm going to use 0.1. So here I'm actually going to get 0.1 multiplied by one. And here finally you'll be able to see that I'm going to expand this. Uh, in short this is going to be 0.1. So this sigmoid activation function is given by one plus e to the power of minus whatever. I'm actually getting as 0.1 over here. So if I probably apply this and if I try to calculate it how much I will be getting. So here, uh, let me open a browser and let me just do it in front of you. So I'm just going to search for over here one divided by. One plus e to the power of minus one or point one. Always remember with the help of sigmoid activation function, we are going to get a good probability because sigmoid activation function always gives us an output between 0 to 1 that we have already learned. So here you can see it is minus .52. Right. So here you can see alpha one multiplied by this value. And finally you will be able to see when I apply this sigmoid activation function how much is the I'm getting I'm getting 0.52. So whatever .52 I get over here this will basically be my. Predicted output. This will be my predicted new predicted output y hat. So this will be 0.52. Okay. And similarly let's say that I have computed for all the other records. Let's consider for the next record. For the next record it is like less than or equal to 50 k green and one okay. So green. Sorry. Good. Right. So less than or 50 K. Again uh first of all it will pass to the base model. So second record. What will be the output. Okay. First thing is that obviously I'm going to get I'm just going to apply a sigmoid. The first value that I'm actually going to get is zero because it passes through the base model. And then I will be using my learning rate. That is point one. And here we really need to find out what will be the output with respect to Decision Tree one. All right. And this is basically my decision tree one right. So here you will be able to see when I pass less than or equal to 50 K it passes over here. The credit is good I guess in this particular case the credit is good. This is my second record. Everybody see this? This is my second record. The credit is good. So finally I go on to the right hand side. And here you can see that I've got three records over here. But my similarity rate is 0.33. So what I'm actually going to do multiplied alpha multiplied by 0.33. So here again you will be seeing sigmoid zero plus point one multiplied by .33. Right. Then what I am actually going to get it. Will nothing be one plus e to the power of minus whatever the multiplication will be coming with respect to this .33 multiplied by point one. Okay, so here you'll be able to see I'll open the calculator and here it will be 0.33 multiplied by 0.1. So it will be point zero. I think it is ..033. Okay, so this is nothing but .033. Okay. Now again, if I go ahead and calculate with the help of calculator point. 2033. I'm going to get the output as .508 okay. So anytime I will basically be getting between 0 to 1. So this will be .508 which will be my final output for the second record with respect to the predicted point. So it will be .58 right. So like this I will be getting a lot of values again I will go ahead and compute my R2. So whatever R2 I'm actually going to get. Now in this particular case my R2 will be zero -0.52. So here I'll be getting 0.58. And similarly here I will be getting 0.42. So like this all the values will be there. And then again we are going to construct a next decision tree. Considering this input features and R2 as my output feature. Then again we'll go ahead and calculate similarity rate and calculate the gain. And this process will be keep on going unless and until we get all this kind of trees which will start with a base learner. And then it will multiply with alpha multiplied by another decision tree which will be over here. And alpha is basically my learning rate to avoid overfitting. Then again there will be an alpha multiplied by another decision tree output okay. And this will continue till the end. And so this in turn is nothing. But this is my XGBoost classifier. Okay. This is my super XGBoost classifier. Now this was the entire working about the XGBoost classifier. I know there were a lot of little bit of ups and downs because I have written up and down. But again, if you just go through this, go through the video again in order to understand two important things. First of all, with respect to the base model, I have to calculate log of odds and with respect to finding the final output, I have to apply and sigmoid activation function, which is also called as logistic function. This is also called as logistic function okay. Logistic function. Okay. If there are multiclass classification, then the logistic function will be converted into a softmax uh, activation function. Okay. And then whatever I'm actually using I'll keep on adding it. So finally you'll be seeing your output. Final output with respect to predicted points will be sigmoid function. Uh with respect to your base learners. Base based learner probability. Okay. The base learner plus alpha one. Your decision tree one output plus alpha two. The decision tree two output plus alpha three decision tree three output and so on okay. Now, one super important point that I wish I did not say. Whenever we compute a similarity rate, there will be also one more parameter which is called as lambda. And this lambda is basically your hyper parameter. Okay, we basically do a lot of cross validation in order to select the specific lambda. And one more thing is that. There is always a thinking that till when we should basically keep on splitting this. Should we keep some restriction with respect to the similarity with right? So in order to find out that specific thing, what we do is that we set up a cover value. The cover value is given by this equation that is probability of one minus probability. So whenever any weight is less than this particular output that we get when we apply this, you know, whatever output that we are getting, what we do is that we slice down that entire tree. Suppose in this particular case, let's say we are going to get 0.25. And now suppose if my similarity weight is basically less than 0.25. So we stop the we stop the splitting at that point of time. So this is specifically called as your cover value. Okay, so this is nothing, but this is your cover value. We also say this as cover value. So similarly in case of regression or in case of classification. Also we basically say this as cover value. Now in classification we'll be seeing the formula will be changing with respect to similarity weight and gain only. That is the changes. But almost the process will be entirely same thing. Suppose if you are doing the regression problem statement at that point of time, you'll be seeing that there will be some changes with respect to similarity, weight and gain. Okay, other than that, I think everything is fine and you just have to follow this particular step in order to construct the XGBoost machine learning algorithm. So I hope you like this particular video. Uh, yes. Please make sure that you keep on revising things. Then only you will be able to understand multiple things. And this is how you basically create a sequential decision tree in XGBoost. So yes, I'll see you all in the next video. In the next video we'll discuss about XGBoost regression. Thank you.

In this video we are going to discuss about XGBoost regression machine learning algorithm. Again, like how we discussed about XGBoost classification machine learning algorithm. We took a simple data set and then we saw that how the sequential decision tree will get created. Similarly we are going to do that follow that same step in boost regressor. And we'll try to compare what is the exact difference between both of them okay. So to begin with here I am taking a simple data set. This is a regression data set altogether. So my input features are experience and gap. Over here you can see based on experience and career gap I need to predict what is the salary. So here salary is my output or dependent feature. And so this is a regression problem. So in short with the help of XGBoost regressor we can also solve a regression problem statement okay. Now if you remember about bush classifier again guys remember in XGBoost classifier to select the feature and probably construct the decision tree, we really need, we we use something called as similarity weight. Right. So if I talk about similarity weight this was the formula with respect to similarity. So the similarity weight was nothing but summation of residual square divided by summation of probability one minus probability. Right. So this was the formula that we specifically used for classification problem. And after that we also used a formula for gain. Now in this when we are solving regression problem again this formula is going to change slightly a bit. Uh you know the denominator will be changing since this is a regression problem statement altogether. So let's start with the first step. Then again I'm just going to note down all the steps that you have to probably follow in order to create an XGBoost regressor. That basically means sequential decision trees. First of all, we basically create a base model. So let's say we create a base model and this base model. Since this is a regression problem statement, the base model will be giving me the output of what is the average that we are going to get over here. Right. So if I probably calculate the average it is 40 plus 42 plus 52 plus 60 plus 62 divided by five. We approximately get somewhere around 51 K okay. It is somewhere around 51.2. But I'm just going to consider as 51 K. So whenever any data, any data that passes, if I probably take experience and gap and pass through this base model, my output that I'm actually going to get is something like 52 K. Okay. So we have understood our base model is obviously we are getting somewhere around 50. Sorry, 51 K, not 52. I am extremely sorry about that. So 51 K because this this is the average that we calculated out of all the salary. And this is a base model which is not at all biased towards any values. Okay. Whatever is the average over here with respect to the output that we are going to get. Now we understood my base model output over here is specifically 51 K. Now the first thing that we calculate again over here I will write it as y hat. Now when we are calculating with respect to the y hat one thing you have to make sure that here I am having 51 k. So every time when I write with respect to whenever I pass my any data over here, like let's say let's pass this experience as two and gap as yes, it is going to give me 51 K as my output. So in the first case my y hat will be 51 k. Okay. Now what we are going to do in the second step is that we are going to compute our residuals okay. So like how we did it in the XGBoost classifier. So the residual I'll write out what all I am going to do is that find out the difference between 40 and 51. So obviously I'm going to get -11 k. And here I am probably going to get minus nine k. And similarly over here also I'm going to get 52 -50 11K. And uh I'll not write k over here because again it will just look like okay I'm providing some unit. So here residual we're just trying to find out the error. So this will be minus nine. This will be one. This will be nine and this will probably be 11. So these are my residuals. Now in the second step what we did is that we computed the residuals okay. Residual computation by using this predicted off the predicted output of the base model. Now coming to the third step which is super super important is that in the third step we are going to construct. Construct a decision tree using because after this we really need to create one sequential decision tree, right? Using x of I as an inputs and r one as my output. So for with respect to my decision tree what I'm going to use x of I with respect to all my inputs x of I is nothing but my experience and gap and r one is specifically my output feature for the first decision tree. Now let's go ahead and construct it. So first of all, let's say that I'm going to pick up experience as the feature and start creating the decision tree. So in the first instance let's say that I am using experience. In this experience okay. Let's say this is my experience now with respect to experience. Since this is a continuous feature, you know that how we have to split in a decision tree, we can take up every record and we can start splitting it. Like we'll just take up two. Okay. And here I will say that okay, if my value is less than or equal to two. So two splits I'll make binary less than or equal to two and greater than two. So with respect to this particular split, like I'm just taking the first record, I'm keeping two as threshold and I'm splitting it. And similarly if I want to try you can also try with other records also. Okay. So let's say I have taken this. So obviously if it is less than or equal to two, I'm going to get a residual as -11 on the left hand side. So here I'm going to use -11 okay. So this will basically be my leaf node in this case okay. -11. Now greater than or equal to two I will be having four records minus nine one nine and 11. So here I'm basically going to use. Minus nine. One. Nine and 11. Okay, so these are my records over here that I've actually got it with respect to experience. Now as soon as I come over here now we have to calculate the similarity rate. Now if you remember in XGBoost classifier also we computed the similarity. But in the formula was very simple. For classifier it is nothing but red pseudo square divided by uh, the thing that we are going to use over here is nothing but summation of probability one minus probability plus lambda. Okay, now here we are going to specifically change the formula a bit. And over here the formula will not be probability one minus probability multiplied by one minus probability. Since this is a regression problem here, the thing that we are going to consider is we are going to use number of residuals okay. So here we are going to use number of residuals. Residuals basically means how many residuals are there with respect to this node with respect to this node okay. So let's go ahead and compute the similarity rate I'm going to write as w for the left child first of all. So for the left child here it is nothing but residual squared. So 11 squared is nothing but 121 divided by one plus. Now lambda value is a specific hyper parameter okay. Lambda value is a specific hyper parameter. And as you know that if the lambda value increases, your similarity rate decreases. Okay. So this is basically a kind of cross validation we have to do in order to select the lambda value. For this particular use case. Let's consider lambda is equal to one. So I'm just going to write lambda is equal to one plus one. So here in short I will be getting 121 divided by two. So in short I'm getting so six to the 12. And this will be 5.5 right. So 65.5 I'm getting my similarity rate over here okay with respect to this okay. Now similarly uh let's go ahead and compute the similarity rate with respect to the right child. Similarity weight with respect to the right child. And now here you can see that the right child has lot of values like they have four different values. So I'm just going to use this minus nine plus one plus nine plus 11 whole square divided by how many number of residuals are there. Total number of residuals that I can find is 123, four. So it will be four plus one which is nothing but five okay. So this and this will get cancelled. So it is nothing but 144 divided by five. Let me just open up the calculator so that you will be able to see it. So 144 by five is nothing but 8.5, not 8.5. I guess it is 28.5. Yeah, 28.5 is my similarity weight off this side? Okay. Now similarly you go ahead and calculate the similarity rate for this root node. Because in the root node I will be having all this records. Right. So we if we combine both of them. So here I'll be having -11 minus nine 919 comma 11. So when we consider all these things here, if I apply the same formula I'll be getting similarity rate as. 1.16 okay. Now, once we calculate the similarity rate, which was my fourth step, the fifth step is basically that we calculate something called as gain. Okay. Again. Now, in order to calculate gain, I'm going to just write gain is equal to similarity weight of the left side. That is 65.5 of the right side I have to add. And I have to subtract this with .16 I guess. Right. Or it is .16. So when I do this calculation here, you will be able to see that I will be getting somewhere around 98 point. Three four. Right. So here I'm going to get 98.34. Now you may be thinking Chris why did we do this. Because see, if I do one more split and let's say I'm going to change the threshold. Now initially my threshold is experience two. Now if I make my experience with 2.5, I can again make another split. Right? So let's say if my experience split is basically based on less than or equal to 2.5 and this is greater than 2.5, then again my split will change here. Now I'm going to get -11 comma nine. And here I'm probably going to get one comma nine comma 11. Right. So I can what I can do is that our main aim is that within the decision tree, whether should I use this split or this split for my decision tree, one that you really need to think about, right. Which split should I use? Okay, so obviously if I get if I probably calculate the similarity rate here, here and here, and let's say if I get the gain as 100 or 108 okay. And obviously for this it will be greater than this. You can go ahead and compute it. Uh, if I probably do show you all the calculation, uh, the gain that I will be probably getting for this entire split is somewhere around somewhere around. So the gain will be somewhere around 143 .42 for this. Okay. Now here you can see 143 .42 is greater than 98.34. So out of these two split I am going to definitely use this split. Similarly I can compare with other other thresholds over here. Okay. So that is how we basically select our first decision tree. And we can do further splitting with the help of gap also. So probably I can go ahead and do the further splitting. With respect to gap, let's say that I have selected this particular slip because the gain is high. So how will my base learner look like now my base learner is over here. This is giving me the output. What output this is giving me. We actually saw it is. It is nothing but 51. Okay then my next decision tree will be having this split wherein let's say my experience is less than or equal to 2.5 and greater than 2.5. Right. And let's say over here I have uh, what all values I have, I have -11 comma nine -11 comma nine. And this will probably have my other values, which is nothing but one comma nine comma 11 okay. Now what I can do, I can further do splitting of this, let's say, uh, for greater than or equal to 2.5 if my gap is yes or no. So if my gap is yes or no, I can probably make, uh, another split with respect to this. And here I will be having yes or no. If my gap is yes, uh, then I will probably be getting 11 over here. If my gap is no, I will be getting one comma nine. Okay, so let's say that I have done this kind of further splitting based on the data that I actually had. And I have made this entire split like this. Okay. Now, super important thing that you really need to note down over here. Okay. Uh, is that how do I find out my final output? Because this is just my decision. Tree one. Right. This is just my decision tree one. This is my base learner. Right. And you know that with respect to XGBoost classifier, also what we do is boost. Sorry regression. First of all, let me just recall XGBoost classifier. In a boost classifier, what we do, we take the log probability of a log of odds of the base learner. Okay. Plus alpha one, which is my learning rate. And then I take decision tree one output, then alpha two, decision tree two output. Similarly like this up to alpha and decision tree and output similarly with respect to XGBoost classifier. Sorry. XGBoost regressor. Let's say I pass this test record.

This record, let's say I am just passing this record. So let's see what will be the output in my case. So when I pass this record. So obviously experience is greater than 2.5. So if experience is greater than 2.5 so I will take this route. And over here in my this specific record, my gap is no. So if my gap is no, I'm going to probably take this record. Okay. This part now here you can see that my output has one comma nine. Now in order to find out the output of this, I just have to find out the average of this two numbers. So it will be nothing but five. So my regressor first of all passes through the base learner the the test record. New test record passes to the base learner. So my base learner has the value as 51 k plus alpha one here. Also alpha is basically be my learning rate. And let's consider learning rate over here as point one. You can also consider .2.3. This is again a kind of hyper parameter okay. So here in this particular case I'm taking alpha as point one. Once we take this alpha all the other alpha value will also be having this. And the decision tree one output will basically be five because I got the output over here as five okay. So here if I multiply by five. So here you will be able to see I'll be getting 51 plus point five. Now when I say 51 plus point five here you'll be able to see I'm going to get 51.5 okay. Now with respect to this this will be the output okay. Output for the third record that we are given. And over here what was that? Uh, you could see that my experience was three years and my gap was no. So for this, the output that you will be able to see I've got 51.5. So I'm just going to write down the Y predicted output, the new predicted output after Decision Tree one. I'll write it as y hat and my new predicted output. So here you can see for this, uh how much I'm getting I'm getting basically. 51.5. Right? So finally I can see that I'm getting 51.5 over here. Okay. So guys we have got the new predicted output for this record like 51.5. Now what we are going to do is that we are going to calculate the predicted value for all these records. Okay. So let's consider I want to pass this record where the experience is two and gap is yes when the experience is two. So here obviously you can see it will first of all pass through the base learner. So what I can basically write over here first of all when it passes through the base learner 51 will get added. Plus my learning rate is point one. Now I really need to find out what is the output of the decision tree one okay, now if I go over here. So since my experience is less than two. So here you can see it is coming to this node. But here in this node I have two values -11 comma minus one minus nine. So here the output will basically be the average of this two node which is nothing. But it will be minus ten okay. So here I'm going to basically multiply by minus ten. In short, what we are going to do, we are going to subtract with point one, which is nothing but 49.9. So my output over here will basically be 49.9. And this record also will go on the left hand side. So this will also be 49.9. What about this record. That is four no 60 K. So here you know that what will happen. The first record first base learner will give 51 okay. And then I have a learning rate point one. And my next output of the decision tree one will be when I go over here four is greater than 2.5. That is fine. We are going over here. And with respect to this, what is the record that I have over here with respect to gap? No. So the No will basically give me one comma nine, which is nothing but one plus nine divided by two, which is nothing but five. So the output will be five in this case. So here you can see point one multiplied by point. Uh. So multiplied by five, which in turn will be equal to this value itself to 1.5. So here also I'm going to get 51.5 here. Also I am okay here I'll be getting a different value because here we will again multiply 51 plus point one multiplied by. Now the decision tree output with respect to gap as yes will give me the output as 11. Okay. So here the output is 11 okay because I just have one node. So here you can basically write point one multiplied by 11. Okay. And this is nothing but 51 plus 1.1, which is nothing but 52.1. So here is my 52.1. Now, after I calculate my new predicted output, because I have just created one based learner and one tree. Okay, now my work is that I have to create another tree and to create this another tree, I have to give my dependent feature and new residual R2. Now R2 will basically get computed by the difference of this and this. Right. So if I probably subtract this and this it will be -9.9 here. Probably it will be -7.9 here it will be 0.5. Here it will be 51 8.5. I guess 8.5 or 9.5 8.5 will come over here, and here you will be able to see I'm going to get somewhere around uh, 52 to 62. This is nothing but 9.9. Okay. So I hope I have done the calculation correctly. If there is a slight mistake, don't worry about it. You can do it now. In my next decision tree that I'll be constructing here, I'll be taking my independent feature experience and gap and my output feature R2, and I'll be constructing this decision tree. And this will keep on going unless and until we have multiple decision tree that is created like this. Right. So this is what is the entire process of implementing the XGBoost regressor. In short in XGBoost regressor also will be applying these things. Now there is if I consider my similarity rate formula. And inside the similarity. Wait. We have found out. What is the formula? Residual square. Divided by number of. Number of residuals. Plus lambda. I hope this is the formula with respect to your similarity weighting regression. Okay, this is super important to cover up. Okay. And with respect to classification, you already know what is the similarity weight. And we obviously have to calculate similarity weight and gain in order to find out which is the best one. So this is nothing but residual square divided by probability. Summation of probability one minus probability plus lambda lambda. Understand is a kind of hyperparameter. As we keep on increasing the similarity, weight keeps on decreasing. Okay, now this is super, super important to understand all these things. And uh, you know, uh, they are also some amount of hyper parameters which will be using and that I will definitely show in practical. But here I think you have got an idea about XGBoost regressor and its working. And what is the importance of Lambda and how the decision tree is basically getting constructed. But in short, these all are sequential decision trees that are getting constructed. First we go with base learners and then we go keep on continuing to create different, different decision trees. So I hope you understood this video and yes, I will see you all in the next video with some practical sessions. Thank you.