In this video, we are going to discuss about the AdaBoost machine learning algorithm. In our previous video, we have already discussed about two ensemble techniques. One is called as bagging and boosting. In bagging we have covered about. We have covered the machine learning algorithm which is called as Random Forest Classifier and Random Forest Regressor. So now we are going to move towards boosting algorithms. Now in boosting we are going to start with AdaBoost. Then we are going to see gradient boosting. And then we are going to see XGBoost. Remember one thing guys, with both this bagging and boosting, you'll be able to solve both classification and regression problem statement. Now to start with, one more important point is that all this algorithm uses decision trees as the base learner or weak learner. Whenever we are talking about boosting, we basically see something called as weak learners over here and here in the bagging technique, we use something called as base learners. And then all these things we will be using decision tree our as our base machine learning algorithm or model. Okay. Now let's go ahead and let's quickly discuss some of the important or revise some of the important points. Now in Decision Tree guys, we know that if we are trying to create a decision tree to its complete depth, right? So if we are trying to create a decision tree, let's say to its complete depth. Now in this scenario, what will happen is that in this scenario you will be seeing that okay, fine. Let's say if I'm creating this particular decision tree to its complete depth, in this scenario, it usually leads to something called as overfitting. Now, whenever we say overfitting, that basically means our training accuracy is very high, right? But our test accuracy is very less right. So during this particular scenario we have a condition. If our training accuracy is very, very high, we basically say it has low bias. And if my training test accuracy is very less, then we basically say this as high variance. Now. As we went towards random forest right in Random forest. What we have we have multiple decision tree right in random forest we have multiple decision trees. We use multiple base learners like decision tree. So the question that people may ask you is that our interviewer may ask you is that if decision tree is there, why do you use Random forest? And I have already explained you in Random forest since we are using multiple base learners. Right? Suppose I have a data set. I will be providing a sample of data set to multiple decision trees, and those are my base learners here. What we do is that we try to convert this low bias and high variance to something called as low bias and low variance, right? How do we convert this into low variance. Low way low this high variance into low variance. Since we are using multiple base learners and we are giving sample of data set for training of that specific model. Now this random forest technique that we were using is something called as bagging technique. Right. And in bagging technique we use base learners. Now if I go to boosting now in boosting what happens is that. Boosting. What happens is that we try to create from this particular data set. Let's say this is my data set. We create a weak learner. Right. Suppose this is my, uh, decision tree. We create some something called as weak learners. Right. So this weak learner or let me just not draw this decision tree right now, let's say that I am going to use multiple models. Let's say this is my decision tree one. Then I will be using all these models sequentially. I'll be connecting this particular model sequentially. And what we do in boosting technique is that whatever records have been, uh, you know, wrongly answered over here or wrongly predicted, we send that particular records to this particular decision tree. Similarly, from decision tree to whatever records we have wrongly predicted in decision tree to that we send it to the next decision tree. Like this. All the models are sequentially connected, okay. Sequentially connected. Sequentially connected. And here for this models we see something called as weak learner. And if I combine all these weak learners sequentially. Final thing that we are going to get is something called a strong learner. Okay. So usually in the case of boosting technique, what does a weak learner basically mean? What does a weak learner basically mean? Weak learner basically means that this models have not been trained properly

with the training data set. That basically means this weak learner haven't learned. Much from the training data set. So guys, now you already know that in Random Forest. Suppose if you are going to consider Random Forest. And in in in case of classification algorithm, we use something called as majority voting classifier. Right. Majority voting classifier. In the case of classification problem, that basically means after training our model, whatever will be the output from all the models, we will basically do the majority voting and we'll get it. Similarly, in the case of regression problem, we basically find out the average of all outputs right now with the help with the help of AdaBoost. AdaBoost will not work like this. In AdaBoost we assign some weights. We assign weights to this weak learner. Weak to the weak learner. This is the most important thing in AdaBoost. Whenever you use boosting technique that is sequentially connected. Technique, we cannot go with maximum or majority voting classifier. Instead, we focus more on assigning higher and lower weights. Everything will make sense as we go ahead, but just let me just tell you all the important points with respect to AdaBoost, okay? So if I really want to create a AdaBoost function, the function will look something like this f is equal to. Alpha one multiplied by model one plus alpha two multiplied by model two plus alpha three multiplied by model three like this up to alpha and multiplied by model N. Here M12MN are my. Decision. Trees. Stumps. I'll talk about what exactly? Stumps. Okay. Just let's consider that m one to m m n are my decision tree stumps. Now this Alpha one. Alpha two. If I probably talk about this Alpha one, Alpha two, Alpha three up to alpha n. These are my weights. These are my weights. Which weights I'm actually talking about over here. These are the weights that are assigned to my weak learners, because these are also my weak learners. This is also my weak learner. This is also my weak learner. This is also my weak learner. And based on this particular function, we try to solve both classification and regression problem. Classification and regression problem. Now, one very important thing that probably you have discussed, right. See, in Random Forest, I already have low bias and low variance. Now suppose let's go ahead and discuss about AdaBoost. What is this decision tree stump? Decision tree stump basically means here we are going to create a decision tree whose depth will be just one level okay. So suppose if this is my decision tree stump one, this will be my decision tree stump two and we are going to connect this decision tree sequentially. Okay. This decision tree sequentially like this. Any number of decision tree we will be having. And this decision tree is basically called as a stump stump. What the stump basically means that the depth is just one one level. Just one level depth. Okay. Nothing more than that. So that is the reason obviously if you see that if my decision tree basically have just one depth, then this can definitely be considered as a weak learner. Whenever I have a weak learner this basically leads to underfitting, right? Underfitting. Now whenever I have underfitting, what does this basically mean? This basically means that, uh, my training accuracy. Is less. And there may be a scenario that my test accuracy may be more or less. Let's consider that it is more right, but your training accuracy is very less okay. If I probably say if my training accuracy is 40% and let's say if my test accuracy is 45%. Now in this scenario, the test accuracy is more than your training accuracy, right? But here your training accuracy is very, very less. If this was somewhere around 90 and 95, I would have been very happy, right? But over here, since our weak learner is just for one death, we can definitely say this is actually underfitting the conditions, right? So usually initially in this particular decision tree stump, what we have, we have something called as high bias. Because whenever we talk about bias we are basically talking about a training data. And then we have something called as low variance. Right now when we combine, when we combine multiple decision tree stump like this sequentially, when this all are basically combined together, then this will in short lead to something called as low bias and high variance. Right? So in short, if I'm connecting all this sequentially together, one after the other, this entire combination will basically be leading you to

low bias and high variance initially with the help of decision tree stump. We were getting high bias and low variance. Right? Now let's go ahead. Because here I just told you what exactly AdaBoost is basically happening in AdaBoost. What exactly is happening? We are just creating weak learners. Why do we say this trees or this stumps as or I can also say decision stumps right? This stumps as weak learner because it just has a depth of one. But when I combine all together, if it has just a depth of one and this if independently, if we are trying to do some prediction, it will lead to overfitting, right? Overfitting basically means high bias and low variance. But if I try to combine all this decision tree stump along with weights, okay, because here we will be. As I said, we are going to assign some weights right? When we combine this we are going to get low bias and high variance. Okay. Now in the upcoming video we are going to see how we are going to solve this AdaBoost. Are we going to understand the maths intuition? Okay. How does AdaBoost actually work? Okay. And when we understand once how the AdaBoost will actually work, you'll understand how the boosting technique actually comes up in solving any classification and regression problem. So yes, I will see you all in the next video. Thank you.

In our previous video, we have got a brief understanding about AdaBoost and what is our main aim in AdaBoost. Will try to combine multiple weak learners and then will try to create a strong learner and the weak learner. In case of AdaBoost is basically called as decision tree stump. Okay, decision tree stump and this decision tree stump basically has a depth of only one when we are constructing the decision tree. Now let's consider this is my data set and we will try to create this entire AdaBoost. I will try to understand the entire math in depth, intuition, and will also make sure that will create the decision tree stump over here. And I'll try to show you the example how the entire AdaBoost tree is basically constructed. Okay, now first step always in AdaBoost like this is my this is my entire data set that is given. I have features like salary, credit approval. Okay, so this credit is nothing but credit score. It says that if less than if my salary is less than or equal to 50 okay. And if my credit score is bad, I'm not going to get a credit card approval. Okay. Similarly, if my salary is less than or equal to 50, okay. If my credit score is good, I'm going to get the credit card approval. So like this I have all the data set here and basically means normal. The credit score is normal. Okay I just created my own feature. Now let's go ahead and let's try to understand the first step. In the first step we create decision tree stump okay. Decision tree stump. Obviously you know how to create decision tree with the help of entropy and information gain. But here we are focusing on creating the decision tree stump. That is just with respect to one record, one max depth. Okay. Now first of all, let me consider one feature salary. So this will be my feature salary okay. Here I'm just going to write the condition salary is less than or equal to 50 k okay. So here I obviously have salary less than or equal to 50 k. So let me make this circle a little bit bigger so that it will look good. So here I have salary is less than or equal to 50 k. Now with respect to this I just have two situation yes or no. Right. Let's say that this is my yes output and this is my. No output. Okay, so here I have basically yes and no. This is perfectly fine. Now let's go ahead and see over here with respect to

salaries less than or equal to 50 K. If it is how many of them are. Yes. So here you can see that less than or equal to 50 k. Here I have a specific record I have one. Yes. Right. So I have two years right. So obviously I have two years. So here I'm going to basically write two years okay. This is fine. But when less than or equal to 50 K is there. Here I'm also getting no's right. So total number of no's will be how much. One no and one more no is over here. So two no's right. So here obviously some kind of misclassification has definitely happened. Okay I'll talk about that as we go ahead. Now with respect to this, the total number of yes and nos that you have is basically two yes and two nos. Now what about the other case when the salary is greater than 50 K? Right over here you can see I have one one. No. And I have two. Yes. Right. So over here I'm just going to write two. Yes. And one more okay. So let's say this is my decision tree stump one. And similarly I will try to create my decision tree stump two with another feature which is called as credit. Okay. Let's say if this this feature is basically having credit is equal to good okay. And I hope everybody knows how to create a decision tree. Right. We can use different different lesson based on this features that we specifically have okay. So here I have two uh if the credit score is equal to good I may have yes and no. So let's go ahead and see. This is my decision. Tree stump one. This is my decision tree stump two okay. Now but out of this I'll be only selecting one. Right. And from the first instance, because in this, uh, in AdaBoost we create weak learners, right? Weak learners. We combine it together. And in order to create this weak learner, I can use any of this feature. It can be salary or it can be credit also. Okay. But out of this how do we select. We basically use entropy okay. But that I'll discuss about it. Now let's say if my credit score is good then how many number of yes I have and how many number of no's I have. So if my credit score is good, I have one year, two years, three years, right? No number of nos are there. So here I have three yes and zero nos. Similarly, in the case of whether my credit score is not equal to good then I have one. No two no three nos. Right. So total three nos. And how many. Yes. Uh one. Yes. Right. So one no is definitely there. So one yes and three nos I'm actually getting okay now this is fine. Okay I can construct I can construct this kind of, uh, many, many decision tree stump. Okay. But understand from this two how which one is the decision tree stump that needs to be getting selected as my first decision tree stump in AdaBoost. Okay. Because now, like this decision tree stump, I can create many, right? Because I have lot of features over here also, right? Let's say that if this two are there now, the next step we will try to understand which decision tree stump needs to get selected. And obviously for this we will be using something called as entropy. Entropy okay. Or Gini index or Gini impurity. Gini index and impurity are one and the same. We can use either of these two. And we can understand that which is having an impure split. Now in this particular case, since you have 50% of probability saying yes and 50% of probability saying no, the impure, this is completely impure split at this point of time, if I try to calculate with the help of entropy, what is the number that is going to come? Just guess, just pause for a second and just understand. This particular value will be one for entropy and 0.5 for Gini impurity. Similarly over here two yes and one nos. Obviously this is also impure split. In this particular case I have a pure split and I have a little bit smaller impure split over here. So obviously if I try to compare where I'll be getting a better Entropy or Gini index, I will be getting for this second decision tree stump. Right. So this is how we go ahead and select it. Otherwise if you didn't, if you if you just want to think that okay, if I do I have any maths formula. Yes. So entropy can basically be used for calculating like this. Right. So this already we know the formula which I have already discussed in my decision tree classifier. Right. How to apply entropy. And similarly Gini index I think you know the formula. Either I can just revise the video with respect to the decision tree classifier. So in the first step what we did is that we created decision tree stump. And we selected the best decision tree stump. Using what technique? Using entropy. Entropy or Gini index, right. Or Gini impurity. So this is what we did in

the first step. Always remember our first step is basically to create all the decision tree stump and select the best decision tree stump as we go ahead. Okay. So uh, in this first case we have done that. Now in the second, uh, in the second step we are going to do some more amazing thing. But in the first step we understood that. Okay, fine. We created a decision tree stump and we have selected the best one. So let's go ahead and understand the second step.

Now let's go ahead and try to discuss about the second step. In the second step, with respect to the Adaboostclassifier machine learning algorithm, we are basically going to perform two different operations. One is sum of total errors and performance of stump. Okay. Now here you have already seen that I had created two stumps. And out of this this got selected. How. Because we used entropy or Gini impurity. Right. And like this I can have many number of stumps when I'm creating the first decision tree stump, out of which whichever will be having the best entropy. Uh, that one we are going to select, because the best entropy will basically say that you're getting a pure leaf node very quickly. Okay. Now coming to the next step, I'm just going to consider the same stump over here. And I'm going to place it over here. Now the first step, in order to calculate the total error and performance of stump I need to assign some weights okay. So here let's go ahead and assign some sample weights. Now how do I assign a sample weight with respect to this same data set. Okay. First of all you go and count that. How many records are there. So here you have 12345677 records. So what do you do is that you just assign equal weights to all the records. So this will be one by seven, one by seven, one by seven, one by seven, one by seven, one by seven and one by seven. I'll tell you why we are assigning this particular weights. Because this first component that you see some of total errors. Right. That is what the sample weights will actually help us to calculate this total errors. Okay. Now when we took this particular stump, when my credit score was good, I got three yes and zero nos. In the case of no I got one yes and zero nos. Right. But over here, with respect to our data set, I don't have any scenario where my credit score was good and I was getting no as an answer. So this one yes is basically my error, right? This is my error. Right. Because here you can see when my credit score is good right. Every time I'm getting yes I'm nowhere getting nos. Right. And if I probably see my credit score is not equal to good at that time, you can see that most of the time I'm getting nos. So this can definitely be considered as an error, right? One yes, I'm actually getting now this is with scenario because this is with scenario. You can see that when my credit score is normal, you can see that I'm getting. Yes. So I can definitely say that with respect to this particular decision tree stump, if I train with this specific data set, I'm going to get an error and the error will be this record, okay. And the error will be this record. Now, when I have this specific error with respect to this particular error card, okay. Now we will go ahead and calculate the total error okay. So coming to the first step let's go ahead and calculate the total error. Now if I really want to compute what exactly is total error. So suppose this is my total error.

I'm just going to use the total number of wrong records. Okay, so how many wrong records are there? In this particular case I have just one wrong record. Okay, so this is my wrong record that I'm actually getting. Okay. Now, in this particular case, let's say that okay, I've just considered this is the wrong record that I have got. Let's say in this particular case the sample weight that you have is one by seven. So I'm just going to sum up all the total errors okay. Sum up all the total errors okay. When I say summing up all the total errors, that basically means I'm just going to add up all the sample weights of all the wrong records. Now in this particular case, I just have one wrong record. So my total error will be one by seven. Okay, so this is my total error okay. Because that many number of wrong records I have, I just have one wrong record. So I'm basically going to get one by seven. Perfect. Now let's go to the next step. Now in the next step, what we are actually going to do over here is that, uh, we are going to calculate something called as performance off stump. Performance of stomp, because we really need to find out how the stomp has basically performed right? How many wrong records since how many wrong records. It is basically getting created and based on that, what is this performance of this specific stomp? Now, in order to calculate the performance of the specific stomp, I use a formula which is like one by two log to the base E one minus total error divided by total error. Right. What is my total error over here? My total error is nothing but one by seven. So if I probably go and compute this then basically mean log to the base E. Okay, so if I probably write over here ln of one minus one by seven divided by one by seven. Right. So if I probably try to calculate this I have already done the calculation. But I would suggest you do it by yourself. Log in of six. This is approximate going to be equal to 0.896 okay. So here you can see that with respect to this particular stomp the performance that I'm actually getting okay. It is somewhere around 0.896. Now you need to understand what this .896 basically specifies. Okay. What this .896 specifies okay. So I'm just going to write over here the performance. Of stomp. Is nothing. But it is approximately equal to .896. Now as we go up guys I had created a function right function for AdaBoost. Right? What is this function? I'll just write it down over here okay. So let's say my function is f is equal to. All for one. And let's say this is my model one. That is my decision. Tree stump two, Alpha 2M2. Like this I have a lot of different different values okay. This alpha basically indicates weights. And the performance of this first stump is basically the performance of this particular model. So here my alpha one I'm going to assign it to something called as 0.896 okay. So this is my alpha one. Or I will be saying this is my weight. And for this model one this is my decision tree stump okay. So this is basically my M one. Or I can also say or decision tree stump. Okay, so I hope you are getting an idea about how we are relating it to now. What we have to do is that see in in in in a specific, uh, you know, whenever I have a weak learner, let's say this is my weak learner, right? And we have created this weak learner. Right. So this is my weak learner that I've actually created it right now. This will obviously do have or predict some wrong records. Predict some wrong records. Now I'm going to make sure that I pass this next record. So pass this wrong records to my next decision tree stump. And that is how this are sequentially connected. Right. So that is what I'm actually going to do now. Now I need to understand that what is the next step. Now you you you basically need to understand that okay, fine. Now this is done. If you are able to find out the performance of the stump over here with respect to the first decision tree stump, then how do I construct this particular second model and how do I set up alpha two so that we are going to understand in the next video. So I hope you have understood till here. If not, please make sure that you revise these things. Okay. Thank you.

We are into the step three of basically constructing a AdaBoost classifier, uh, which uses decision stumps as its weak learner. In a second step, we have already computed two amazing thing that is the total errors and the performance of the stump. And we also understood that why this performance of stump is important, because this is basically telling the weight that is needed to be assigned to the first decision tree stump. Okay, so that is what we computed it. Now in the upcoming steps, we need to make sure that whichever are the wrong records that needs to be sent to the next decision tree stump. Okay. So for that we need to perform two important step. First is update the weights for correctly and incorrectly classified points. Now what does this basically mean? Over here we have understood that our incorrect classified point was this one. Right. So over here right 50 K normal. And yes this was incorrectly classified point from my decision tree stump one. Now we need to update this weights. We have to make sure that whichever is the classified points, we have to reduce the weight and whichever is the incorrectly classified points we have to increase the weight because of that. What will happen is that there will be a high probability of selecting the wrong records by the next decision tree stump. Now you may be thinking how this can basically be done. Okay, so let's go ahead and let's understand the math behind it okay. Now first statement I'm basically going to write over here. So here let me write updated weights okay. So this will be my updated weights. Or let me just write it quickly over here. Something like updated weights okay. Now updated weights. First of all for correct classified points. Okay for correct classified points. You know that what we need to do, we need to decrease the weight. Because I really want to increase the probability of selecting the wrong records by the next decision tree. So what we need to do is that for the correct records we will try to decrease this weights okay. For this weight we have to for for this record we have to increase the weight okay. So for correctly classified points we use a different formula. And for incorrect classified points we use a different formula. So if I really want to find out what is the formula of the weight updation for correct classified points, I will just give a simple notation that is nothing but weight. The older weight multiplied by e to the power of minus performance of stump. Performance of stomp. Okay, so what is my previous weight that we need to compute it over here. So I will be getting one by seven multiplied by e to the power of minus performance of stump, which is the performance of stuff before we got somewhere around 0.896. So here I'm going to basically use .896. So this output that you are going to get is somewhere around 0.058. I have already done the computation. So whichever are the correctly classified points, all the weights will get updated to .058. This will also be .058. This will also be .058 and this will also be .058. This will also be .058. This will be .058. But what about this? Now in order to update this specific weights for the incorrect classified points. So weight updation for incorrect classified points. Incorrect classified points. We have a different formula okay. And the formula is nothing but older weight multiplied by E to the power of performance of stock. Before it was minus. Now it is plus. Performance of storms right before here you can see it was minus right. So here I'm just going to apply. Over here it will be one by seven multiplied by E to the

power of ah. What is the performance of stump. Performance of stump is nothing but it is 0.896. So here I'm going to basically mention 0.896. So overall if I go and see it it will be coming somewhere on 0.349. Okay. So this is my .349 will get updated over here okay. Now this is super super important guys. The third step is the updating the weights for the correctly and incorrectly classified points. So here you can see that now all my correctly classified points have we have decreased the value over here by. Over here you can see that it is decreased. It has decreased. It has decreased here also decreased here also decreased. But this has specifically increased. So I'm having .349 and this has also decreased okay. So this was about the third step. Now in the fourth step we have something called as normalized weights. So let's go ahead and let's see what exactly normalized weight is all about. But I hope you have understood what you have done over here. In short, we have just updated the weights with respect to the classified and unclassified points. Okay. Please make sure that you remember the formula. This is nothing but older weight multiplied by E to the power of minus performance of steps. And similarly for the incorrect classified points, I have weight multiplied by e to the power of performance of step right. So let's go ahead and let's see the fourth step. Thank you. Bye bye.

We are going to continue our discussion with respect to AdaBoost. And in this part, we are going to discuss about how to create normalized weights and assigning bins. Okay. Now in my previous video, we have already seen how the weights needs to get updated. The reason we are updating the weights to make sure that the in classified points, you know, uh, from the decision tree stump, one uh should go towards the decision tree stump two uh, so that is the reason we have increased the weights, which was incorrectly classified for those specific data points and decrease the weight for the classified, uh, for the correct data points that were classified. Now, in the next step, we will go ahead and try to understand about normalized weights, computation and assigning bins. Now what happens is that guys when we have this all the updated weights. So in this same updated weights I have written over here, you'll be able to see that if I do the summation of all these weights it will not be one. Okay. If you probably go and go ahead and do the summation, it will be somewhere around. If I probably add this up, all the weights, it will be somewhere around .697. Okay. But initially when we assign the sample weights, we could see that all the summation was actually one. Right. So we really need to normalize this to entirely to come as a total computation. If I try to do the total summation it should be one, right? So that is the reason we go ahead and do something called as normalized weights. Now normalized weights. How to basically bring up all this to the scale of one itself. All I have to do is that for each and every term that I am considering over here, I will divide this by 0.697.697. Is my this specific total value, right? So if I really want to normalize it and bring this entire summation to should be near to one, I will just go ahead and try to normalize this weight by dividing all this value by 0.697. So after dividing, if, you probably divide .058 divided by 0.697, you are going to get somewhere like point uh .08. So like this, all this value will be point zero, 8.0, 8.0, 8.08. And this value, which is 0.349, you know, this will probably get converted to something like 0.50. And

again it will be point zero. It all I'm doing is that for every number I'm just dividing by 0.697. So when you do this you will be getting normalized weight. And now when you try to find out the summation it will be equal to one okay. Now. This is the first step. Now come, let's come to the second step. Now see, our main target is that after decision tree stump one is created. Right? Suppose if this are making some incorrect. If this are doing some incorrect classified predictions. Right. We need to send that wrong records to my next decision tree stump. Right now we need to find out a mechanism like how do I say and make sure that my machine learning algorithm sends only those records, or mostly those records that are incorrectly classified? Okay. So for that particular case, we will be assigning some bins okay. So here I'm basically going to write bins assignment. Now this bins assignment is super important guys because here we are going to create a range of bins. So the first range will be between zero to .08. Because since this is .08 right then the second range will be between .08 to .16 okay. The third range will be .16 to .24 because we are just keep on adding it. Right. So from here .16 again I've added this. So it is this range. Similarly .24.3 2.3 to 2.40 okay. And over here now you can see this range will now get increased .40 to .50. And finally here you can see that it is .50 to .58. Right now you can see that the maximum bin size is basically this specific value right. Which is between .40. Sorry this should be changing a little bit. So this is nothing but .40 to .90 because I have .50. So here you can definitely see that this is my maximum bin size. Now. Now my target is that whichever are the wrong records. Now here you can see that since this is wrongly predicted by my decision tree stump one here my range is quite huge. So I will try to find out a mechanism wherein I will try to send this records more frequently to my next decision. Tree stump one. Sorry. Next decision tree stump two. Right. That is what we are trying to do. And based on that, my next decision Tree Stump two will be able to train the wrongly predicted records. So that is the reason we have basically done the bin assignment. Now in the next step, we are going to find out a way that what are the records that are going to get captured to go to the next decision tree stump one okay. So again, let me revise it over here. So this was my decision tree stump one which I have already created. And you know what is the alpha one value that I got. Alpha one weight was 0.896. So this is alpha one value is .896. Now we did many steps after this. We updated the weights, we assigned the weight, we normalize the weight and we created a bin. The reason we did it is that because we wanted to send the wrong records to my next decision tree stump. Now before sending this, I have to prepare my records over here, right? I have to prepare which records needs to be sent, which data points needs to be sent. And that is the reason why we have created this bin size. Okay. Now in the next step, what I will do is that in order to send my records to the next decision tree stump toe, how I'm going to select the records from this, we are going to understand. So I hope you are able to understand this. These are the steps. And still one more final step is there so that we prepare our new record and the entire entire steps that we have done from step 1 to 4, that all will get repeated. So let's go ahead and let's see the next video where we'll try to create our data set that needs to be sent to our next decision tree stump. Thank you.

In the step four. We have created this bins and we have assigned some bins size. Okay. How did we assign. We just kept on doing the summation with respect to this normalized weights. Now how do I select my new data points to send to the next step. Okay. So that is what we are going to discuss in this video. So we will continue an iterative process. Let's continue an iterative process. In this iterative process we will be selecting random values. Random values between 0 and 1. Okay, so we'll be creating some random values between 0 and 1 okay. Let's say in the first instance I'm just going to create all my new records that are going to get selected from here. Beans or this beans assignments. I'm just going to create a dark Division line over here. Let's say these are my features. I have my salary. I have my credit, okay. I have my credit so that I don't take much space. I have my approval. Okay. And let's say this is my random number that is getting generated. And this random number will be between 0 to 1. Let's say for the first instance in the iteration. And I have to do this iteration till I get all the seven records over here. Okay. So what I'm actually going to do over here, or let me just write it or write down this columns once again. So here I have salary credit approval. And finally, my random number and random number is getting generated between 0 to 1. Now let's say if my random number have got generated as 0.50. Now in this particular case, we will go and see in the bin assignment where this 0.50 range will come. And here you can clearly see that it is between 0.40.0.90. So here obviously your incorrect predicted records are getting selected. So greater than 50 K normal and yes has got selected okay. And whatever records are getting selected this will get passed to the next decision. Now let's say that my random number that is getting generated is one zero. So .10 is somewhere here. So here I have less than 50 K. Uh, green. Uh sorry. Not green. Good. And yes this is got selected. Now similarly let's consider the my random number that is generated is .60 again. So .060. Now see over here. Since the gap is more right there is a high probability this row will get selected again and again. So here I have greater than 50 k normal. Yes. Right now the weak learner the next weak learner will focus on this kind of records. Let's say after this again .75 is my random number that I've got generated. Then where is .75 that is coming over here. Right. Then again this will get repeated greater than 50 K and end. Yes. Let's say it is again another record that is got generated is .24.24 is nothing over here. So this record will get selected less than 50 K green. And yes okay. So how many records 123452 more records will get selected. Let's say that I am going to basically write .32 over here. .32 is the random number that got selected. So .32 is coming over here. So I will uh this record will get selected which is greater than 50 k b and no. Okay. And finally let's say again over here, point uh, .90.87 has got selected. So again it you can see over here this incorrect records will again get picked. So greater than 50 K normal. And yes. Now here you can see one thing is that this record that you see right is getting picked up again and again. So here I have one record which is my incorrect classified point. And then here I have two more records which is incorrectly classified points. And finally, here you'll be able to see one more record that is incorrectly classified points. So these all are getting selected again and again. Right now what will happen is my this entire records will be now sent to my next decision tree stump. And then once this entire record is getting selected. So let me just select this entire thing. This will be my new data set. Okay. That is going to get which will be my step six. The step six. What we are going to do is that this entire records, this records. Will be sent. For training sent to the next decision tree stump. So here, if I'm pasting it again, the same process will start, wherein we'll start with something called as sample weights. Okay. Let's say over here we are going to again create a sample weight. Again sample weight will be assigned by one by six, one by six, one by six, one by six, one by six, one by six, one by six. Then all the steps that we have done from step two to step six will keep on repeating, okay? Unless and until we

don't finish up all the decision tree step sequentially by default. If we do practically, you know they are around 100 decision trees term that are just selected ran. Uh, it is initialized in the initial stage when you're training the model. Okay. Then again sample weight will get updated. Then again you'll after creating the sample weight you'll again try to find out the total error. Then you'll try to find out the performance stump. Okay. And let's say for this particular model you got performance stop as 0.65. Then what will happen again. Your function which was having alpha one, model one which is my decision. Tree stump to alpha two. Model two. Model two basically means this will this particular decision tree stump will get selected. And here your alpha two value will be 0.65. And before your alpha one value is nothing but point somewhere around what it is .896. So you will probably combine all these values and we'll try to see how the output will be predicted. Okay. So similarly we keep on continuing creating multiple stumps. And we do this particular step again and again okay. Now in the next step we'll try to see how the prediction will basically happen and how the entire prediction will happen with respect to a new test data that we are getting by using this decision tree stump. Okay, so yes, I'll see you all in the next video. So I hope you are able to understand all these things. Okay, so in this step we had focused on selecting how the next data points needs to be selected to the cell and send it to the next decision tree stump. Thank you.

Now let's understand. The final step is that how does the final prediction actually happen with respect to a new test data in AdaBoost? And this is specifically with respect to a classification problem. Now let's say, uh, we try to solve this with respect to this particular problem. Let's say that I got a new test data where it is saying the salary is less than 50 K and uh, the credit score is good, then what happens is that let's say this particular decision tree is saying, okay, fine. Here my output is yes, here my output is no decision tree stumps. Let's say this is my decision tree stump one, decision tree stump two decision tree stump. And like this I can have any number of decision tree stump. Right. So this we have already discussed let's say this is saying yes. And let's say this is saying no. Okay. Now here, as I said that like random forest, we don't focus on majority voting classifier here. One important parameter that we have discussed with respect to this, we have an uh, weight called as alpha one. This is basically weight alpha two, alpha three. Similarly alpha. And let's consider that initially we computed alpha one as 0.896 right .896. And this was .650. Let's say I'm just considering and this was with respect to this is uh like a .24. And let's say this alpha N is basically giving me a weight off point. Or it can also be a negative value. So let's say it is minus .30. Right. So final function that you will be able to see is is nothing but alpha one model one plus alpha two, model two plus alpha three model three like this. Plus alpha four. Model four. Let's say this I'm considering it as alpha four okay. So here you can see that what is alpha one? Alpha one is nothing but 0.9896. Model one is basically saying us yes okay. Let's say alpha two is 0.650. Model two is basically saying no. And uh, alpha three is nothing but .244 over here. And model three is basically saying yes. And finally alpha four is basically uh, -3.30, minus .30. And this model is basically saying no as an output. Right. So once we do this we just need to combine what is the value with respect to. Yes okay. With respect to yes. If I probably add up .896.24. So how much is .89, 6.2406 1311. And this is one. So here I'm getting 1.136. Plus how much is this. No this is nothing but .6590 minus .30. So here I'm going to

get .350. And here this will be no. Now here you can see the maximum number right. Whatever it is saying over here 1.136 is maximum more than .350. So definitely the entire output of this entire weak learners which is combined as a strong learner here the maximum output is yes. Since I get the performance of say we also see it as like this performance of say. With respect to yes is high. That is 1.136 performance of. C with respect to no. We also give this particular problem statement also. Or you can also say that the weight with respect to yes is very high when compared to no. It is 0.350. So this is obviously greater than this right. So finally our output is that. Right. Whenever I get a test data with respect to less than or equal to 50 K and the credit score is good, the credit card will definitely be get approved. So the final output will be yes with respect to this particular use case okay. Considering this particular data set okay. So this is how the final prediction will basically happen okay. And in the next video I'll also talk about how does the classification sorry regression problem statement will happen. Only two things will be changing with respect to regression here. We will not be using entropy. Uh here will not be using entropy. Instead we'll be using mean squared error okay. To basically compute which decision tree stump is the best. And at the last everything will be almost same. Okay. Everything will be almost same. But here we'll be having a continuous value with respect to some weights assigned to it. Okay. So that is the difference. But I hope you have got an idea with respect to the classification problem of AdaBoost. So here you can see how many pages I've written. This is how you should definitely practice to get the mathematical intuition in a better way. So yes, this was it. I'll see you in the next video. Thank you.