In this video, we are going to discuss about something called as ensemble techniques. And in this ensemble techniques we specifically have two types that is bagging and boosting. So at the end of the video you will be able to understand what exactly is bagging. And we'll also try to understand what all algorithms we are going to learn inside bagging. And similarly we also try to understand what exactly is boosting. Now first of all, let's go ahead and understand what ensemble techniques are okay. Now till now you have learned about so many machine learning algorithms. But we have never seen wherein we combine multiple machine learning algorithms for doing some kind of predictions. Okay. So Ensemble Techniques actually helps us to combine multiple algorithms together. Probably train that specific kind of model. And then with respect to that we get some kind of output. When we are combining multiple machine learning models, there are chances that we definitely get good amount of accuracy. So in most of the platforms that you will probably be seeing or you may have heard about, like Kaggle, they are hackathons, competitions and Kaggle and different, different hackathons, right? And even in different hackathons, many people, while they are solving the machine learning problem statement, they use this kind of ensemble techniques, specifically bagging and boosting techniques. Okay. And uh, with respect to any kind of problem statement, uh, ensemble techniques like bagging and boosting gives you very good accuracy. So now let's go ahead and understand, first of all, what exactly is bagging okay. And uh, what we exactly do in bagging okay. Now some of the algorithms, uh, that we will probably learn in bagging is something called as Random Forest. Okay. Random forest. Now here we'll discuss about random forest algorithm in the later stages. But let's understand the basic. What is the idea behind bagging okay. Idea behind bagging. And then we will probably go and learn about, uh, you know, what is the idea behind boosting? So in bagging, what we do is that. Suppose let's say we have a specific data set. Okay. Let's say that we have a data set over here. And data set can be of any size any number of data points. And it probably has some amount of features and some amount of data points itself. So let's say this is my data set. Okay, so over here, this is my entire data set. Now for this particular data set, let's say that this is my training data set altogether okay. So training data set. Now in bagging technique what we do is that we have multiple base learners okay. We have multiple base learners. Like this may be my separate base learner. This base learner when I say base learners these are specifically different different machine learning algorithm. Or it can be same machine learning algorithm. Let's say that this is my decision tree. Okay. This is my decision tree. And let's say this is my logistic regression for a classification problem. Suppose if we consider for a classification or for a regression problem. So over here, what I'm saying is that we have multiple base learners like this. And this base learners are basically our machine learning algorithms. And we can have many number of base learners like this okay. Now what happens in bagging is that we give some sample of data set to our model one. Let's say this is my model one. This is my model two. This is my model three. Like this I can have any number of models. Let's say this is my model N okay. N. And. Now, what we do in bagging is that we use base learners like this. It can be same algorithm or it can also be a different algorithms. Okay. What we do, we give a sample of dataset to a model one and another sample of data set to a model two. Similarly, another sample of data set to model three and another sample of data set to model N. Finally, we. When we give this particular sample of data set, this model gets trained with respect to this specific data set. Similarly, this particular model will get trained with the other sample of data set. Similarly, model three and model N will also be getting trained like that. Okay, let's consider this is my another model like XGBoost okay. Not sexy boost because the boost is already a boosting algorithm. Let's say this is my new bias. Okay, so these are all algorithms which we have already discussed. I'm just writing it over here. Now after this, you know, once the model is trained with different different sample of data set, then what happens is that when we give our

new test data for the prediction. Let's say if you are trying to solve a classification problem, let's say this is a binary classification problem that we are trying to solve. Okay. Binary classification problem. So what happens is that when we pass this specific new test data, then this new test data will go over here. And obviously decision tree will do some kind of predictions. Let's say this is giving me zero. This is giving me one. This is giving me zero. And this model and whatever algorithm we are using this may be giving you this. Let's say that this is also giving somewhere around zero. Okay. Zero. Now, overall, if you really want to find out the final output in this specific way, then what we use specifically is we use something called as majority voting classifier in the case of. In the case of regression problem statement, that basically means they will go and see that each and every model which we which we are specifically saying it as base learners, what output they are giving, like maximum number of models, are actually giving the output as zero. So our final output based on majority voting classifier will be zero. Okay. This is the idea behind bagging okay. Here we have some specific base learners. We pass some amount of data to machine learning different different machine learning models. Let it be a classification problem. Now in the case of regression problem like if we have a regression problem statement okay. Now in this particular case what we do is that we specifically like in the regression case this output will be a continuous value. Right. So the final output will be the average of all the outputs average of all outputs. So this exactly is basically how a regression problem basically works in the case of bagging techniques. So some of the key points let me repeat over here you will have a data set. And what you will do is that you can take many number of base learners, like how many number of base learners you want. You can basically take up okay, there are options. By default it takes up 100 base learners. If one of the algorithm. If I talk about like random forest, then what we do we take sample of data set from here. Let's say I'm giving D dash over here D double dash over here D triple dash over here. And d four dash over here. So these are different different sample of data set. And when this get model gets trained you know then it will start taking the new data and make the prediction. The final output will basically be the majority voting classifier in the case of binary classification or multiple multiclass classification at that time. Also it will be considering majority voting. So who is voting what maximum amount? Like if this is voting maximum models are voting zero, then it is going to consider zero, otherwise it is going to consider one. In the case of regression, whatever output I'm getting continuously, we are just going to find out the average and that is how the output is coming up. Okay. Now this is the idea behind bagging okay. Now let's go ahead and understand one important thing about bagging. All these base learners are getting trained parallelly. You have to understand this important point. All this base learners are getting trained parallelly okay not sequentially parallelly. So if I now talk about boosting, let's go ahead and understand about boosting. And in case of boosting, they are different algorithms that we are going to learn. First of all, let's say we are going to start with something called as gradient boost. Okay. Then we are going to start with XGBoost. Not gradient boost. First of all, we'll start with something called AdaBoost. Okay, so we are going to start with something called as AdaBoost algorithm. And this bagging and boosting algorithms are used for solving both classification and regression problem. Okay, AdaBoost is there. Then I have something called as XGBoost and before XGBoost we can also use something called as gradient boosting. And then my third is basically extreme gradient boost. Okay. Extreme gradient boost. I will probably be teaching you all these things as we go ahead. Okay, extreme gradient boost. So this is also called as xG boost. So this all algorithms will try to see in much more depth and detail. We'll try to understand how does this basically work. Now let's understand the fundamental idea behind boosting. Okay. So what how does boosting algorithm actually work. Idea behind boosting. How we are going to combine multiple models in the case of boosting. In boosting, like in bagging, we had base learners,

right? In boosting, we basically say that we have weak learners. Okay. So let's say this is my one model M one. It can be anything. And then I will be having my another model M two. Or let me just make one small change over here. Let's say that this is my data set. Go like Mr. Dataset over here. So let's say this is my entire dataset. Okay. And let's say I have separate models. So I have all the weak learners not say this has weight, uh, base learners, but I'll say weak learners. So let's say this is my model M one. This is my model. And to. This is my model M3 and it can be any number of models, right. All these models are connected sequentially. Okay. Sequential. So this is my M1, this is my M1 and this is my M2. Okay. So I can have any number of models. This all models are basically called as weak learners. Okay. Weak learners. These are also called as weak learners. Similarly, in this case, this is also a weak learner and they are connected sequentially. Okay. Understand these are connected sequentially sequentially sequentially okay. And similarly this is also a weak learner. Now the idea behind the boosting technique is that we combine all the weak learner. And finally when I combine this all sequentially we actually get something as a strong learner okay. Strong learner. Now see in this particular case how it actually works in boosting. First of all I will give out my entire data set to model one. Okay, now my model one will train and it will give some kind of accuracy. And it will obviously be able to not like let's say for some of the records it has not predicted correctly, let's say they are around 10 to 15 records that are not predicted correctly over here. Then what model one will do is that it will pass those records that are wrongly correct, incorrectly predicted, along with some more records from this data set to model M2, okay, and similarly model M2, it will train from that specific records, and it will also not be able to predict some wrong records. It will obviously predict some wrong records. So it is going to pass that wrong records to M3, along with some more records from this data set. Okay. And this process will keep on continuing sequentially with some n number of models. Now what happens is that whenever I try to give a new test data, okay, then each and every model will have its own, uh, strength. And based on that, it will be able to give the predictions. Right. Once it is being able to give the prediction. Overall, when we combine all the predictions again the same thing, we can actually include my majority voting classifier. Or we can also get with respect to if you try to find out, the average will be able to get the regression problem statement. So in short, in boosting we are combining multiple weak learners together to form a strong learner. Just in a basic example, if I really want to give, let's say there is a problem statement that involves that involves four to different 4 to 5 different domain. Okay. Uh, that domain can be physics, chemistry. It can include geography and all. Now just understand this weak learner are like the people. Suppose this model is basically very good at geography. So he will be able to give he or she will be able to give some amount of information from this particular problem statement, some solution for this problem statement. Then when we go to the model two, then it will be able to give some solution from uh, from this particular problem statement. Let's say model two is an expert in physics. Similarly this is an expert in chemistry domain. This is an expert in some other domain. So when we combine all this little, little knowledge, finally you'll be able to see that we are getting a strong learner. And we'll be able to get the prediction right. Similarly, in the case of bagging here, what you see that each and every model is an expert with respect to its own specific subset of data. Right. So this model, this model, this model, they are own expert in their own specific subset of data. So definitely they will also be able to give some good solution with respect to any kind of problem statement that we have specifically when I'm talking about regression and classification. So I hope you got an idea about what is the basic difference between bagging and boosting. Now in my next video we'll discuss about we'll start with Random Forest. So Random Forest is one bagging technique which we are going to learn about this particular algorithm. And we'll try to understand how regression and classification is basically solved using random forest okay.

Classification. We'll try to understand the detailed implementation about this and the mathematical intuition. Right. So and after that we'll continue the boosting part. So yes I will see you all in the next video. Thank you.

In this video, we are going to discuss about the random forest classification and regression machine learning algorithm. We're going to understand how this algorithm actually works and how we are able to solve a classification and regression problem statement. So already in my previous video, I've spoken about what is the differences between bagging and boosting. And these are specifically ensemble techniques. One of the example of bagging techniques is random forest classification and regression machine learning algorithm. So let's go ahead and let's try to see how this algorithm actually works. Usually let's consider that I have a specific data set. Let's say that this data set is there. And the size of this data set is I want to mention the size of the data set as d okay a small d. So this is basically small d. So this is the size of the data set. Similarly if I consider what is the size of the how many features I have, let's say that I have total m number of features okay. I'm just giving some notation so that. So there may be like f1, f2, f3, f4 like this up to f m features. Okay. Some n number of features. And obviously this is my entire data set okay. Now in Random Forest as it also uses the approach of bagging only. But over here the base learners that you actually have all the models of the base learners are decision trees. Okay, so this is my decision tree one. Let's say this is my decision tree two. Okay, let me do it again. Much more. Better. And this is my decision. Tree two. This is my decision tree. Three. Okay, this is entry three. Similarly I have a lot of decision trees. So over here your base learners are specifically decision trees. And it can have any number of decision tree okay. Now this is fine. This is okay fine. My base learners are decision tree. Now let's see how this entire and obviously you know how to create decision trees right. We have seen techniques related to regression in classification. In classification we have seen entropy Gini impurity information gain and how we construct the entire decision tree. Similarly, in regression we use something called as mean squared error. Now what happens is that with respect to this particular data set and obviously in bagging technique, I told you that we need to give a sample of data set. So here we are going to do something called as row sampling. Raw sampling. So I'm basically going to do some raw sampling over here. So let me write it down properly over here. So we are going to do some raw sampling raw sampling from this specific data set. Let's say from this particular data set I'm going to pick up some rows along with row sampling. I am also going to do something called as feature sampling okay. Feature sampling. So here in short form I'm just trying to write it as r s plus fs okay. So we are going to do some amount of row sampling. Row sampling basically means we are going to pick some amount of rows from this data set. Not all the data set, but some amount of rows. Uh, let's let's mention this is like D dash. And obviously if I say d dash that basically means d dash is less than d, right? D is the total number of data points. Right. And d dash I'm just picking up some of the rows and similarly feature sampling. Let's say that in this I'm going to pick up F1F2F4 and probably give it to this specific model. That is Decision tree one, which is my model one. And it will get trained over here right now. In the second case, uh, what will happen is that again, uh, we will take up some more data set and give it to our decision tree tool. So here again we are going to do row sampling with replacement

okay. Please make sure that you remember this word replacement. Why we are doing replacement. Because some of the rows may get repeated from here. But we are also replacing some more rows right. So row sampling we are doing, but we are also replacing it so that all the rows should not be same when compared to the model one rows that our training data set that we have given. Right. So here we'll do rows uh sampling with replacement. Similarly here also we'll do feature sampling with replacement. So in this particular case I may give some other features F1F2FFIF6. Yes. Some of the rows and features may get repeated, but that is fine okay. But majority of the rows will get replaced. Majority of the features will get replaced. The same thing we are going to do for this particular data set also. So here also we are going to do row sampling plus feature sampling here. Also we are basically going to do row sampling for feature sampling. Now all these decision trees. One important thing is that in Random Forest we use only decision trees. Right now. All these decision trees will get trained with respect to this all training data set. Okay. Now whenever I give my new test data, whenever I try to give my new test data, let's say this is my new test data. New test data. Now, whenever I give my new test data, what will happen is that, first of all, let's say decision tree one. In the case of classification, it gives my output as zero. This also gives my output as zero. This also gives my output as zero, and final tree basically gives the output as one. Now in this particular scenario, what we are going to do is that we are going to take the majority voting classifier in classification. Okay. So here I'm going to basically write in classification. You'll be seeing that the final output will basically be the majority voting classifier. So. Voting classifier. Right. So here with respect to classification we do this similarly with respect to regression. If this is a regression problem statement, then the output will be continuous. So in regression we basically calculate the average of all the model output okay. So again I'm going to write this two specific conditions over here. So with respect to classification. So with respect to classification. Majority voting classifier will actually happen. Majority voting. Classifier. Will happen in case of regression, we are going to just find out the output. So in the regression output of the models sorry average output of the models. Here I'm just saying going to write average. Output of the. Models, right? So this is how we are going to get the output. Now this is fine. Very simple row sampling feature sampling okay. All the models are getting trained. There was one very important interview question, uh, that was asked to some of my subscribers, you know, when they went for the interview and that is that. Why should we use. Why should we use? Random forest. If decision instead of decision. Tree. Okay. Instead of decision tree. Now understand why I'm specifically asking this question, because obviously to create this many trees and probably train it right, the time complexity is usually high, right? In this particular case, what is the major disadvantage that I feel the time complexity to train this model will obviously be high, but accuracy, it will be better because I now I have so many base learners and so many experts over here. Right. So decision tree one is a separate expert. Decision tree two is a separate expert. So maximum number of people what they are saying or maximum number of models what they are saying. We are going with respect to that specific output. But to train this model time complexity will be huge. Okay. Now the question was why should we use random forest instead of decision tree? So let me go ahead and write about decision tree. If we try to create a decision tree by its default parameter, then we know that it leads to something called as overfitting. Overfitting. Missingness. The training accuracy is high. And the test accuracy is low because we try to divide or we try to split our entire decision tree. Okay. Till it's complete debt. So our like by default it is going to do overfitting. If we don't do uh let's say uh, if we don't do post pruning or pre-pruning. Right. So there are high chances that it may overfit. Right now, whenever I am actually overfitting, I have a I. In the case of when the training accuracy high, I basically have low bias, and in the case of when the test accuracy is low, I basically have high variance. Right. So this high variance. But if I talk about a generalized

model right, a model a suitable model, if I talk about a generalized model, this model should basically have low bias and low variance. Now. Our main aim is that this low bias should be low bias. Okay. And we will try to convert this high variance to low variance with the help of. Random forest. Okay, so with the help of Random Forest, we are going to do this now. How it is going to happen? Just just think over it, how it is going to happen since we have multiple decision trees, okay. Since it has multiple decision trees. And one more very important point that you can note over here is that for each and every decision tree, I'm giving a separate set of records, right? Yes. Some of the records some of the rows may get repeated. Some of the features may get repeated, but we are giving some amount of data set to M1M2. Similarly, I'm giving some data set to M2M3 men. Right. All this particular models. Right. So this models are having some amount of information with respect to different number of sample of data set. So obviously this with respect to the new test data. New test data we want our test accuracy will not go down but instead it will go up now because why. Because now we all have separate separate models who are expert in something else. Now let me say that let's bring some 200 new records, 200 new records to this data set. Let's say I'm just going to bring this 200 new records, and I'm just going to add to the data set. Now, will my previous trained model get impacted by this 200 new records? I'm just saying as an example, the answer is no. Why? Because if I really want to train this 200 new records, let's say it will get divided between all these models. And just sample sample of data will get here and there. Right. And if we also try to predict those right. Since majority voting will always be working in the case of classification and in the case of regression, we are going to continue with respect to average. So it won't have that much impact even though we add this new set of records, because it is going to get splitted among all the records. So none of the model will get much impacted by that, right? So that is the reason with the help of Random Forest, since we have many, many base learners, this high variance automatically will become low variance because our now training test accuracy will increase because we have many base learners that are trained on in a separate, separate sample of data, right by using this auto sampling and feature sampling concept. Okay, so I hope you have understood about training random forest classification and regression. Internally, it uses decision tree so that it becomes very much easily. Right. So now in this decision tree we can set up different uh, this is the entire separate model. This is a entire separate model. Because here we are also changing different different records over here. Right. So I hope you have understood this. Uh, and yes, in the upcoming videos we are also going to see about boosting techniques and all. So yes, I'll see you all in the next video. Thank you.