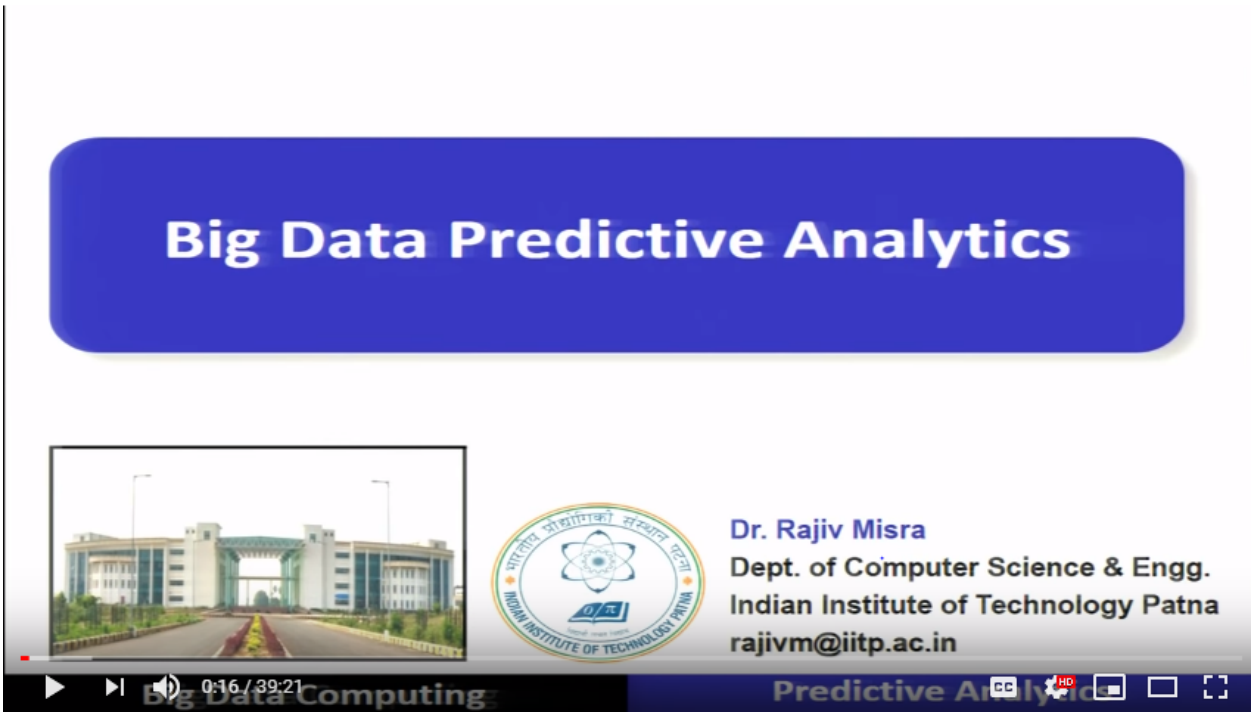
Lecture - 28
Big Data Predictive Analytics
Part - I

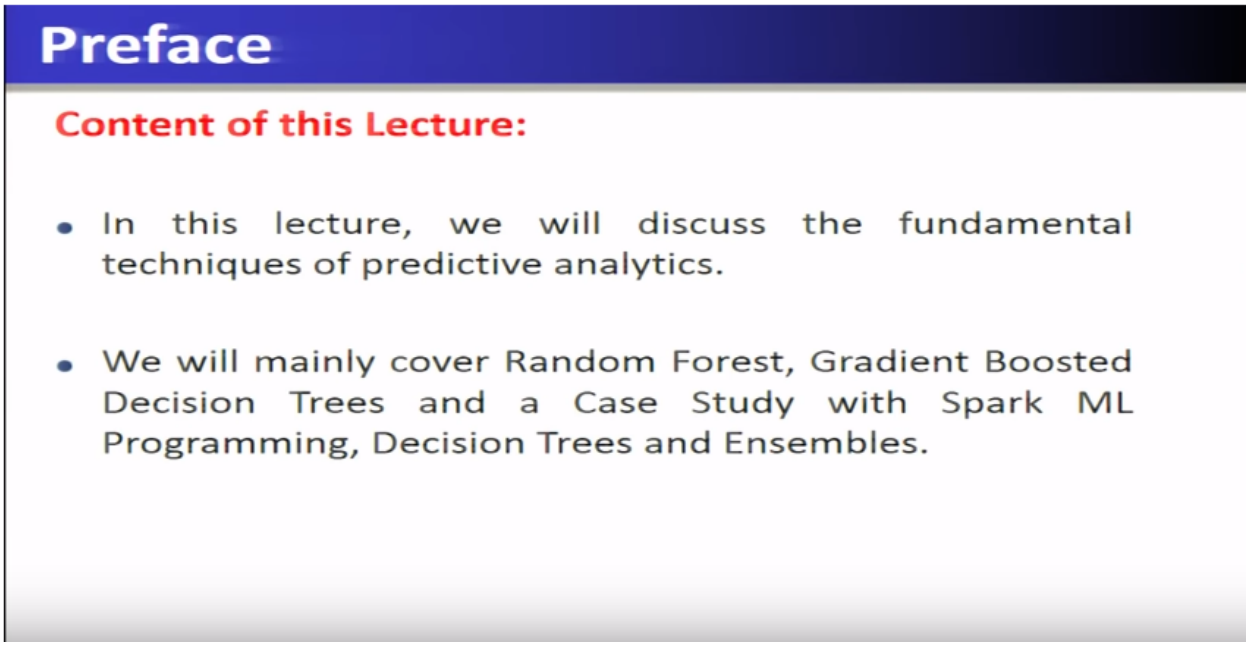
Big data

Refer Slide Time: (00:15)



Predictive analytics

Refer Slide Time: (00:17)



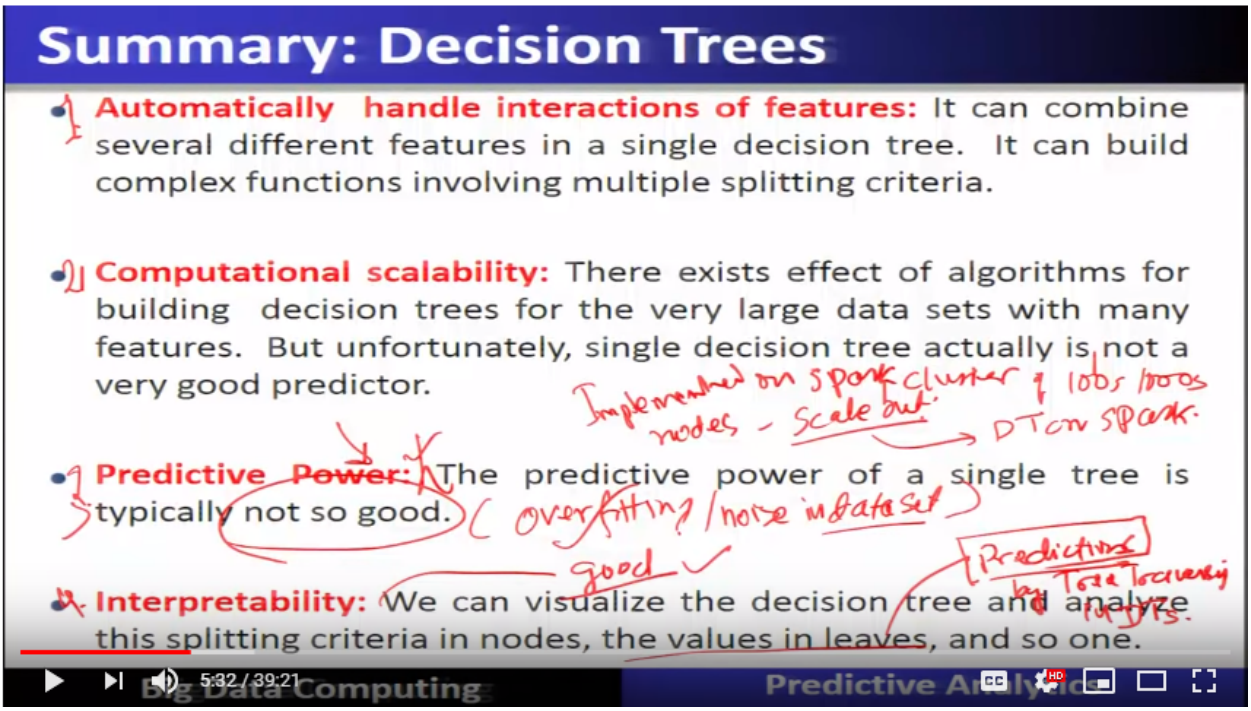
Preface content of this lecture; this lecture, we will discuss the fundamental techniques of predictive analytics, basically we will cover, mainly the techniques such as random forest, gradient boosted decision

trees and a case studies of various implementations using spark ML on decision trees and and ensemble trees learning for predictive analytics.

Refer Slide Time: (00:49)

Decision trees,

Refer Slide Time: (00:51)



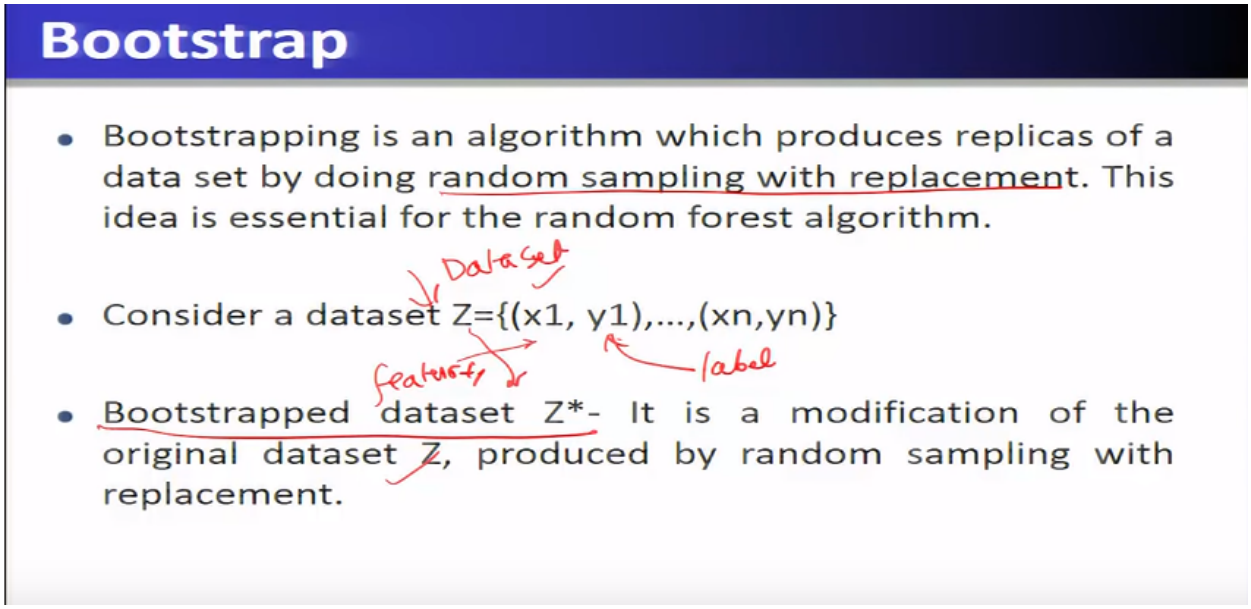
let us summarize the decision tree which we have covered in the previous session. So, in that decision tree, what we have seen is that it automatically handled the interaction of features that is, it can combine several different features in a single decision tree that we have already seen. So, the decision tree which is covered or which is drawn as the tree induction which is called is basically the tree which is splitted on different features, hence there is a interaction or combine different features in a single tree, now it can

build a complex functions involve multiple splitting criteria. So, that means if the tree is very big obviously, it is built based on very complex functions. So, all these things are automatically handled as far as the interaction of the features is concerned, that is the important aspect of the decision tree which we have covered, second part is called computationally scalability, that is there exists an effect of algorithm for building the decision tree, where every large data set and with many features, but unfortunately single decision tree actually is not a good predictor, why because of the overfitting errors and so on, and due to the noise. So, computation and scalability is there, that means this particular decision tree can be implemented, can be implemented on the, the clusters and using a spark system, of hundreds and thousands of nodes which can be further is killed, therefore it has the computational scalability of implementing the decision tree, we have also seen the predictive power. So, the predictive power of a single decision tree typically is not good. So, due to the overfitting, because of the noise in the data set, therefore the predictive power is of a single tree is not typically good due to the overfitting. So, a lot of techniques, we have seen about overcoming from the overfitting and thereby improving the predictive power in the single decision tree and fourth important part of this the decision tree which we have seen is about interpretability. So, we can visualize the decision tree and analyze this splitting criteria of the nodes, hence it has a good interpretability, that means once a decision tree is built, we can understand about the intricacies of the data, with the help of the decision tree rules and we can also do the analysis of the splitting criteria on the nodes and also the values on the leaves they basically are giving the, the predictions on the new data set, by the tree traversing in decision trees. So, all these are basically the summary of a decision tree and we have already discussed. So, important part here is that it has, though it has a good interpretability, but it has not that good not so good predictive powers. So, here we are now going to see if let us say we are going to solve the predictive analytics problems and this single decision tree having not that good predictive power needs to be augmented. So, we are going to see in this part of the discussion, what is the the techniques which can be used for predictive analytics of the Big Data.

Refer Slide Time: (05:34)

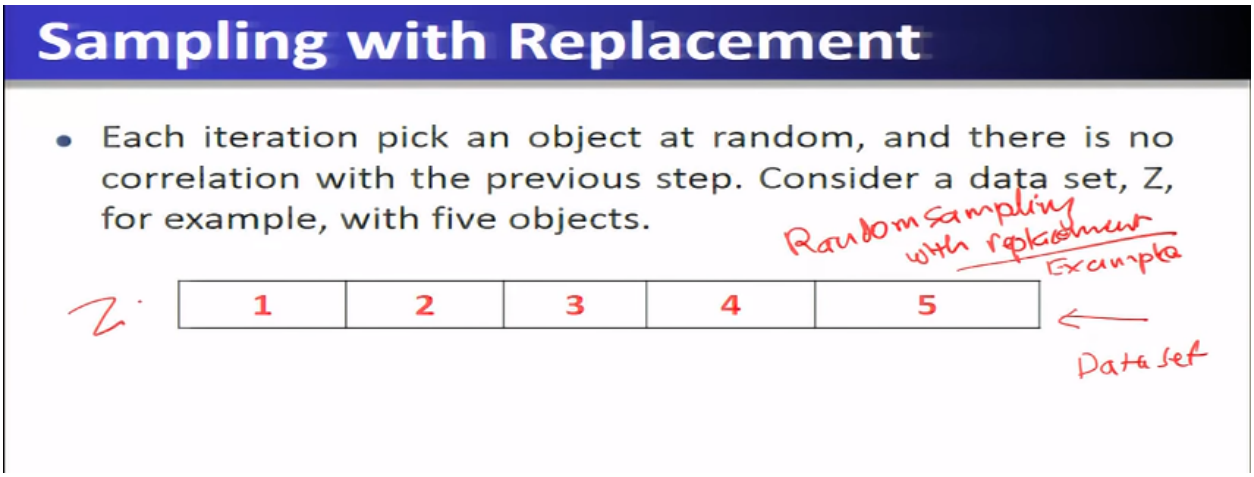
So, before going ahead into the basic techniques, some of the fundamentals which are required to build the new algorithms or which is used for the predictive analytics, we will see some of the features of it, as the bootstrap and the bagging.

Refer Slide Time: (05:53)



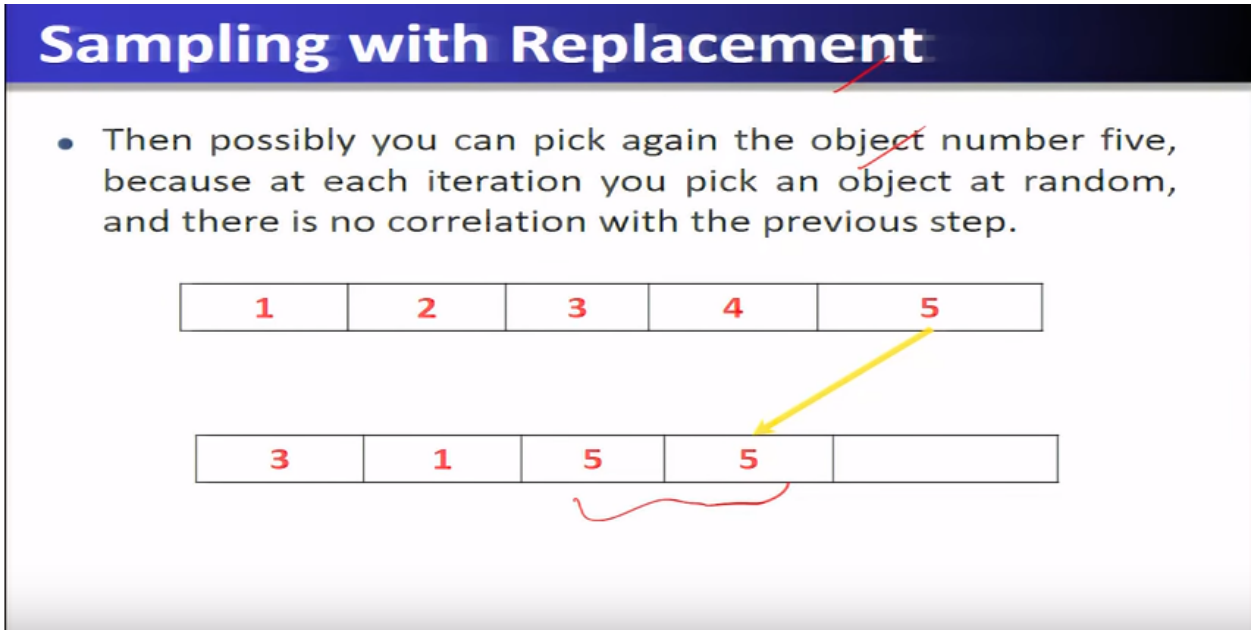
So, bootstrapping is an algorithm which produces the replicas of the data set, by doing the random sampling with the replacement. So, that means we using the technique which is called the random sampling with the replacement, we can build the replica of the data set, that means, if the data set is given we can produce another data set using bootstrap techniques by applying the random sampling with replacement. So, this idea is one of the important ideas or essential for the building the random forest algorithm. So, let us consider the data set Z, which consists of x1 and y1. So, here x1 is the set of features and y1 is nothing but the labels and this particular data set is called Z. So, this particular data set is called the bootstrap data set, if out of this data set, we can generate another one, let us see the bootstrap data set which is built out of this particular data set Z. It is the modification of the original data set produced by applying the random sampling with replacement that is called bootstrap data set, let us understand how this is generated.

Refer Slide Time: (07:19)



So, let us see the random sampling with replacement techniques. So, let us understand this random sampling with replacement using an example. So, let us say that this is the data set, which is given, let us call it as Z,

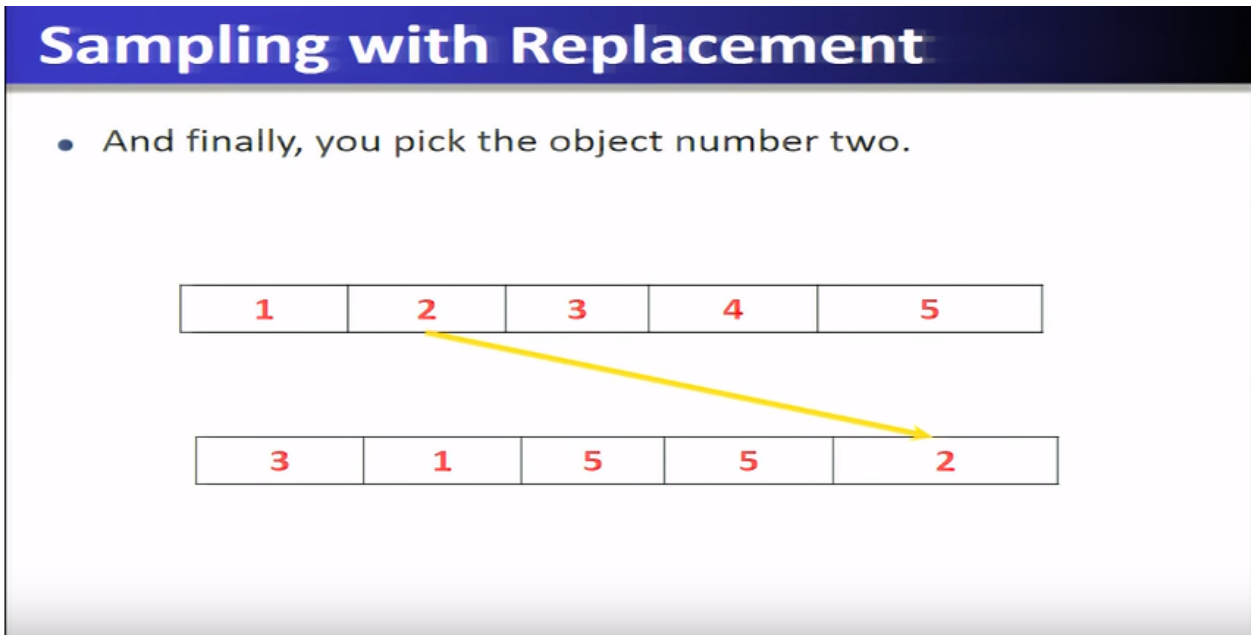
Refer Slide Time: (07:59)



then we can build another data set which is called Z star, by the application of the same data set with, with the replacement. So, let us see that we will randomly pick an object of a data set, let us say that it is an object at number three and this is replicated here in the Z star which is the replicated data set which we are building. Now, next we will pick another object in random, let us say that this time, we are picking the object number one and we will now replicate it at the position number two in the replicated data set. Similarly, we will now pick the object number five and we will replicate at the position number three and

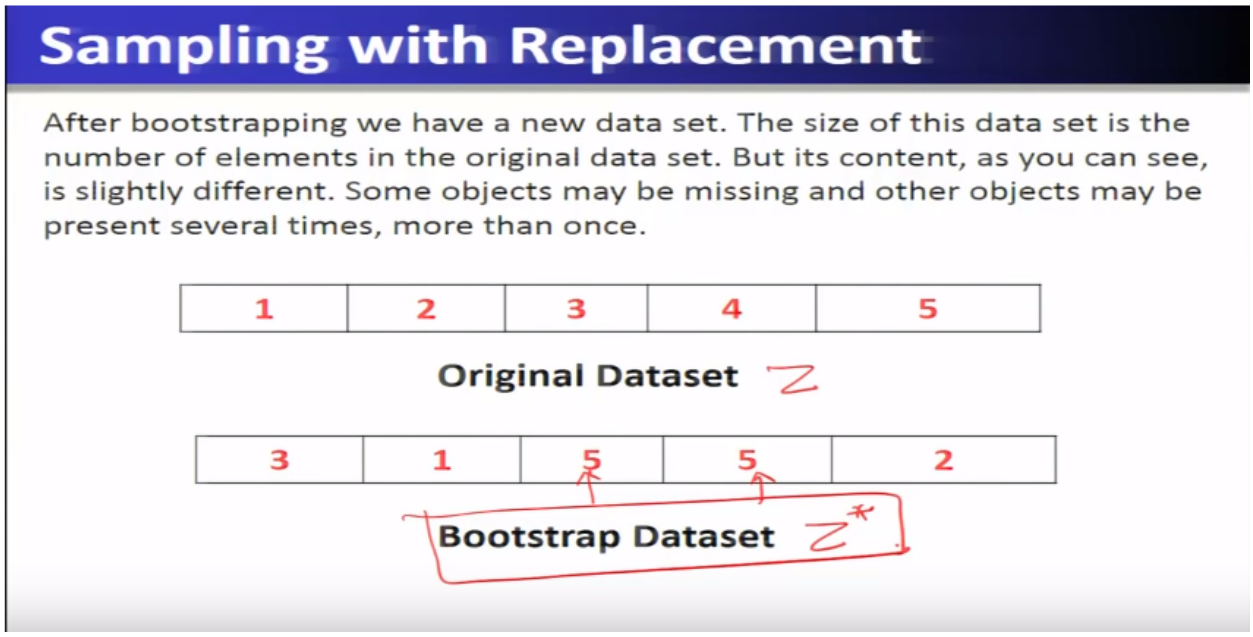
we can also because this is the sampling, random sampling with the replacement, we can pick the, the object number five again, second time. So, it is being duplicated. So, that' allowed in this random sampling with replacement. So, then possibly you can pick again the object number five, because at each iteration you pick an object at random and there is no correlation with the previous step, hence it is called random sampling with replacement. So, therefore five, we the object number five can be picked again

Refer Slide Time: (9:47)



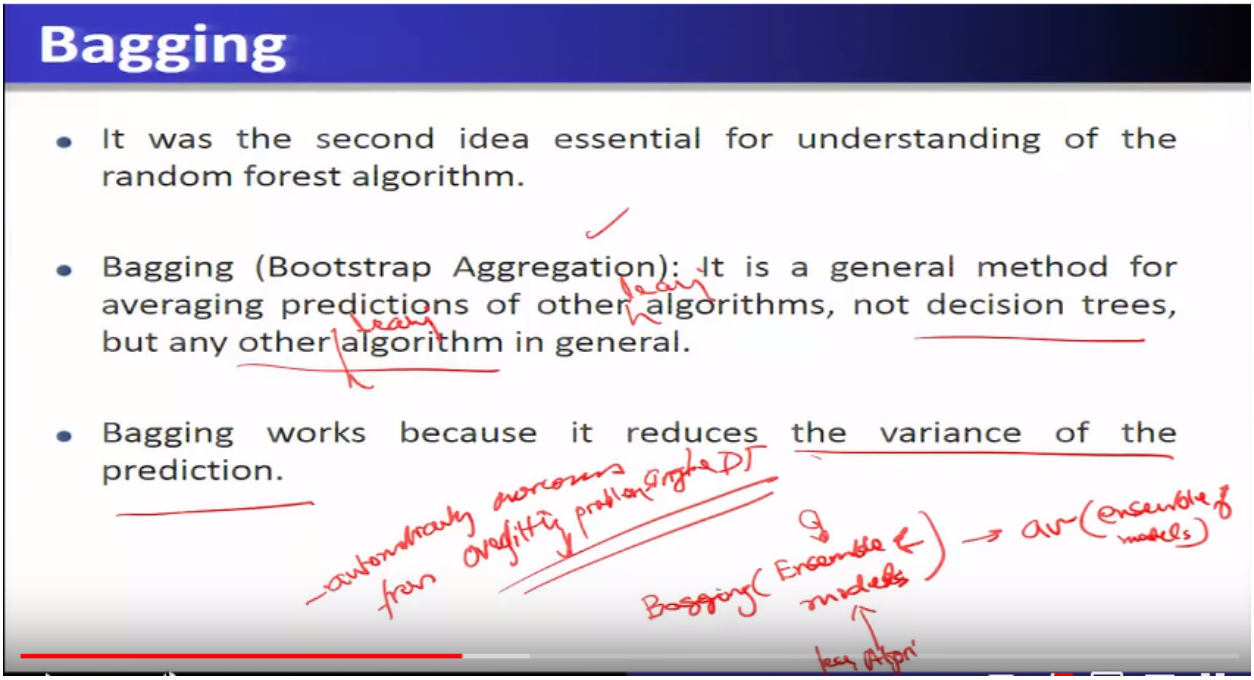
and this is shown in this example, finally we will pick another object, let us say object number two in random and we will replace it, we will replicate it on the fifth position.

Refer Slide Time: (10:02)



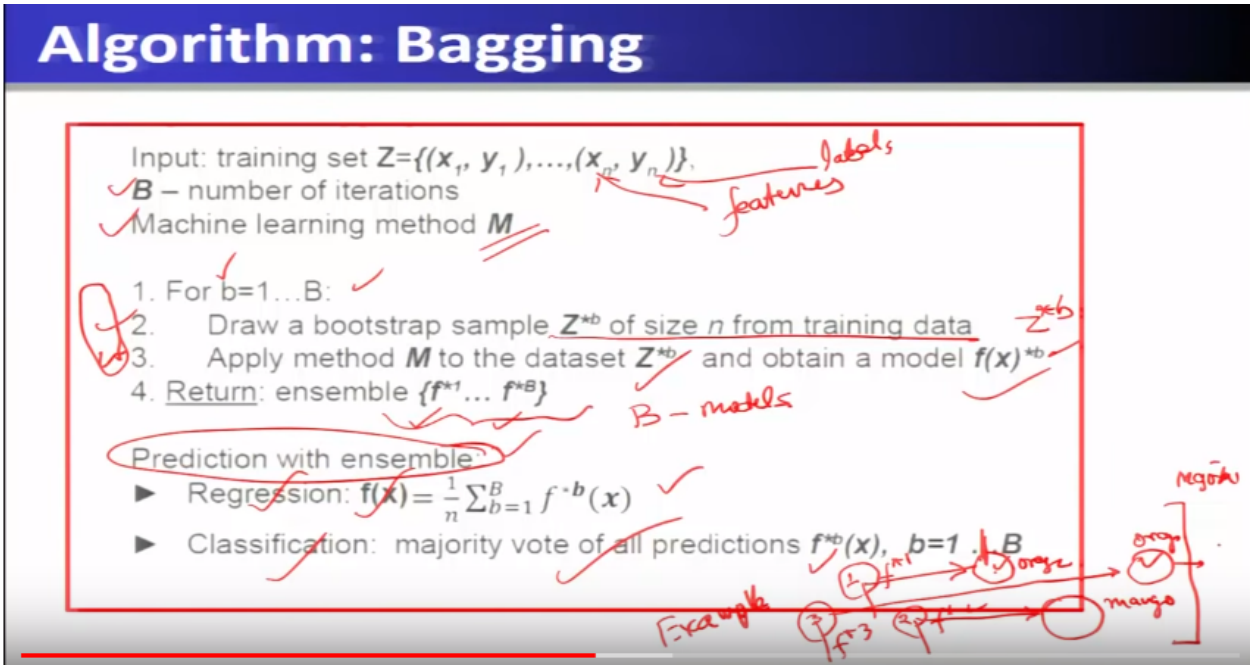
So, this particular data set which is now generated out of the original data set which is called Z, and the bootstrap data set which we have generated called Z star it's called a bootstrap data set. So, after bootstrapping, we have a new data set, the size of the data set is the same number of elements as the original data set, but its contents as you see is slightly different. Now, some object may be missing and in some cases and other objects may be present several times, that is more than once that we see here, that this object is repeated or is being to n, more than once that is allowed in random sampling with replacement to generate the bootstrap data set, this particular data set is called bootstrap data set, which is generated using boot strapping. So, after understanding the boot strapping,

Refer Slide Time: (10:50)



let us see what is another technique which is called Bagging. So, it was the second idea essential for an understanding of random forest algorithm. So bagging, is in short hand it is called bootstrap aggregation. So, it is a general method for averaging predictions of other algorithms and not only for decision tree, but for any other models, it can be applied. So, the bagging works because it reduces the variance in the prediction. So, bootstrap aggregation, bagging is called bootstrap aggregation, or it is the method for averaging the predictions based on the, the different models which are based on different learning algorithms. So, the different learning algorithms not necessarily the decision trees but any, any algorithm, it can be a learning algorithm. So, different learning algorithm will now give out the models. So, this bootstrap aggregation that is called bagging is nothing but everything of ensemble of, of models. So, it will give an average of different ensemble of models. So, these models will be generated by different, these models will be generated by different algorithms, learning algorithms, may be that these models are nothing but an n sample of different decision trees also. So, hence the bagging works, because it reduces the variance of the predictions. So, this particular method reduces the variance of the predictions and by doing this averaging, it automatically overcomes, overcomes from the overfitting of problems of which we have seen in the single decision tree, we will see more details in the further slides. So, bagging is an important idea which is essential to the understanding of the random forests. So, bagging is nothing but a bootstrap aggregation, it is a general method for averaging the predictions of different learning algorithms and models and thereby. So, it is not only confined to the decision tree, but any other algorithm, many any other learning algorithm is used here can be used for bootstrap aggregation. So, by bagging works, because it reduces the variance of the predictions,

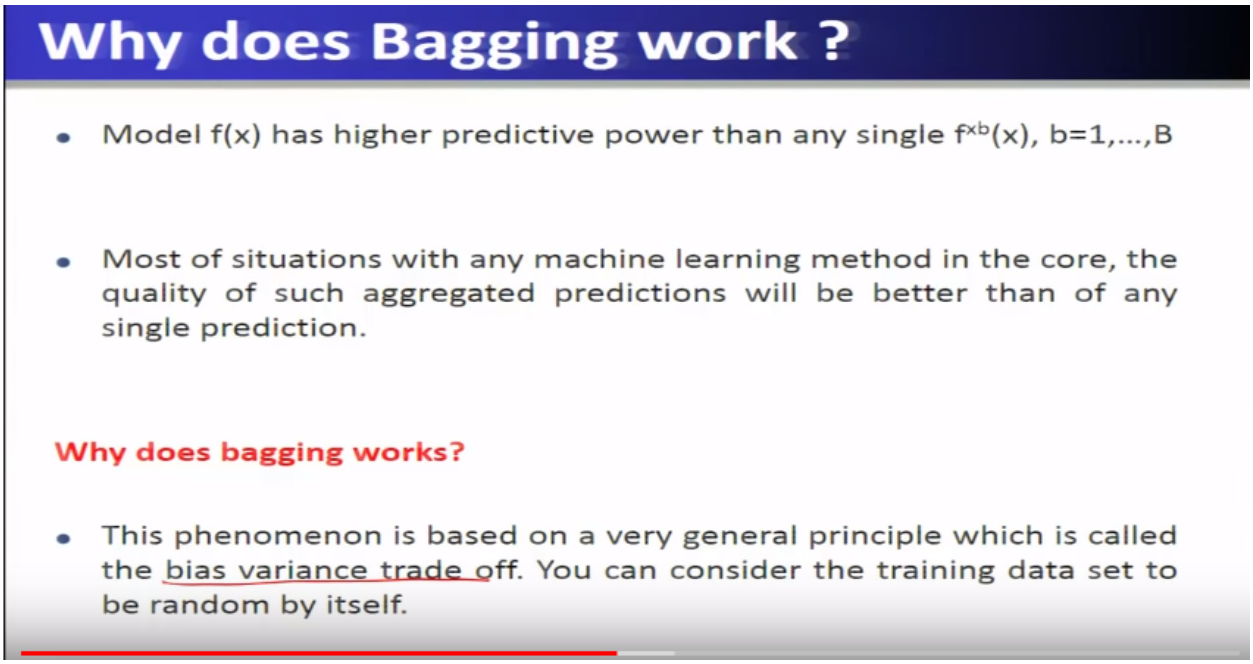
Refer Slide Time: (14:20)



let us understand the bagging algorithm in more details. So, input to the bagging algorithm is nothing but a training set that is Z, which comprises of x1, y1 and so on. So, where X is the features or in a data set and Y is basically the, the labels which are given as part of the target. So, the input as the training set is given and let us assume that another input which is also required in the bagging algorithm is the number of iteration, let us say it is B and also we require one machine learning algorithm method, let us say it is M. So, now the bagging says that for, for the iterations from 1 to B and in each in the step number 2, it says that it draws a bootstrap sample Z star, Z star B of size M, from the training data. So, this particular Z star is a bootstrap sample, that we have seen, how to generate in the previous slide. So, once we draw a bootstrap sample that is called Z star B, the B is the B^{th} iteration, we have generated this particular training data to be used in this iteration, B^{th} iteration that is Z star B. Now, we will apply the machine learning method M on this data set, that is Z star B, and obtain a model, let us call it as F, X star B that is at the B^{th} iteration, the model will be F, X star B and finally after carrying out the iterations B, we will generate the that is the ensemble collection of different models, how many models, B models, capital B one for each. So, the first iteration will have F star one F star 2 and so on up to F star B. So, it will be ensemble of B different models, we have generated in this iteration, then this particular model will be used in the prediction with ensemble. So, let us see that if the problem is the regression problem, then the effects that is the prediction, that is the predicted model, the model for, the biking model called F, X means nothing but it's an average of all the prediction menus with the ensemble and if it is a

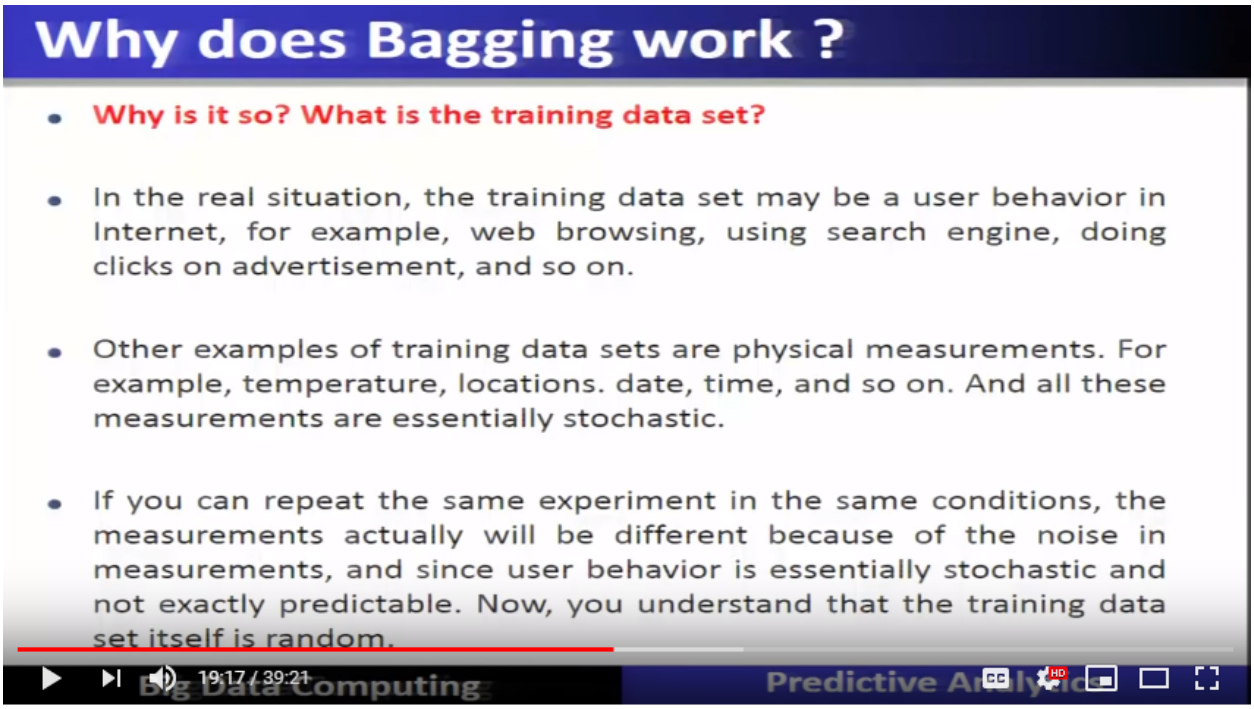
classification, then we will go for the majority of all the predictions and which is shown as every model will give out one prediction and the majority of that will be the other prediction with the ensemble, to understand this, let us consider an example. So, let us say that this is the decision tree and every DCL, let us say decision tree number 1, 2, 3 and they are represented as F star 1, F star 2 and F star 3, now these different trees let us say will give out result in the form of let us say fruits, let us say this is orange and this particular tree will give a fruit which is called mango and third tree also will give out the result the prediction result, let us say that it is again orange. Now, we have to do a voting, majority voting. So, in that majority voting, we will see that this orange is appearing twice, hence the, the prediction output will be in the form of orange.

Refer Slide Time: (18:37)



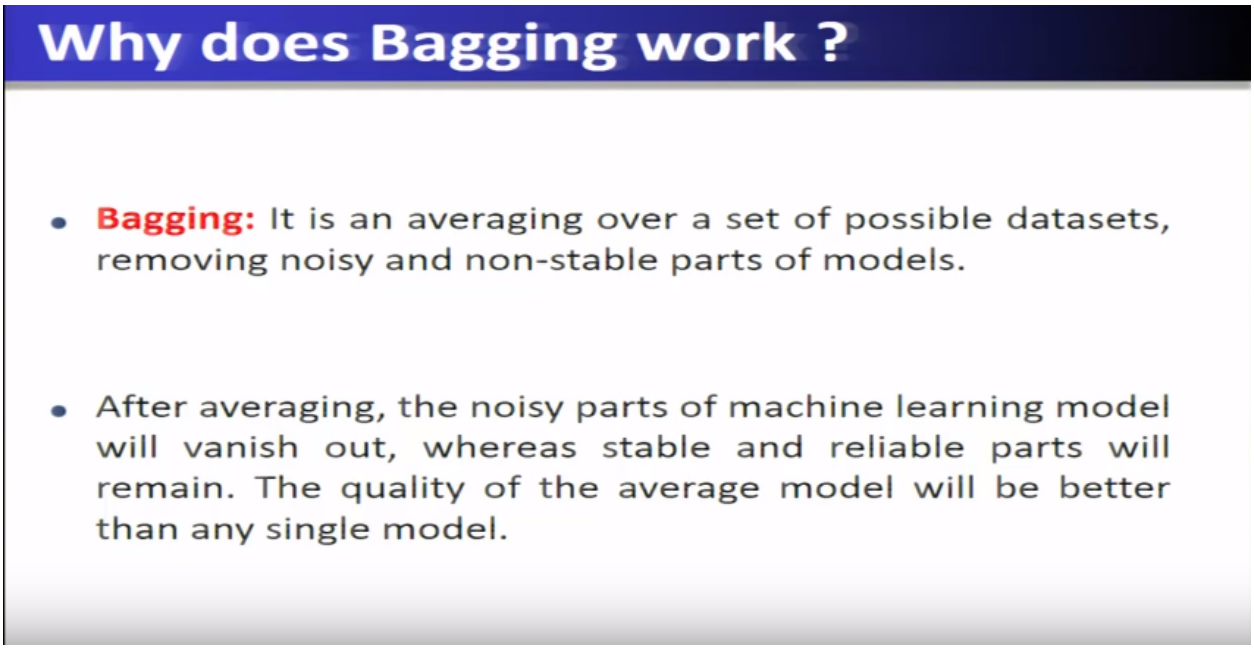
Now, the question is why the bagging works? So, the model F, X has the higher predictive power than any single decision tree or any single model and so most of the situations with any machine learning method is the core, the quality of such aggregated prediction will be much better than any single prediction and why does the bagging works, this phenomena is based on the very general principle, what is called as a bias variance trade off. So, you can consider the training data set to be the random by itself.

Refer Slide Time: (19:17)



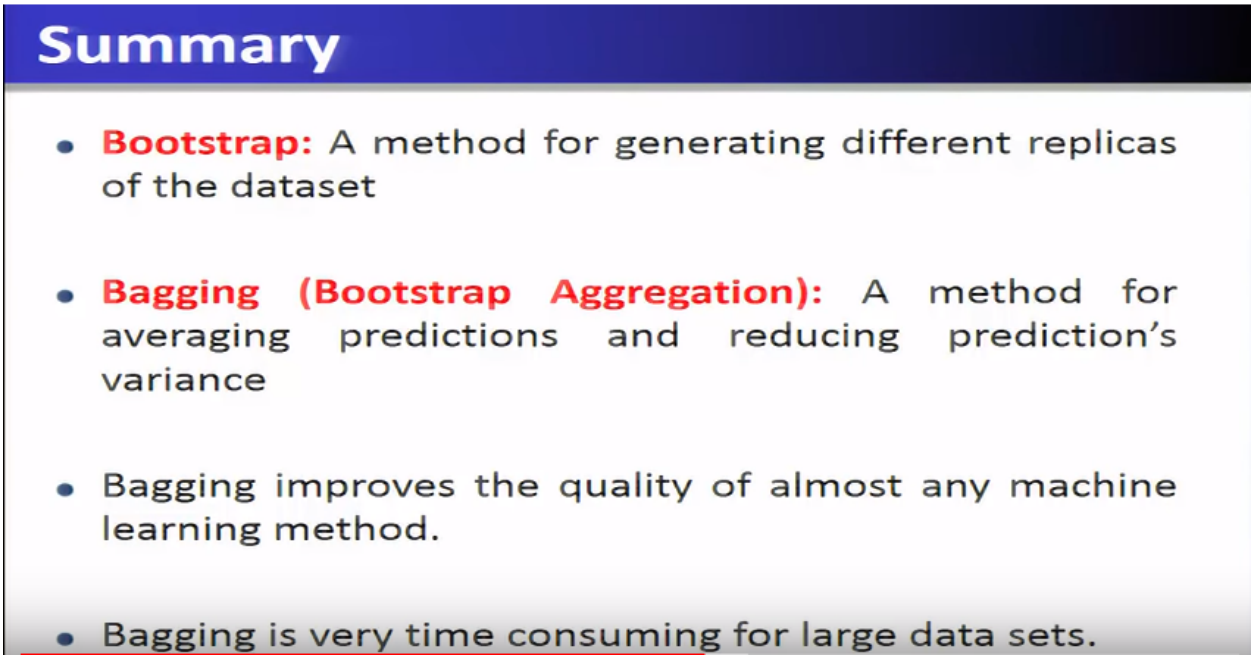
So, in the real situation the training data set maybe the user behavior on the internet for example, while web browsing and which web pages the user clicks using search engine, while doing the, while clicking on the advertisement and so on. Similarly, other examples of the training set data are the physical measurements for example, temperature, location, date, time, and so on. All these measurements are also essentially stochastic which are not same. So, if you can repeat the same experiment, in the same condition, the measurement actually will be different, because of the noise in the measurement and since the user behavior essentially are stochastic. So, not exactly predictable and so you can understand this particular training data itself is random. So, because of this particular nature of the randomness in the data or a stochastic nature.

Refer Slide Time: (20:17)



So, the bagging is also works. So, the bagging is an averaging over the set of possible data sets and removing noisy and non-stable part of the models will be achieve during bagging. So, after everything the noisy part of the machine learning model will vanish out whereby the stable and reliable part will remain in an effect. So, the quality of the average model will be much better than a single model. So, by this way, we have understood that you know, what is the bagging and by averaging, the bagging, due to the stochastic nature of the, the data set generation will allow the bagging method to work and but the important part of the bagging is that so it will be averaging of different example predictions. So, this averaging is more accurate, because the noisy part of that machine learning model will vanish out in this averaging process, whereby the stable and reliable part will remain. So, the quality of average model will be much better than any single model, here in this case of the bagging scenario,

Refer Slide Time: (21:29)

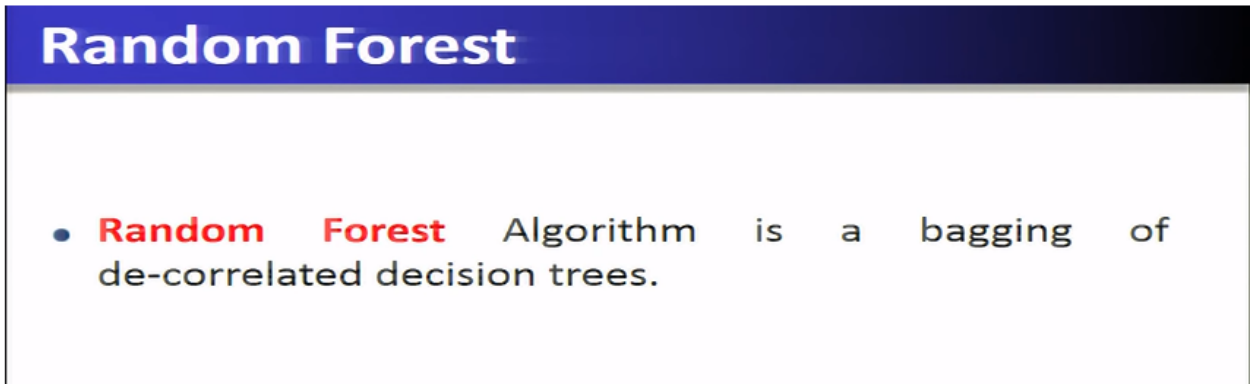


now in short, let us summarize these two basic concepts which are going to be used in the random forest algorithm, which we are going to discuss next. So, the bootstrap is a method for generating different replicas in the data set and bagging is an bootstrap averaging aggregation. So, this method is based on averaging predictions and reducing the predictions variance. So, bagging improves the quality of almost any machine learning method, quality of prediction based on any machine learning method. So, bagging is very time consuming for the large data set,

Refer Slide Time: (22:10)

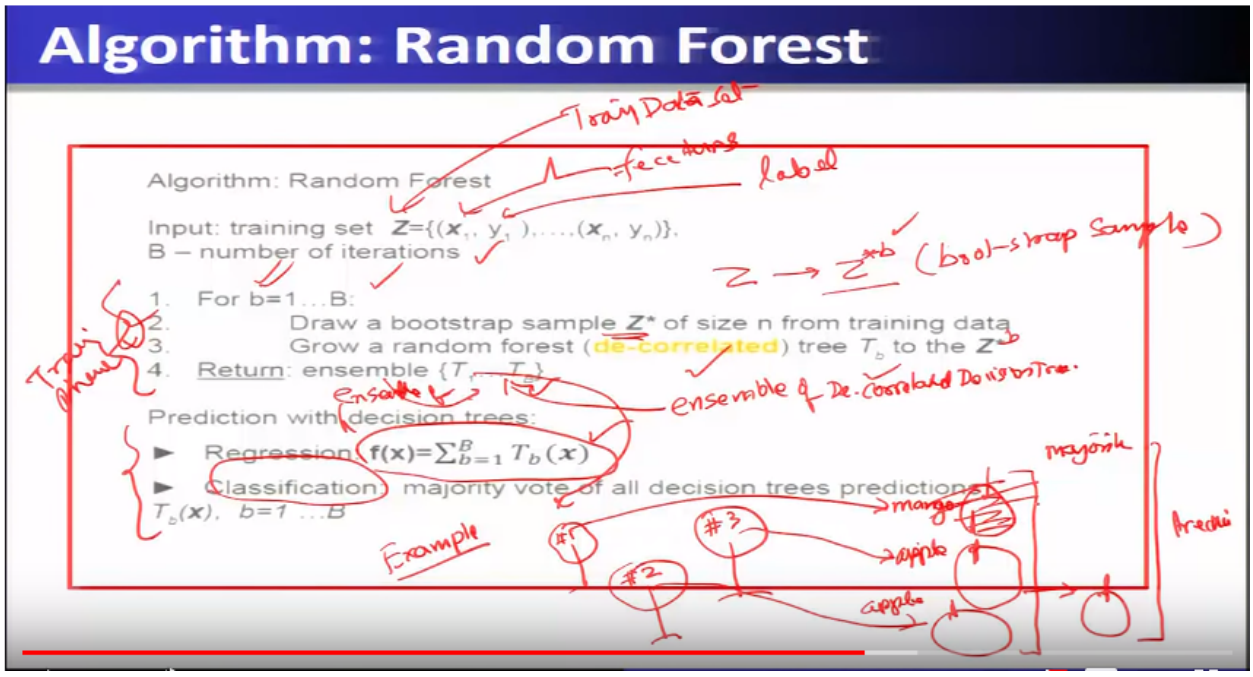
now let us see the random forest algorithm which uses the bagging and boosting methods.

Refer Slide Time: (22:20)



So, random forest algorithm is bagging of the decorrelated decision trees. So, again I am repeating, random forest algorithm is a, is a bagging of decorrelated decision trees,

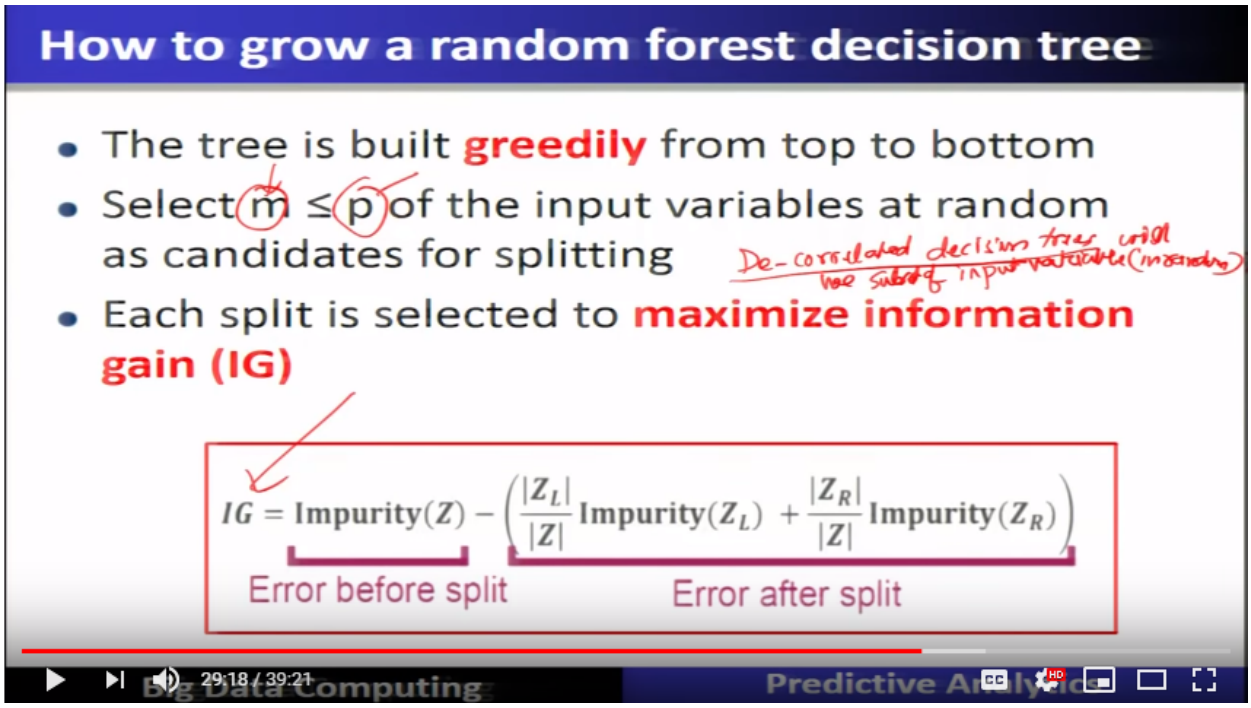
Refer Slide Time: (22:36)



what you mean by this let us understand the random forest using the algorithm of random forests. So, the algorithm of a random forest requires the input in the form of the training data set again X, I, it represents the features of a data set and why I represents the label which is given in the training data set. So, Z is represented, Z is a training data set as an input to the random forest algorithm and total number of iterations, let us call it as capital B, now we will see, how what steps are followed in each iterations for small B is from 1 to capital B, total number of iteration, let us see in each iteration that is, let us say, step number 2 it will draw a bootstrap sample which is called Z star of the size n, from the training data set. So, training data set Z will produce Z star, let us say that Z star B, is the bootstrap sample for the iteration

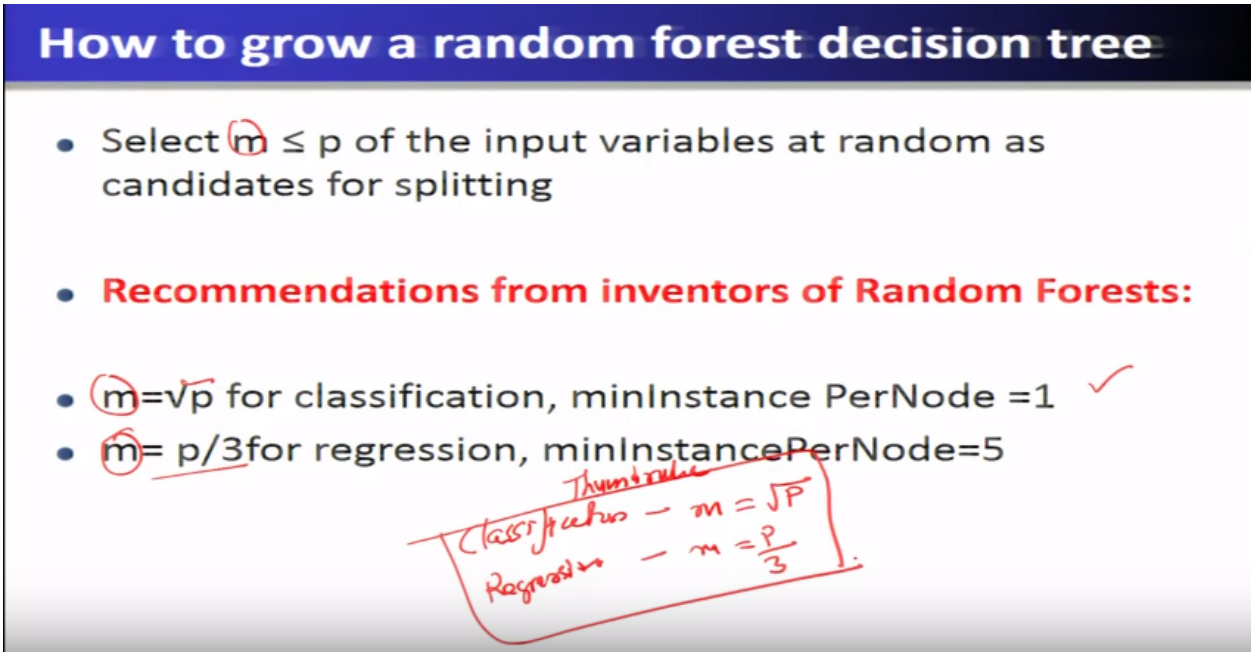
number B, now then we will apply then, we will grow the random forest, that is nothing but decorrelated decision tree and let us call it as T_b. On this particular data set Z star B, and finally we will return this particular iteration will be repeated at 1 to B, different iterations. So, that means we will generate a be different decorrelated decision trees and call it as T1, T2 and so on, up to capital B, different trees and this is ensemble of the different decorrelated decision tree, now using this particular training phase model which is called an ensemble of different decorrelated decision trees, that will be the ensemble of models which will be generated by the training phase, now this model will be used for the prediction with these ensemble tree decision trees, ensemble of decision trees. So, if it is regression then, we will now as far as the prediction is concerned which is nothing but an average of the prediction of all the models, which will be given as the outputs and if it is a classification, than the majority vote of all the decision trees will be prediction, let us see again by the an example, to understand what do you mean by the, so classification will require the categorical output from every ensemble tree, let us say that this particular ensemble or crease is represented by, let us say a tree number one, a tree number two and let us say it is having a tree number three, these trees will generate an output, let us say this tree will generate an mango as an output, this tree will also generate, let us say apple as an output and tree number two will again generate an apple as an output. So, now if we take at the majority then, majority comes as an apple. So, this will be the prediction. So, prediction will suppress the noise part therefore these random forest trees are much more accurate as far as the prediction compared to the single decision trees,

Refer Slide Time: (27:25)



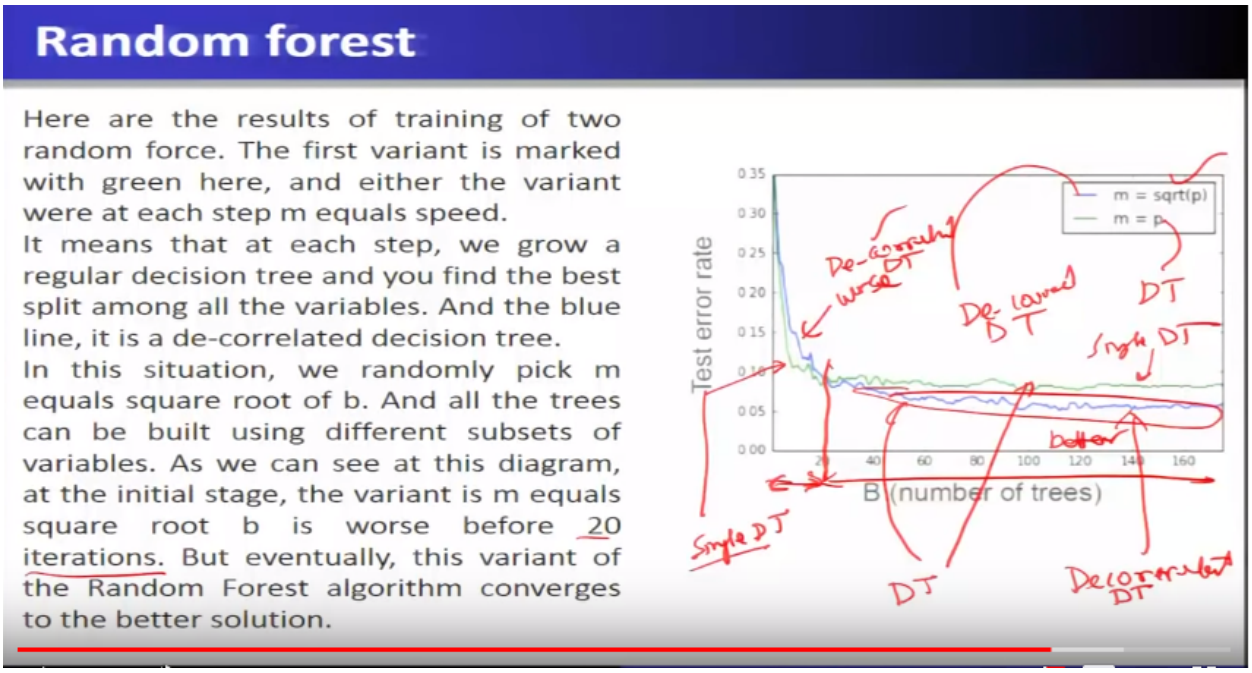
how to grow a random forest decision tree's? So, the trees is built greedily from top to the bottom as we have seen in the in the decision trees, but the difference here is that it will select M out of P different features. So, select M out of P input variables at random as the candidates for the splitting. So, in decision tree all variables are used as the candidate for splitting but here this is called decorrelated decision trees will use the subset of the input variables in random order. So, that is why it is called decorrelated decision tree, because not all input variables are used, but subset of the input variables are used and they are used for the tree induction, that is the they will be used to decide the candidates for the splitting, now each split is selected to maximize the information gained. So, information gain as we know that is dependent up on the impurity of Z, and that is the errors before the split and then errors after the split and which is shown by this particular equation.

Refer Slide Time: (29:20)



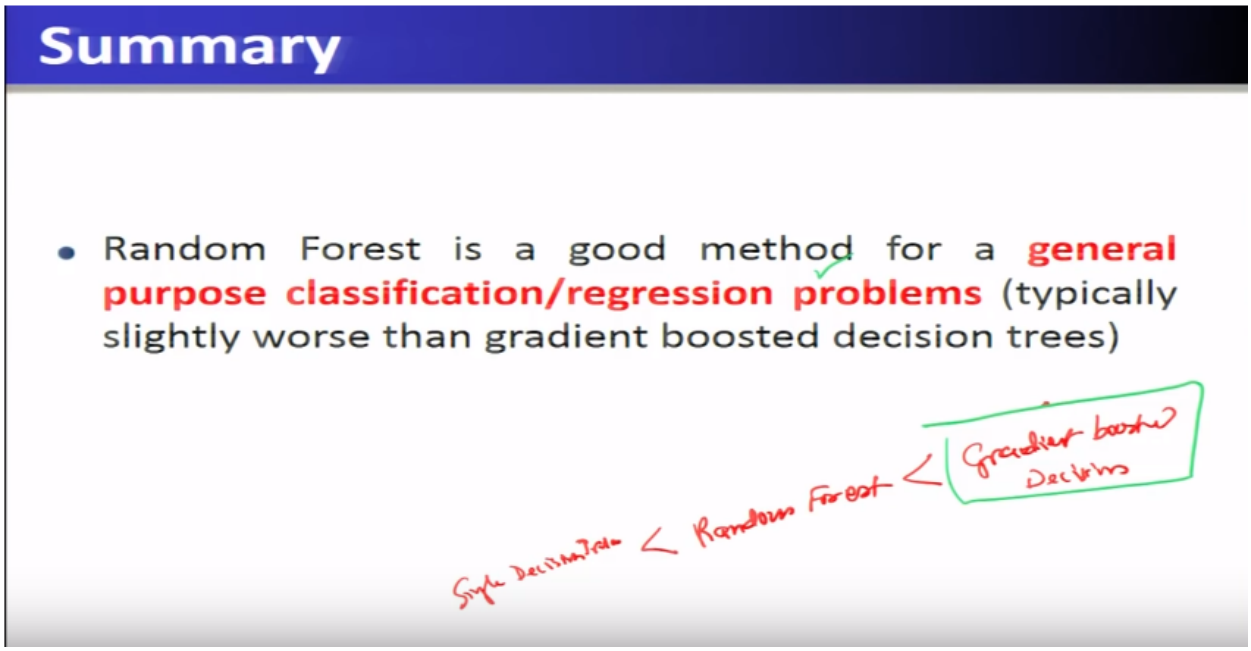
Now, how to grow the random forest decision trees. So, select M out of P input variables at random, as the candidate for the splitting. So, here the recommendations from inventors of random forests are summarized here, so for classification, the value of M has to be chosen as root of P, for the classification problem and the minimum instance will be per node is 1 and M will be P times, P divided by P by 3, for regression problems. So, for classification, the rule of thumb is, thumb rule says that the value of M should be root of P and for regression, the value of M is equal to P by 3.

Refer Slide Time: (30:30)



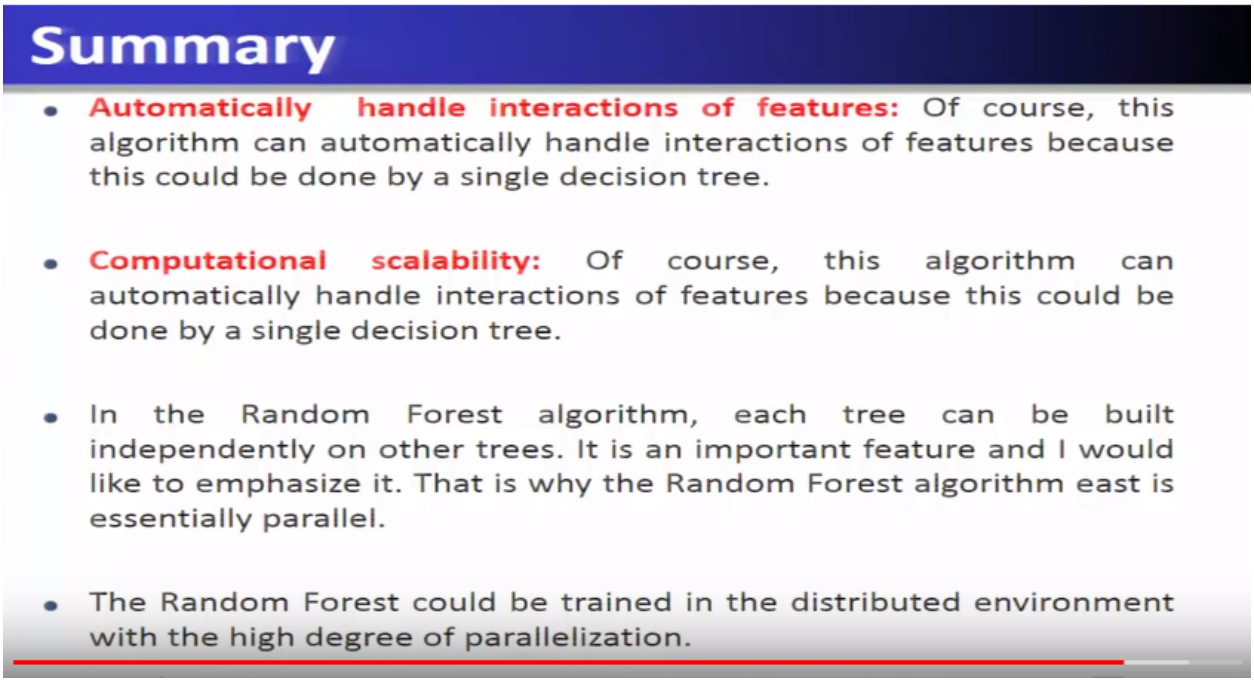
So, random forest here the results of the training of two random forest is shown, the first variant is marked with the, with the green one and, and either the variant where each step M equals, this particular case and it means that at each step, we grow the regular decision trees and you find the best split among all the variables and the blue line, that is, it is the correlated here it is the decision tree, and the blue one is decorrelated decision tree. So, this is a decorrelated decision tree and this is the decision tree which is shown as the green one. So, this means that at each step, we grow the regular decision tree and you find the best split among all the variables and the blue line, it is decorrelated decision tree. In this situation, we randomly pick M equals square root of B, that is shown over here, square root and all the trees can be built using different subset variables and we can see in this particular diagram that initial stage the variant is, is M equal square root of B is worse before 20 iterations. So, before 20 iterations, if we do this kind of analysis, up to this point here, this decorrelated is worse, is going worse, decorrelated decision tree is performing worse compared to the single decision tree, but for the value which is more than 20, before the 20 iterations that means, if the number of trees are less than 20, then the, the decorrelated decision tree, will perform not that good compared to the single decision tree, but as the number of trees grows, that is nothing but the number of iterations. So, if the number of trees are more than 20 then, we see that this your decorrelated decision tree is performing much better, start performing better compared to the single decision tree, but equivalently this variant of random forest algorithm converges to a better solution, that means after beyond 20 iterations or beyond 20 number of decision trees, this particular random forest algorithm converges to a better solution.

Refer Slide Time: (33:34)



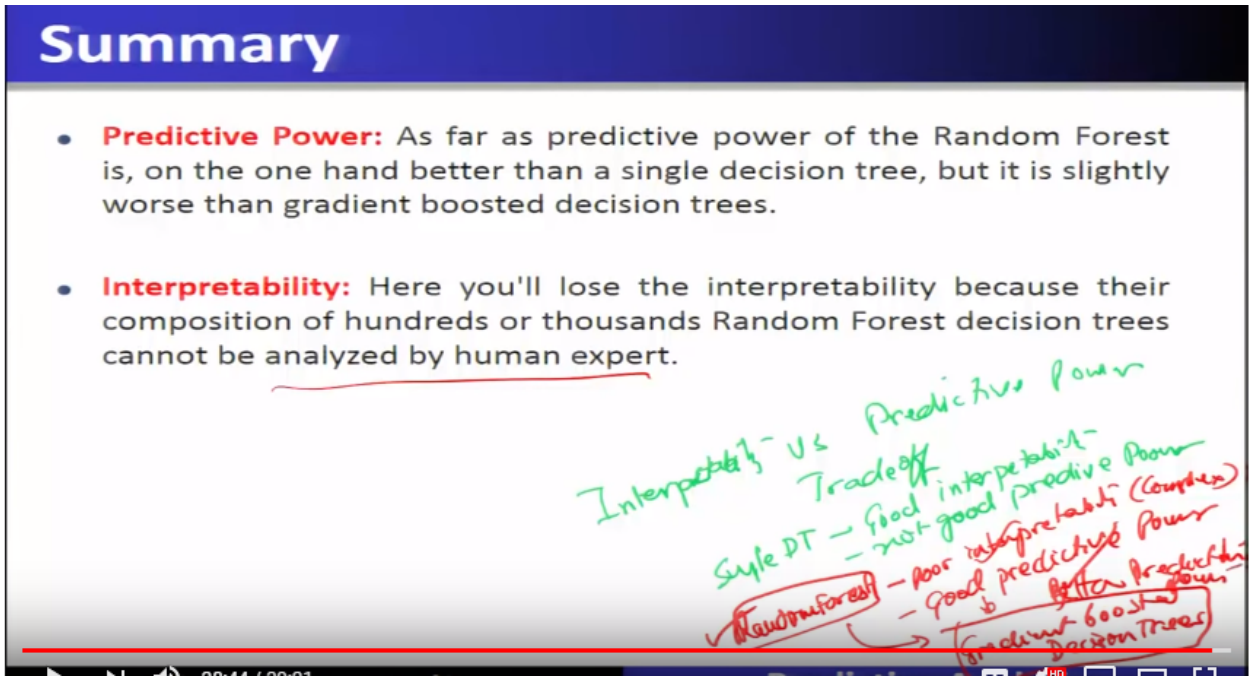
So, random forest is a good method for general purpose classification regression problem, typically, slightly worse than gradient boosted decision tree, we will see that both the cases. So, that means, we have seen that random forest is performing better, is better than the single decision tree, but the random forest is slightly worse than the gradient boosted decision tree. So, that means gradient boosted decision tree is going to be the best approach for a general purpose regression problems and for prediction problems that we will see in the next slide.

Refer Slide Time: (34:32)



So, summary that is, the automatically handles the interaction of the features in the decision tree and of course, this algorithm can automatically handle the interaction of the features, because this could be done by the single decision tree. So, also that particular part or is also covered in the random forest, now computational scalability is also there in the random forest, of course, this algorithm can automatically handle the fraction of features because this could be done by the single decision tree. So, therefore in random forests also, it ensures computationally scalability in the random forest algorithm, each tree can be split independently on the other trees, it is an important feature and I would like to emphasize it, that is why a random forest algorithm is easiest and essentially it is parallel. So, random forests could be trained in a distributed environment with high degree of parallelization.

Refer Slide Time: (35:37)



So, so another aspect of random forest is the predictive power as the predictive power of random forest is on one hand better than a single decision trees, but it is slightly worse than gradient boosted decision trees. Another part is called interpretability, here the interpretability is not that good, that means you lose the interpretability because of their composition of hundreds and thousands of, and simple decision trees involved in the random forest and therefore it cannot be analyzed in the interpretation compared to the single decision tree problem, which are more interpreted, interpretable, but less predictive power, and so therefore interpretability, what says predictive power is having a trade out. So, as far as the single decision tree will have the good interpretability, but not that good predictive power, whereas the random forest has poor interpretability that means due to the complex nature of the tree and sample, and so it is not possible to be understood by the or analyzed by the human expert, whereas the prediction is very good, good predictive power is there in the random forest, therefore for good prediction, random forest is preferred compared to the single decision tree, but there is another better scheme as far as the predictive power is concerned, because random forest is already lost it's good interpretability aspect. So, let us see, how better, how the best, we can do the prediction? So, another improvement of random forest is called a gradient boosted decision tree, which is having much better predictive power compared to the decision trees.