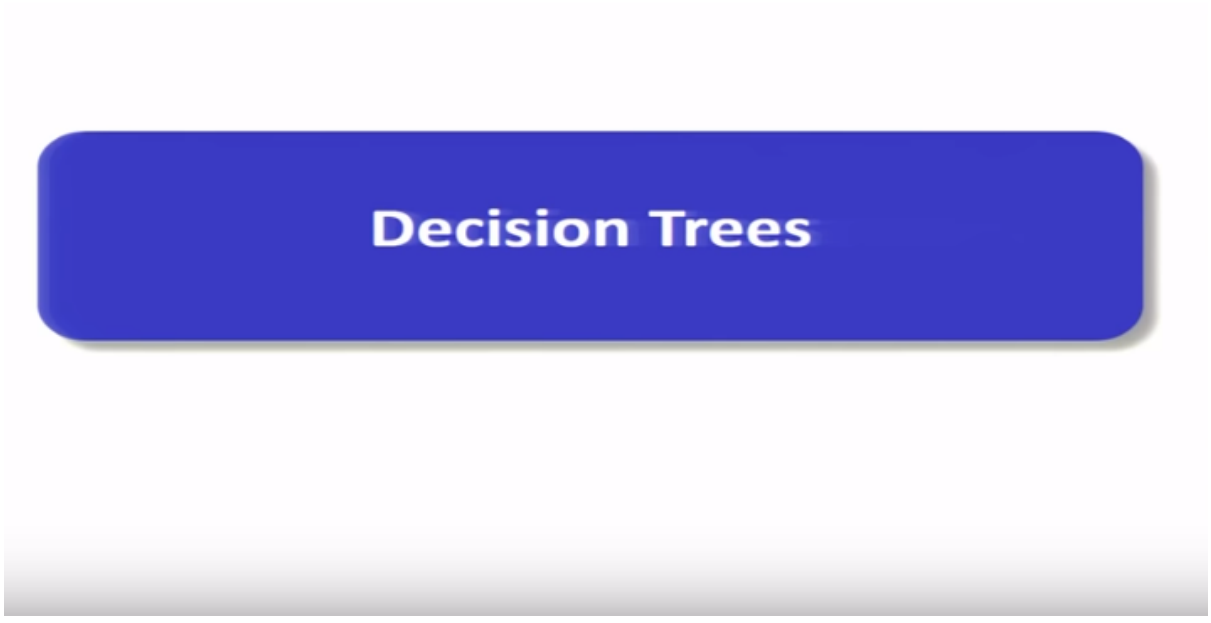
Lecture - 27
Decision Trees for Big Data Analytics

Decision trees for big data analytics.

Refer Slide Time (0:18)

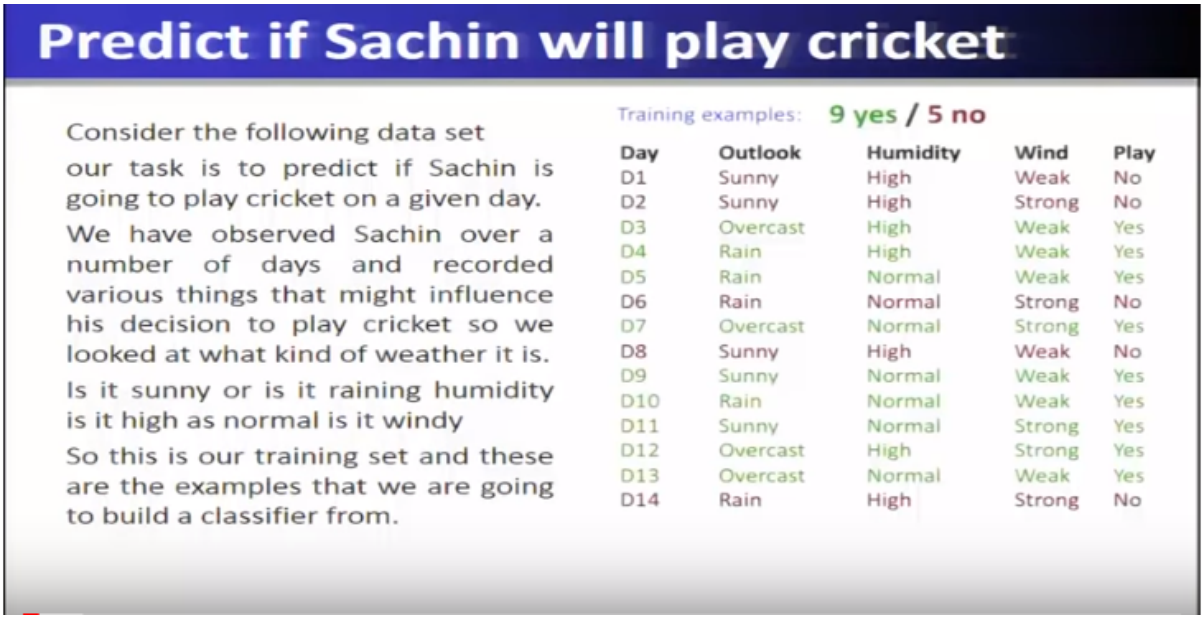
Pre-phase. Content of this lecture in this lecture we will discuss decision trees for big data analytics. We will also cover a case study of a medical application using decision trees in SPARC ML.

Refer Slide Time (0:37)

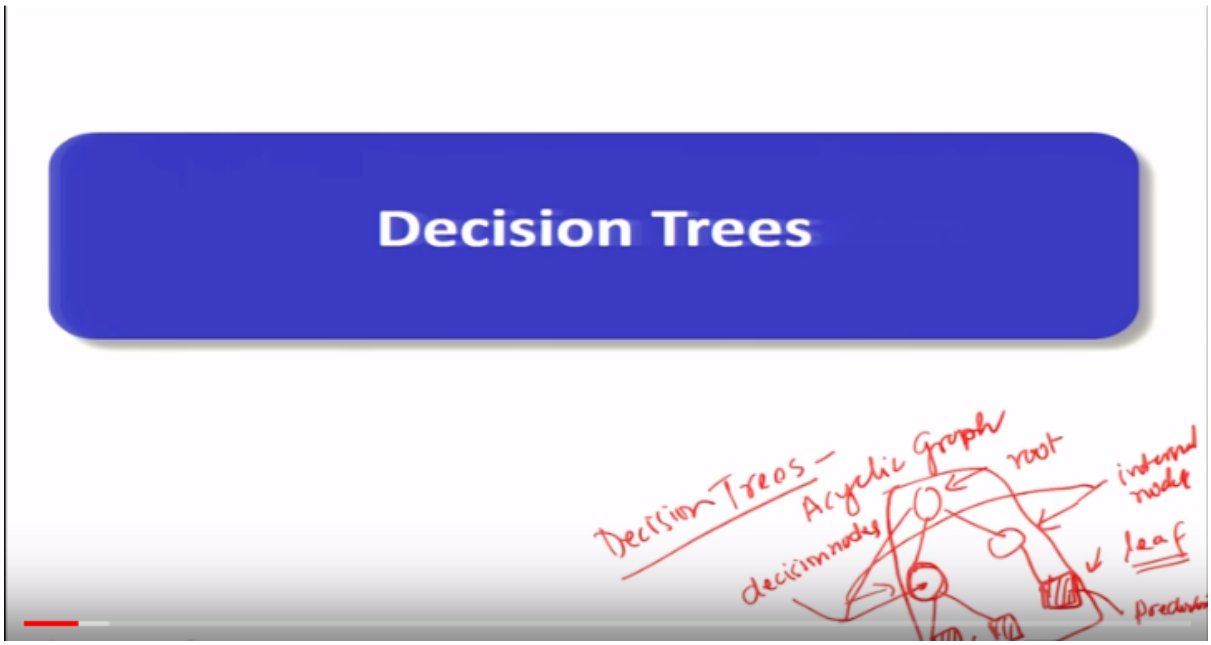


Before, that let us understand about the basics of decision trees.

Refer Slide Time (0:42)



Refer Slide Time (0:46)



So, decision tree is a tree, which is nothing but an acyclic graph, which has the root node and other nodes. This is called the root node of a tree and these are all called internal nodes of a tree and this is called the leaf. So, the square boxes represent the node, which is called a leaf node in the decision tree. The predictions are there at the leaf nodes, and the internal nodes and the root nodes they are called the decision nodes. Now, given a data set we are going to construct this particular decision tree, using the decision tree algorithm, and traversing the tree we can predict for a given query. So, whenever a new data set, we get we will be able to handle this particular data set for doing the predictions. So, details we are going to see in this discussion and, then how this decision tree is going to be useful for doing the big data analytics, also we will cover through an example.

Refer Slide Time (2:39)

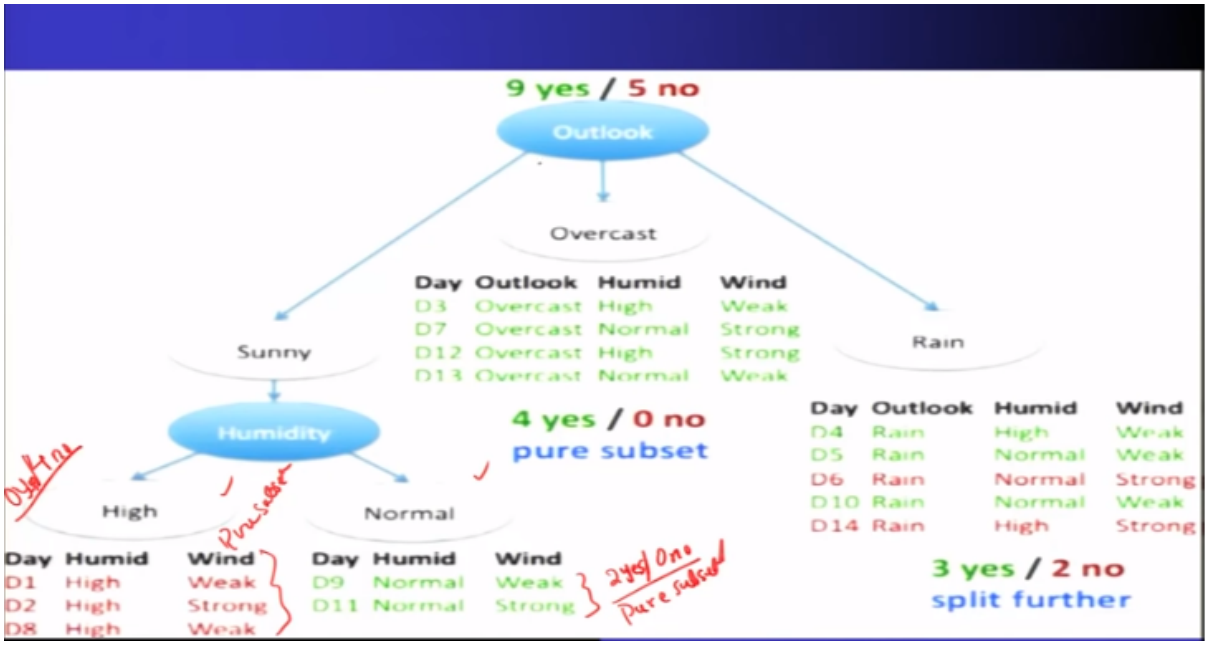
Let us see, simple data set and let us see what kind of predictions this decision tree will will be able to make. So, to understand in a very simple manner. Let us consider this data set add a query. So, the queries, that we want to predict, if searching will play the cricket or not, to answer this particular query, that means to get the prediction whether searching will play the cricket or not. We are going to be have observed the details of Sachin playing and also the weather conditions over 15 days. So, day 1 to day 50 14 or 14days. Wherein we have considered, that what is the outlook of the weather if it is sunny overcast raining and the humidity of every day, that is whether it is high normal or the wind speed whether it is weak or a strong and we have also seen, that in these conditions whether, that player such in highest played or not. So, on day one it has not played he has not played day three he has played well. The weather was outlook was overcast humidity was high and wind was weak and so on. So, we have observed the 14 days of these different parameters and also, we have seen whether the player Sachin has played under these conditions or not. Now, using this particular data set, can we predict whether Sachin will play to the days, which are a new day or a next day or not so decision tree. Let us see, how it is going to be useful? So, consider this data set and our task is to predict if the Sachin is going to play the cricket on a given day. So, we have observed the playing the game playing of Sachin our number of days. In this example we have monitored, or we have observed 14 days and recorded various conditions that might influence his decision to play the cricket. So, under under different weather conditions. So, is it sunny, or is it raining, or it is having humidity, and then in that case or it is having a normal windy temperature what conditions are required and necessary and to predict whether certain will play or not? Even for this is a small or a simple data set this becomes quite complex. So, let us see how we are going to simplify using decision tree for this particular purpose. Now, one thing is, that this particular data set is basically observation of the last 14 days. So,

that becomes the training set and using these examples can be predict this particular decision of a search in whether he will play the game or not? So, that means the query will be that on day 15th if outlook is, raining humidity is high and wind is weak, whether such an will play or not So, from this data set also it seems, that it's not easy to get or understand this entire data set and come out with the decision to predict or not? But, let us understand this particular data set, because decision tree is the technique, which will understand these different parts of the data set or the behaviour of that player Sachin it will understand and based on, that it will be just the predictions. So, on day 15 if it is raining and the humidity is high and the wind speed is not windy slow or a week wind is capacity, then weather Sachin is going to play or not, and by looking to this particular data and this kind it will become very hard to decide. The right kind of predictions, whether certain is playing in other days on one of these parameters and how this kind of condition, whether he will be playing or not? It's very difficult to predict.

Refer Slide Time (7:44)

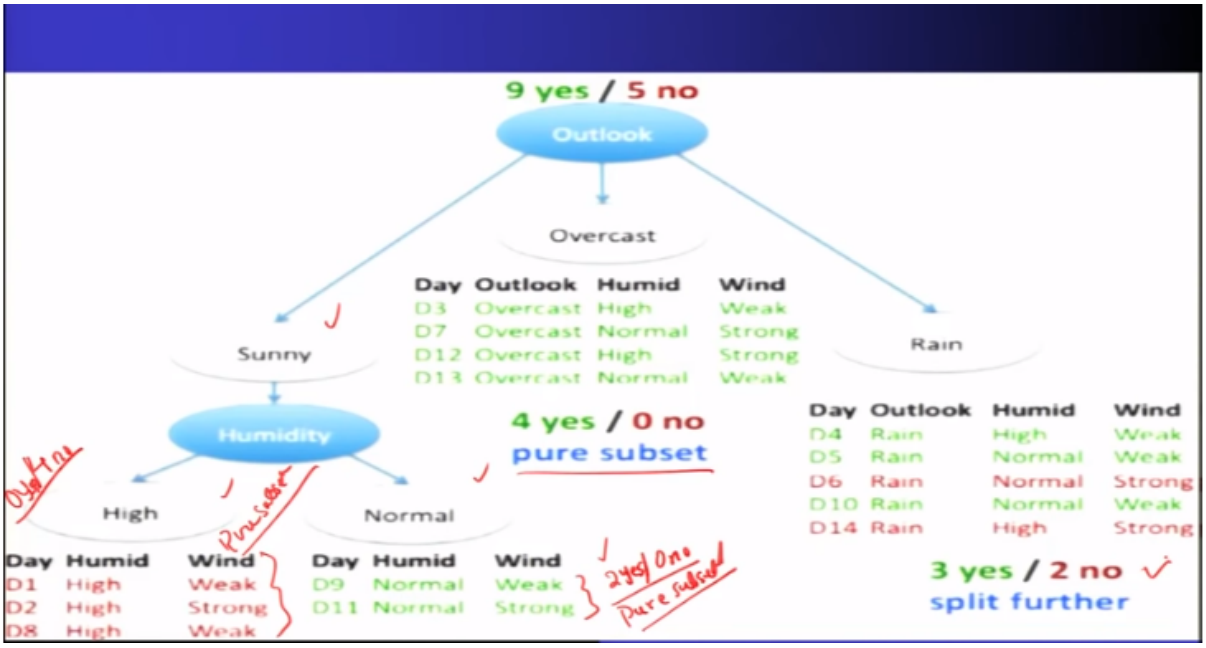
Let us see how in decision tree we are going to do this kind of predictions? So, as we have seen it is very hard to guess, whether Sachin will play on day 15 and but for this we have to understand, when such in play? What are the condition when certain will be when Sachin is playing? So, there are 19 different samples when Sachin has played and under different conditions 9 and these 9 different conditions, when Sachin has played is required as, the training to understand when he has played and where there are 5 different condition, when he has not played in those weather conditions. Now, we will to understand this behaviour of a player Sachin out of this data set, which is an observation of last 14 days. We can do this by splitting into the subset, into subsets of different classes and then we can extract this information and the the decision information we can then basically use this information for predicting a new data set.

Refer Slide Time (9:07)



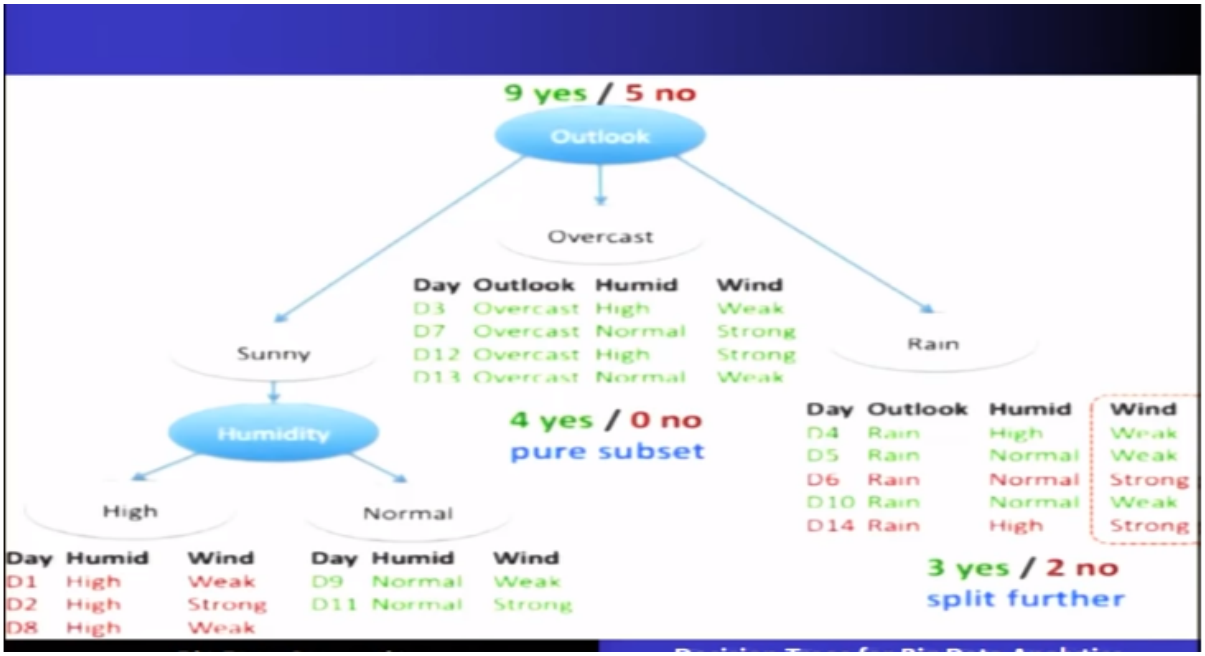
So, let us understand more details of this entire data set, which is given. So, if we classify this data set based on the outlook parameter, then there are 3 different outlook condition. One is sunny, than overcast and raining, and these subsets which we are going to get in these different conditions. So, we basically have obtained three different subsets. Now, out of these three different subsets we see, that in when the outlook is overcast, we can see, that here. There are 4 different entries, or 4 different days are there, when this becomes yes, and there is no entry which is having 0 no's. So, hence, this kind of subset is called a pure subset, which has all yes. When the outlook sunny, then we can see this particular subset, which has three no's and two yes. So, this kind of subset is called impure subset. That means it is mixed, whether to are saying yes when the outlook is sunny, then Sachin has played in two of these conditions, when the humid is normal and the event is a weak or strong and in three different cases, he has in played. So, this kind of subset is called impure subset. Similarly, when the outlook is rain, then there are three different cases, when Sachin has played and two different cases, when Sachin has not played. So, hence, this becomes an impure subset. Now, we can further split these impure subsets, and pure subsets need not to be splitted further.

Refer Slide Time (11:28)



So, now, we can see, that we can split so, that now, we have splitted this outlook sunny, based on the humidity. When the humidity is high or normal. So, what we can see here, that when the humidity is high, all three cases the Sachin has not played. So, four no's and zero yes. Hence, this is called a pure subset. Whereas when the humidity is normal, then there are two different in this subset, that means two yes is there, and there is no zero no's. Hence, this is also called a pure subset. So, when it is a pure subset, then decision can be easily made. For example, when the outlook is sunny and the humidity is normal, then Sachin has always played. So, whenever a data of this category comes, we can decide about this. So, here also when under this split humidity. We have obtained both the pure subsets of drop out of this split of sunny. Now, then let us see, the rain when the outlook is rain, then it has basically a mixed or impure subset which can be splitted further.

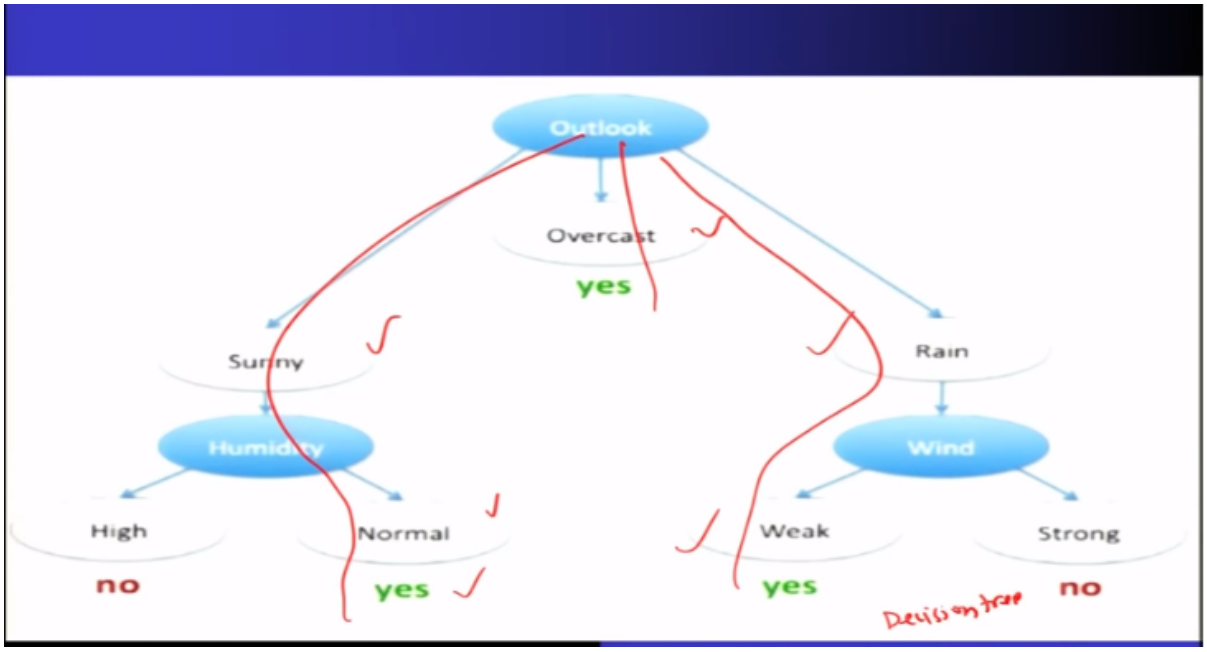
Refer Slide Time (13:00)



Refer Slide time (13:02)

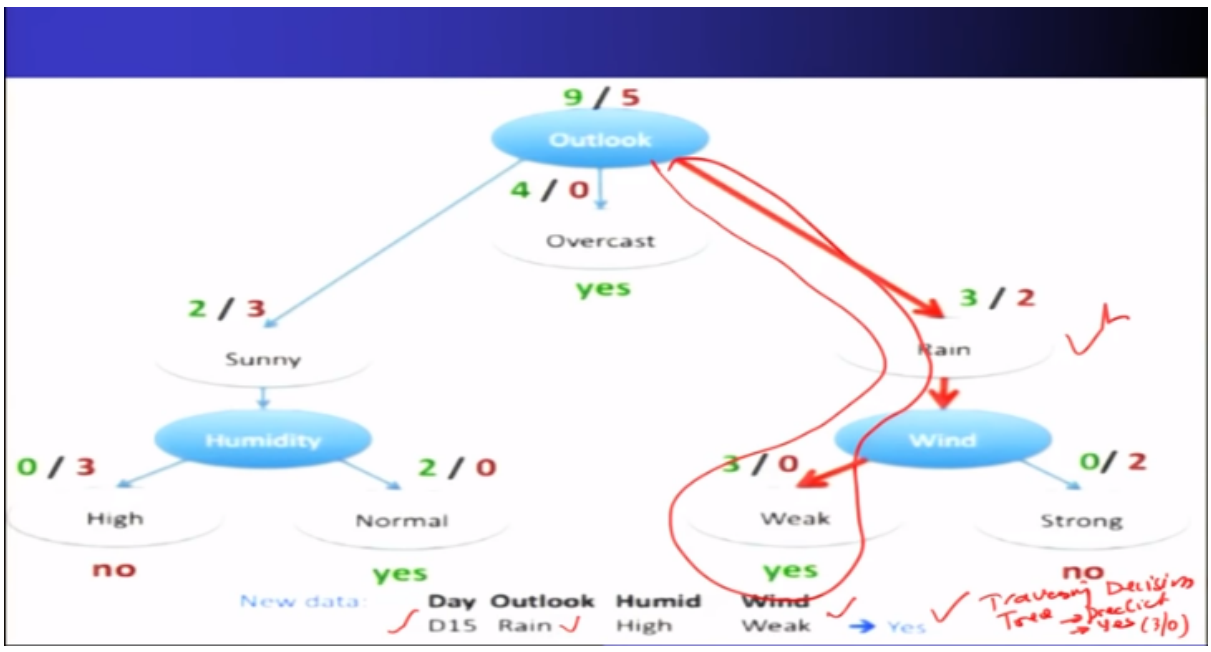
So, if we split further, on this wind what we obtain is, that when the outlook is rain, and the wind is weak, then we have got three different subsets. In this subset the three yes is there. Three times he has played and zero times it he has not played. So, this is also called a pure subset. Now, as far as when the wind is strong, then we have obtained a subset, where the zero yes is there, and two no's are there, and this is also called pure subsets. Now, there is no further division or split required and we have constructed decision tree of this use case.

Refer Slide Time (14:01)



Now, if we summarize this particular decision, that decision tree which is constructed out of, that dataset. We have obtained, that if when the humidity is normal, and when outlook is sunny, then always, that player Sachin hence, played where the outlook is overcast in Sachin has played and, when the wind is weak, and the outlook is rain, then also Sachin has played. So, there are three different cases, when Sachin has played, and which is shown over here. In all other cases Sachin has not played.

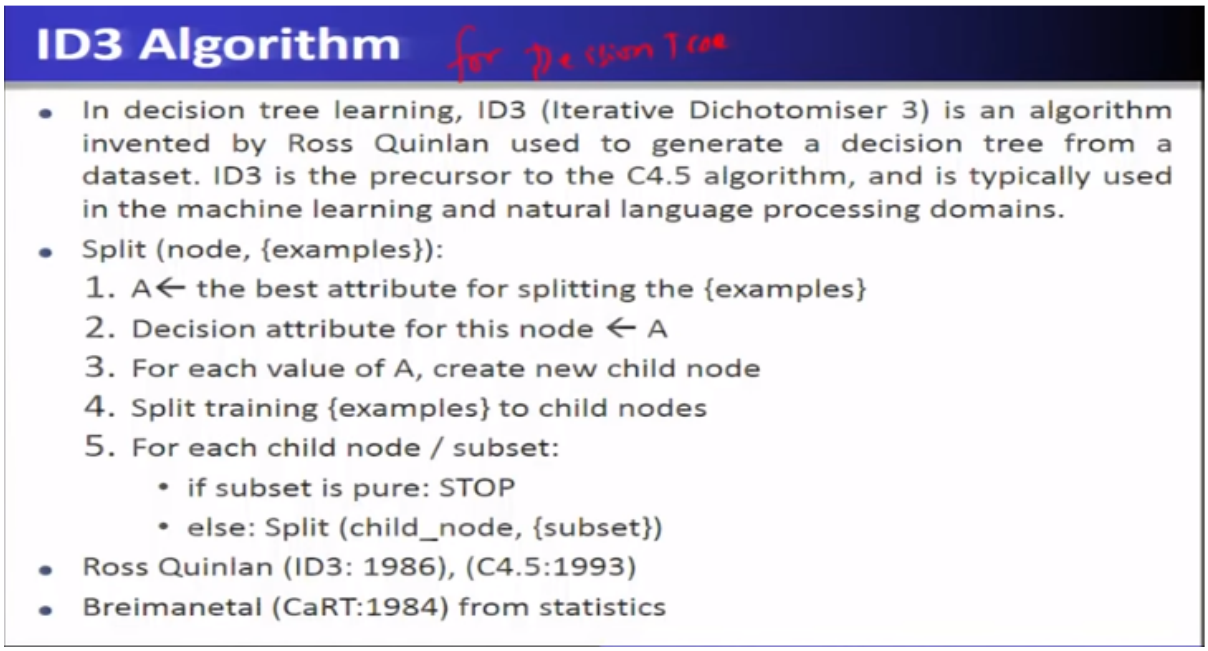
Refer Slide Time (14:56)



Now, not only this particular decision tree has yes and no cases. But, it also has how many cases which are having yes. So, it will add to the confidence on this particular decision using these

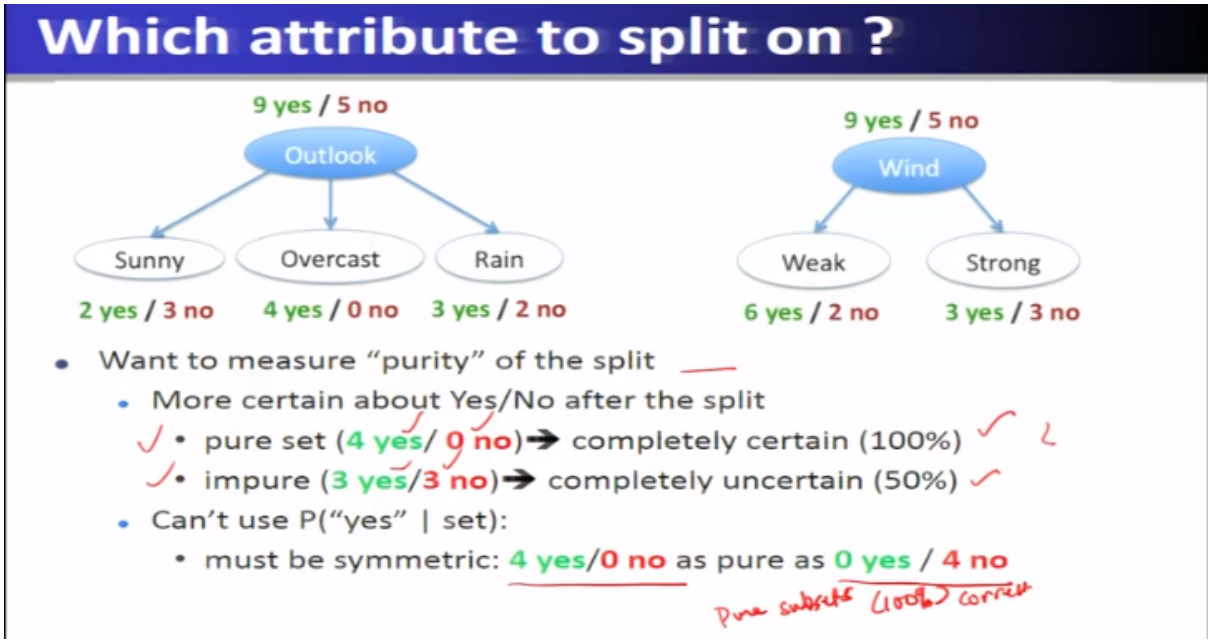
parameters. So, let us see, the query, which was given to us on day 15, that if the outlook is rain, and outlook is rain and humidity is high and wind is weak. When the outlook is rain and and the wind is weak. Obviously, it is going to play in this particular case. So, we are going to say yes. Hence, the decision is yes in this particular case using the traversing of the decision tree. So, by traversing the decision tree, we can predict, that Sachin will play, and the confidence with which we are going to predict is three by zero. So, that means with the high confidence this particular prediction can be made.

Refer Slide Time (16:23)



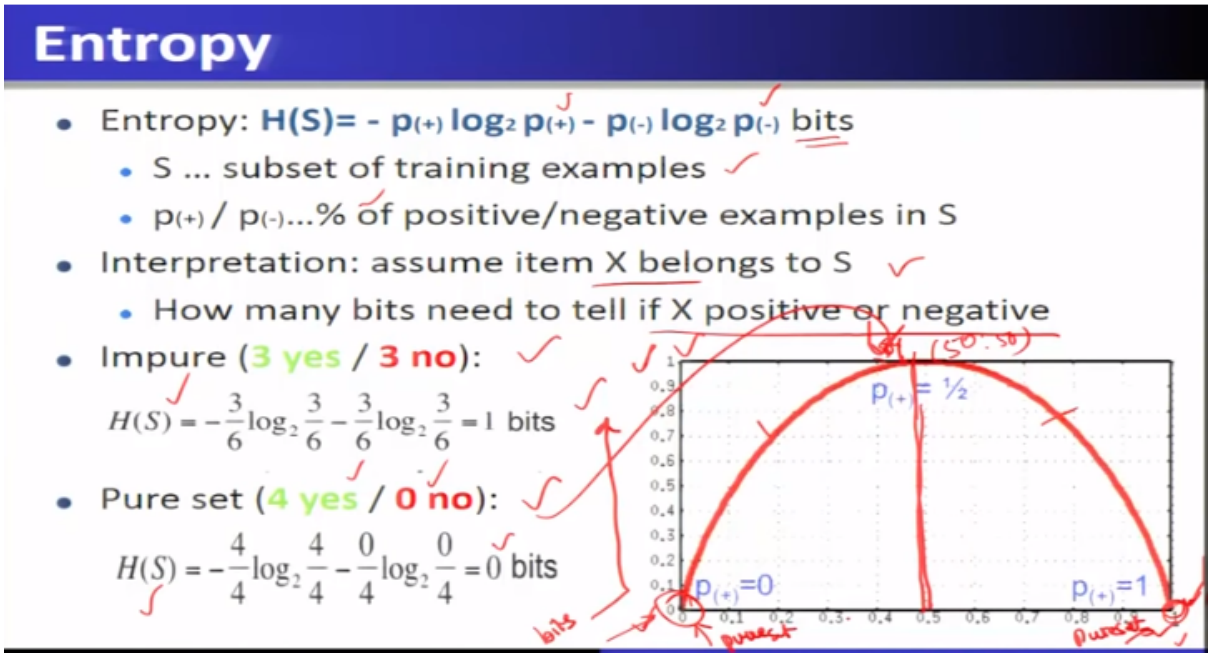
Now, as far as the decision trees are concerned, we are going to discuss one such algorithm which is called ID3algorithm for decision tree. So, this algorithm was invented by Ross Quinlan used to generate the decision tree from the data set ID3 is the precursor to see four point five algorithm and is typically used in the machine learning domain. So, let us see, how the split of a node is done in ID3. Now, let us consider A, which is assigned to the best attribute for the splitting, and the decision attribute for this particular node is assigned from A. For each value of A let us create a new child node. Now, split the training examples to the child node and for each child node or a subset if the subset is pure, then stop splitting further else we have to split the node child into the subsets further. Now, Ross - Lin queuing plan Ross Quinlan which has given this ID3 in 1986 and also C 4.5 in 1993 and Brien man Brein man natal and Briemanatal has given CaRT in 1984 from statistics about this DC entry algorithm.

Refer Slide Time (18:17)



Now, the question is which attribute to split on. So, let us see, that we have seen, that there are three different attributes in that dataset. So, which one we are going to use to split the attributes on, that particular question is going to be, that is the split decision which attribute is split on is a decision which we are going to see how decision tree is going is taking here in this case. Now, here let us see, that it wants to measure, the purity of the split, and this is the way it is going to take the decision, of which attribute to is split on. So, based on the purity of the split the decision is to be evaluated. So, more certain about yes or no after the split. So, for example if we have a pure set where for yes is there, and zero no's are there, then we can say with hundred percentage certainty, that this is a pure and pure set and the decision is very certain. Similarly, if the set is impure, where three yes is there and three no's are there, then the decision which we are going to take is having completely uncertain, that is fifty percentage chance of making the decision correct, is basically is there. So, this way we are going to measure using how many yes is there whether the set is pure or impure and this is going to be the measure to go for the split. Now, so, this particular must be the use of this probability must be the symmetric, that is four yes and zero knows as pure as, zero yes and for no's. So, that means whether it is all yes or whether it is all no's are called pure subsets. Wherein the decision is hundred percentage certain to be correct. Whereas if the set is impure then it is uncertain, that the decision is correct. So, we are going to evaluate this measure of purity to split on.

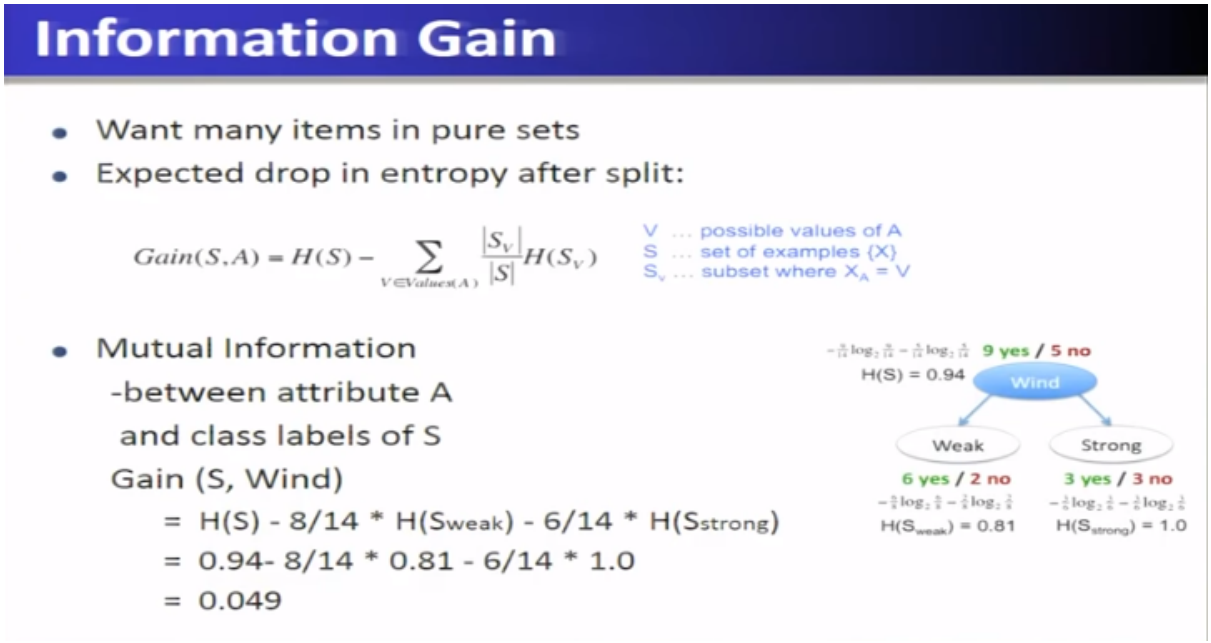
Refer Slide Time (21:05)



Now, this decision tree uses different measures for make the decision whether to split or not? One such way is called entropy, that we are going to discuss. How using entropy, we can decide whether to split further or to not to split? So, entropy is represented by HS here and there are two variables. Let us say P plus and P minus. So, that is minus P plus log to the base 2 P plus minus P minus log to the base P minus I this is nothing but the number of bits. So, if we calculate the entropy of a particular set of training examples, then we will get the bits. So, this the interpolation of the bits we are going to use to infer, whether to split or not how this is all done we are going to see now. Now P plus and P minus percentage of the positive and negative examples in s we are going to evaluate. Now interpolation is let us assume an item X belongs to a set S and we have to ensure or basically estimate how many bits are needed to tell if X is positive or the negative? So, for example if it is impure set of three three yes and three no's, then if we calculate the entropy of this impure set S, then it comes out to be what bit? So, one bit if it is the entropy is one bit, then it is highly uncertain for the decision, whether it is yes or no. So, the number of bits which basically will represent it here it is 1. So, 1 means it is highly uncertain in this particular case. Now, if it is a pure set, let us hear, if it is four years and zero no's and we want to find out the entropy of that set. So, entropy will be zero in this case. So, if the set is pure the entropy will be zero bits. So, zero bits means the decision is very certain, and if it's uncertain, that is highly uncertain, then the bit will be one. Now, and rest of the decisions. are in between zero and one, that is what we have seen. Now, so, this particular diagram shows, that if the number of bits is one if the number of bit is one, here in this case if it is, so, if if it is one then the the decision will be half fifty percent chance, that the decision is going to be correct fifty percent chance. So, it will be a fifty fifty, that means fifty fifty and if all are positive, or all are negative, then the number of bits which is required is zero here in this case. For example, here in the pure set, all are all are yes, and zero are no's. Similarly, another pure set, when all are all are yes, all are no's, and zero yes, then also it is a pure set. In both the cases the number of bits here this represents the number of bits, in both the

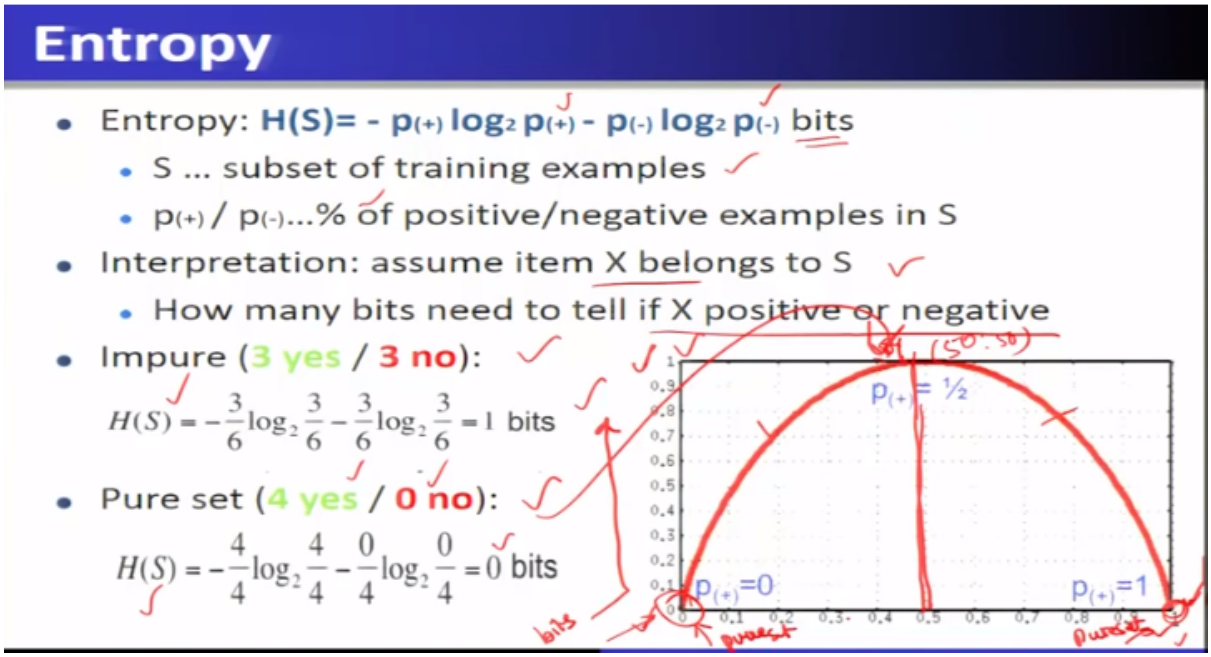
cases the number of bits are zero, hence, this is going to be a pure set, and this is also going to be a pure set, and this is the impure set condition and so, basically either it is pure set, then it will take zero bits or it is highly uncertain then it will take one bit. Whereas all other conditions are lying on either on the ellipse. On the left side or on the right side, where the number of bits is varying between 1 to 0.5.

Refer Slide Time (26:00)

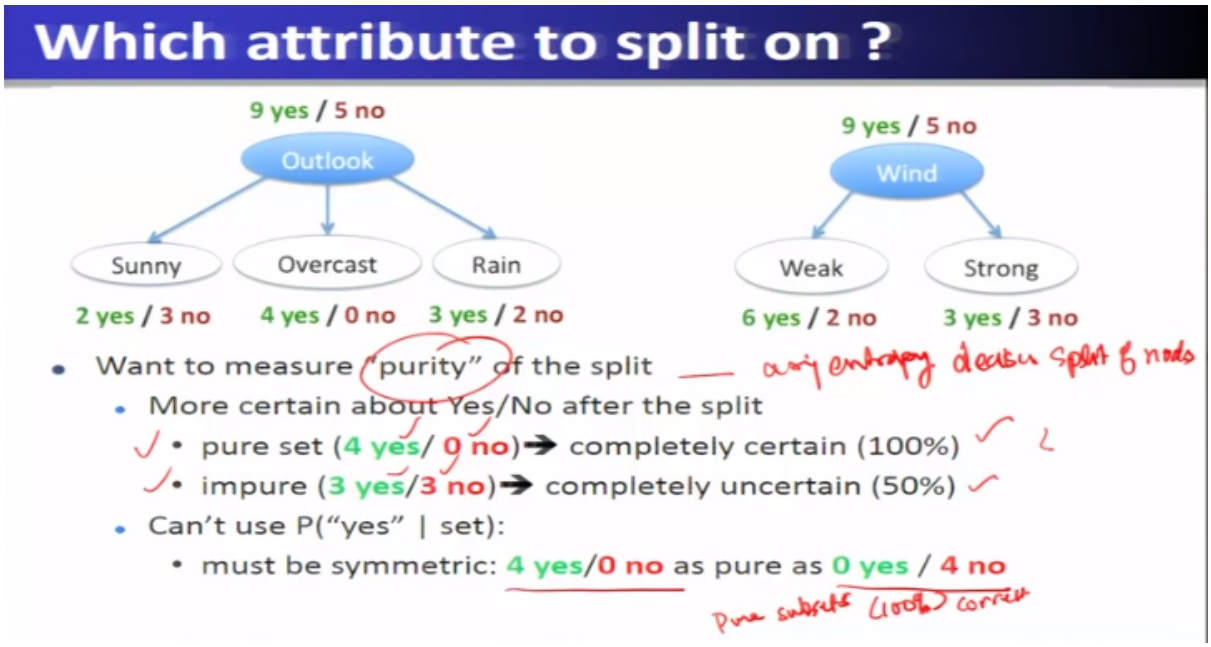


So, with this we have understood, that entropy can be useful to estimate, whether the whether we can go for the split or not? So, if it is a pure set, then entropy the size of the entropy is zero, that means we are not going to a split. If it is a highly uncertain, that is fifty percentage are saying yes and fifty percentage are saying no. Then the value of entropy will become half. So, then trophy will become one, that is highly uncertain, and all other in between which are lying in between. So, their values are between zero and one, that we have seen in the previous example.

Refer Slide Time (26:50)

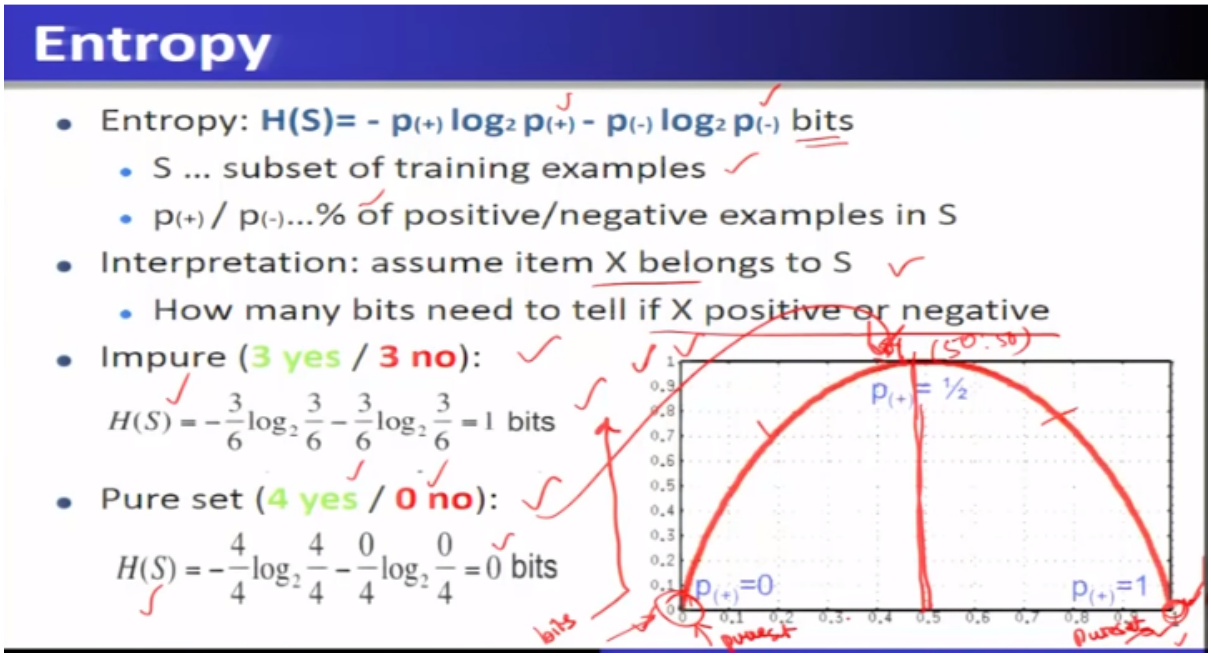


Refer Slide Time (26:51)

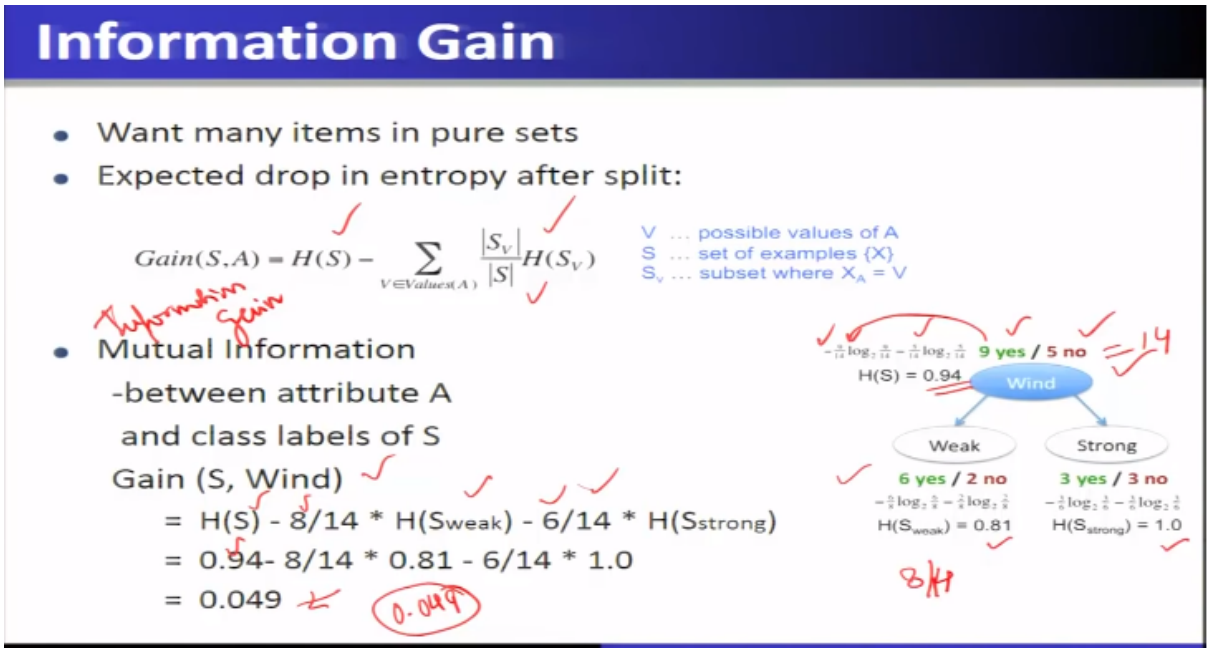


So, we are going to evaluate, this purity of the set using entropy, to decide the split of the nodes.

Refer Slide Time (27:09)



Refer Slide Time (27:10)



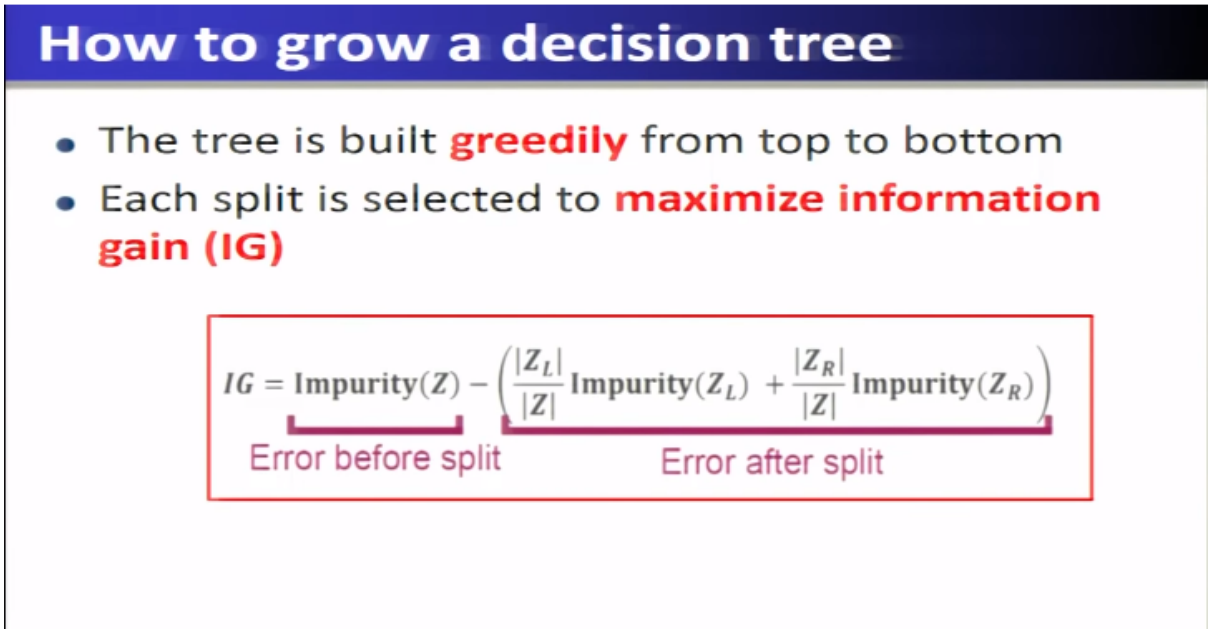
Let us see, how this decision tree has used this concept? This concept is termed as, information gain. So, we want to see, how many items are there in that pure set, and how many and this particular expectation will drop after the split in, that entropy value, and this can be termed as, the gain and that is called information gain. So, information gain will be HS minus the summation of HS $H(S_V)$ and that is the summation of all the values of A and the set of all samples divided by, that means subsets of all the samples divided by the set of all examples. Let us see, how with this information gain, we are going to do this kind of decision, whether to split or not? So, let us see, here in this case of the wind, we are in the wind we have seen, that there are nine yes and five no's. So, the positive variable is nine,

which will be fed, and the total number of samples are fourteen. So, minus nine divided by fourteen log to the base nine divided by fourteen minus five or no five are negative. So, five divided by fourteen log to the base five divided by fourteen, that comes out to be point nine four. So, information game is point nine four here in the wind. How about calculating this same information gained, for weekend for the strong? For the week we obtained point eight one and for the strong we are obtained one. Now, we can see what is the information gained here in this case of wind, whether we can split or not? So, HS is basically the information gain of a wind. So, that is point nine four, and this particular, then for the weak case it is point eight one and it will be divided by eight upon fourteen. So, so, total number of samples if we count here. They are fourteen. So, that total number of fourteen and out of fourteen we are four we case we are taking only eight six two eight. eight divided by fourteen and minus this comes out to be six divided by fourteen and if we evaluate the gain comes out to be zero point zero four nine information gain. So, if we split, then the net gain will be the positive gain of point zero four nine. Hence, it is better to split.

Refer Slide Time (30:41)

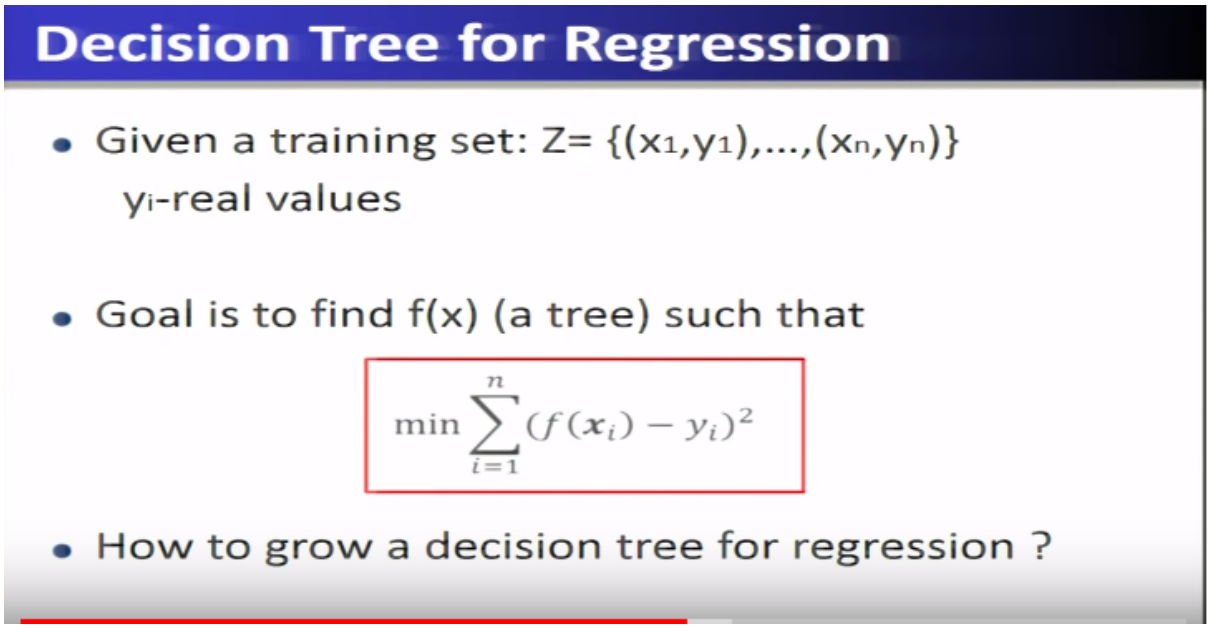
So, this way using information gain, we can estimate, or the decision tree will make the decision whether to split or not?

Refer Slide Time (30:51)



Now, let us see, how to grow, how to build, this particular decision tree. So, the trees is built greedily from top to the bottom and each split is selected to maximize the information gained. So, what we have seen here is, that in the previous example, that after the split the gain is positive. So, we have to maximize after the split. Hence, we are going to split in this particular case.

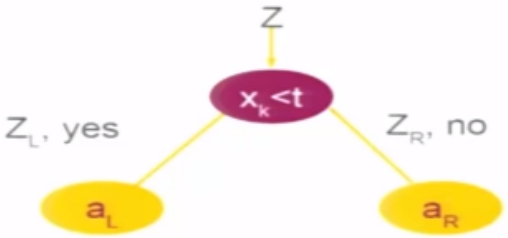
Refer Slide Time (31:16)



Refer Slide Time (31:20)

How to find the best split

- A tree is built from top to bottom. And at each step, you should find the best split in the internal node. You get a test dataset Z. And here is a splitting criteria xk less than t or xk is greater or equal than a threshold t. The problem is how to find k, the number of feature and the threshold t. And also you need to find values in leaves aL and aR.



Refer Slide Time (31:29)

What happens without a split?

- **What happens without a split?**
- Without a split, the best you can do is to predict one number, 'a'. It makes sense to select such a number 'a' to minimize the squared error.
- It is very easy to prove that such variable a equals an average of all yi.
- Thus you need to calculate the average value of all the targets.
- Here ā (a-hat)denotes the average value.
- Impurity z equals the mean squared error if we use ā (a-hat) as a prediction.

Without split: predict one number *a*

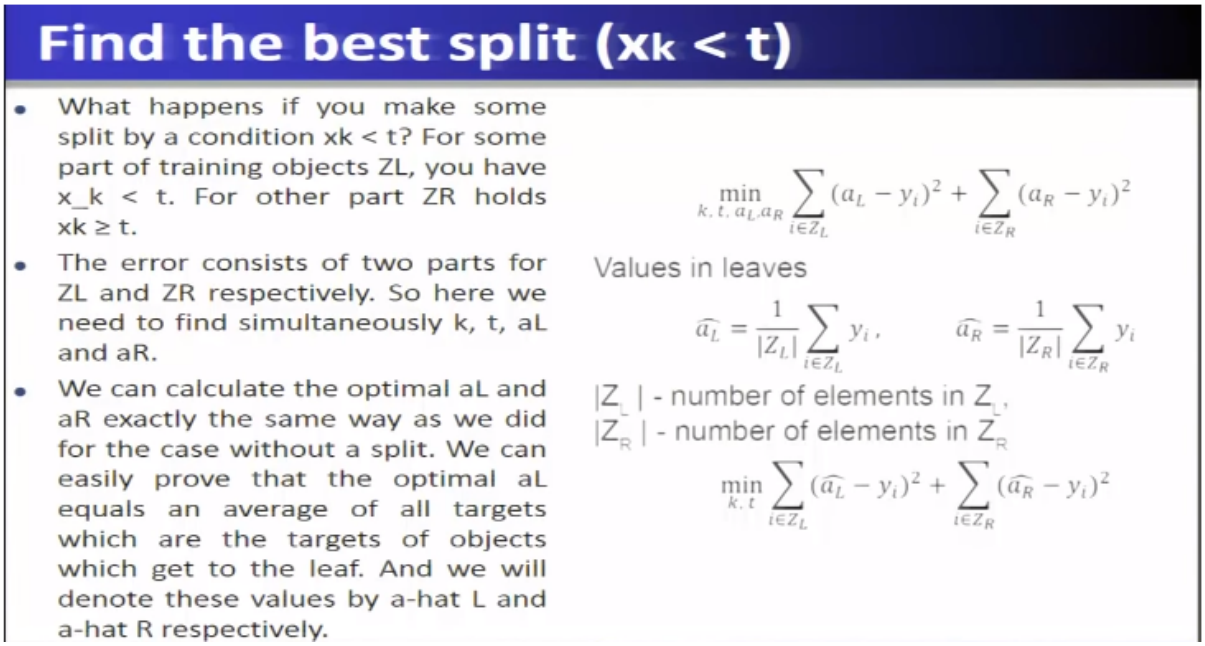
$$\hat{a} = \min_a \sum_{i \in Z} (a - y_i)^2$$

$$\hat{a} = \frac{1}{|Z|} \sum_{i \in Z} y_i$$

|Z| - number of elements in Z

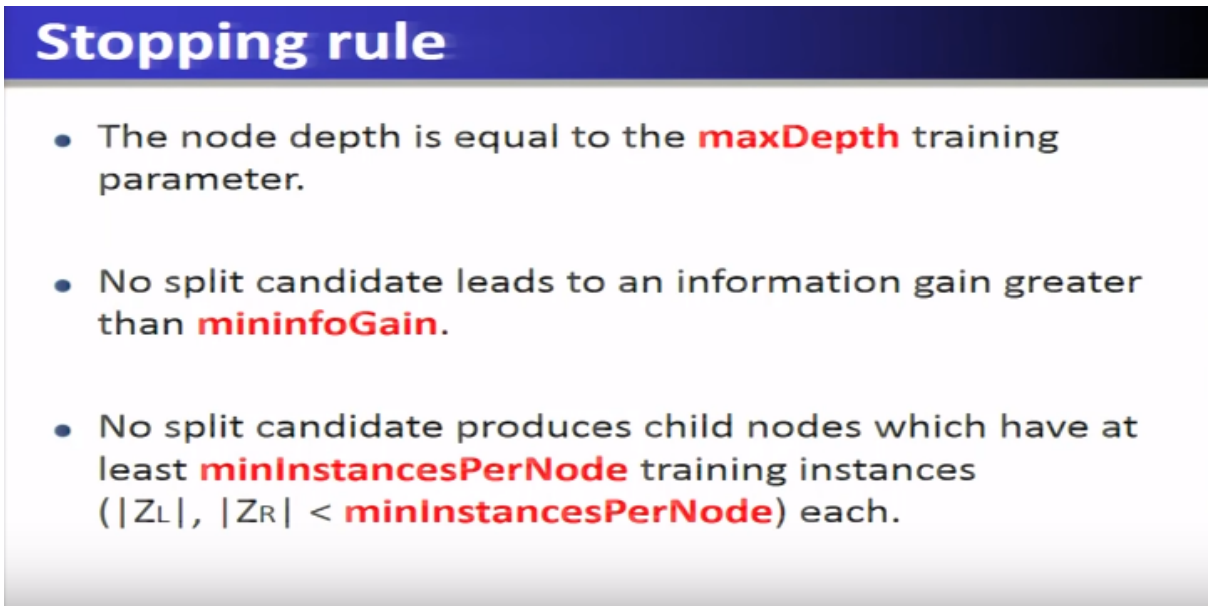
$$\text{Impurity}(Z) = \frac{1}{|Z|} \sum_{i \in Z} (\hat{a} - y_i)^2$$

Refer Slide Time (31:35)



So, we can see here, by this way we can find out the the splits, and we have already seen, that if we can do this kind of predictions. To find out this whether to split a node or not. So, whenever there will be information gained and to maximize the gain according to that maximizing the information gained this particular splitting is to be done. So, we have to find out the best split, and we have to do the decision based on that.

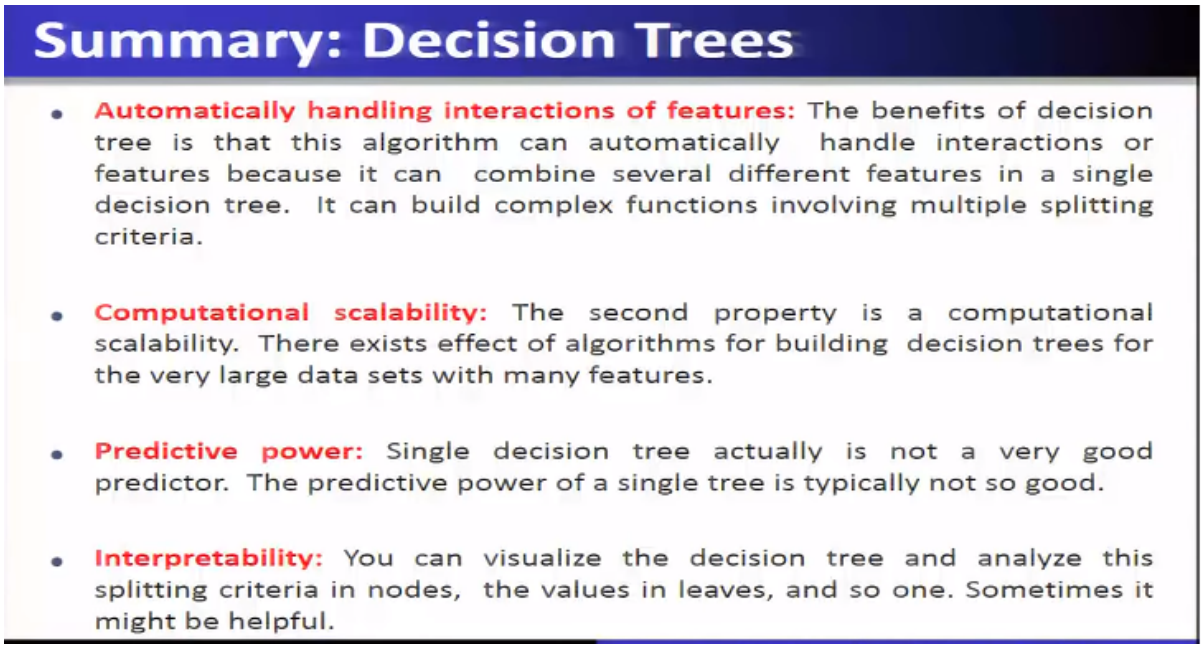
Refer Slide Time (32:00)



Now, there is another criteria which is called a stopping criteria, that the the growth of the spanning tree or the building of the spanning tree it has to be stopped. So, the node depth is equal to the maximum and depth of the training parameter, and no split candidate leads to an information getting

greater than minimum information gain, and no candidate produces the child, which have at least the minimum instances per node training instances.

Refer Slide Time (32:33)

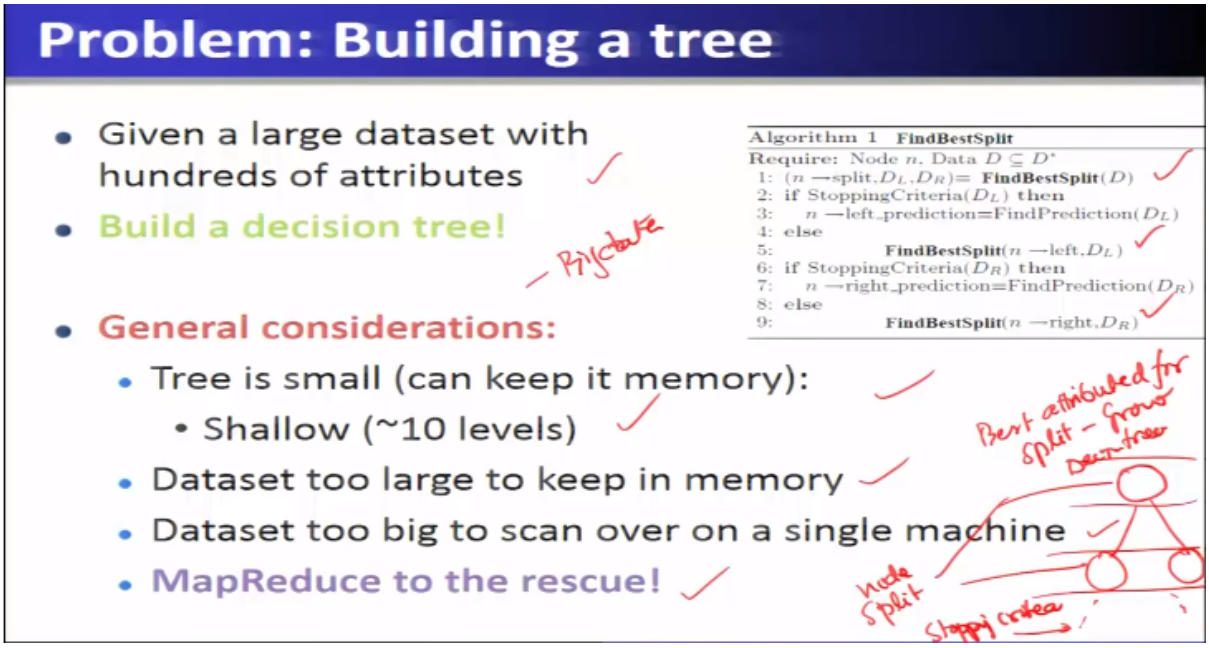


So, in summary what we have seen in the decision tree is, that here the automatic handling of interactions of the features is done, and computationally scalability also we have seen, and interpretability aspects also.

Refer Slide Time (32:47)

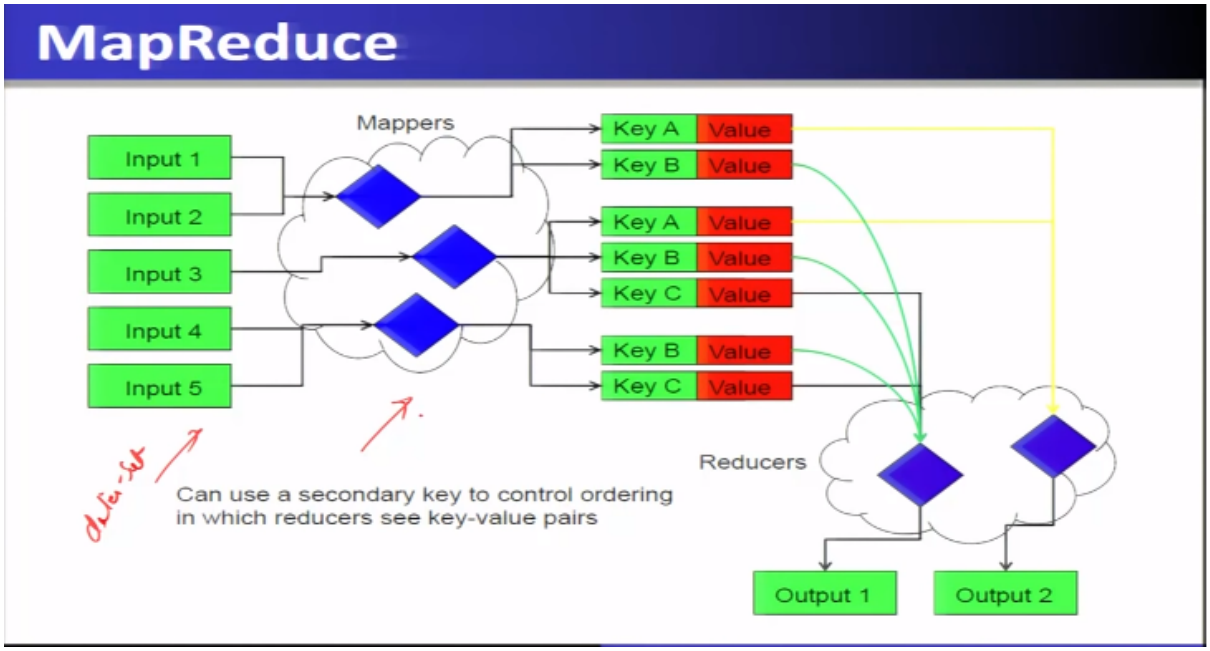
Now, let us see, how to build tree using MapReduce.

Refer Slide Time (32:53)



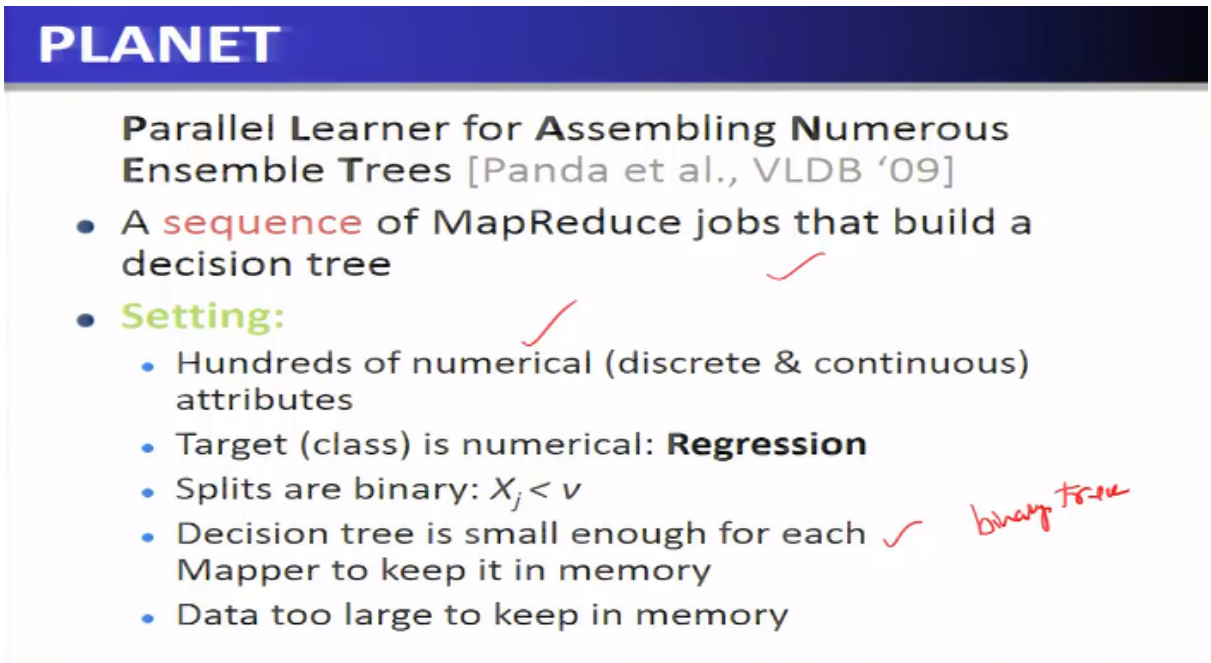
So, now MapReduce will be applied for the decision tree algorithm, and here we will see the approach and we will see more detail of it. How this MapReduce will be applied in the decision tree. So, that the big data analytics can be performed with the help of this parallel a decision tree Clementessin Now, here we are given a large data sets, and which has a lot of attribute the size of the data set is also big, and the number of attributes are also more, and we want to build a decision tree and that is what is called the Big Data applying that is the big data analysis. So, here we can we have to see is, that the tree all the data is very big, but the tree is small and it can be stay in the memory and maximum levels of that particular tree, which is extracted out of big data is on off out of the ten levels, and also we have to keep the data set the very large data set into the memory and this data set is too big to be kept and one machine. So, basically MapReduce we are going to use which is going to going to compute over the clusters of machines. Now, here here in this particular technique which is called the decision tree. We have to find out the best the splits and, then we have to find out which attribute along with, which we are going to build the tree, that is the single tree which we are going to build. So, we have to first find out which is the best variable around, which the split has to be done or the tree has to be constructed. So, the best we have to find out the best attribute, for the split to grow the the decision tree. So, after having identified a particular attribute. Now, we have to identify the how the best split is to be done? So, that in the left side and the right side of the tree can basically grow and so on. So, these particular growth of the spanning of the decision tree which we are going to see level wise growth of the tree and also the stopping criteria. When we have to stop further growth of the tree. So, this particular decision is called when we are going to split. So, this is whether the node, what is the condition in which we are going to split the node, along whether to split or not and if it is then which particular attribute will be the best to split. So, all these things we are going to discuss which are going to be implemented using MapReduce in.

Refer Slide Time (36:17)



So, MapReduce we have already seen, that it is it will the input data set, is now divided into the into the chunks, and these chunks are given to the map or functions and the mapper function will generate the key value pair, which is further given to the reduce functions and they will generate the output.

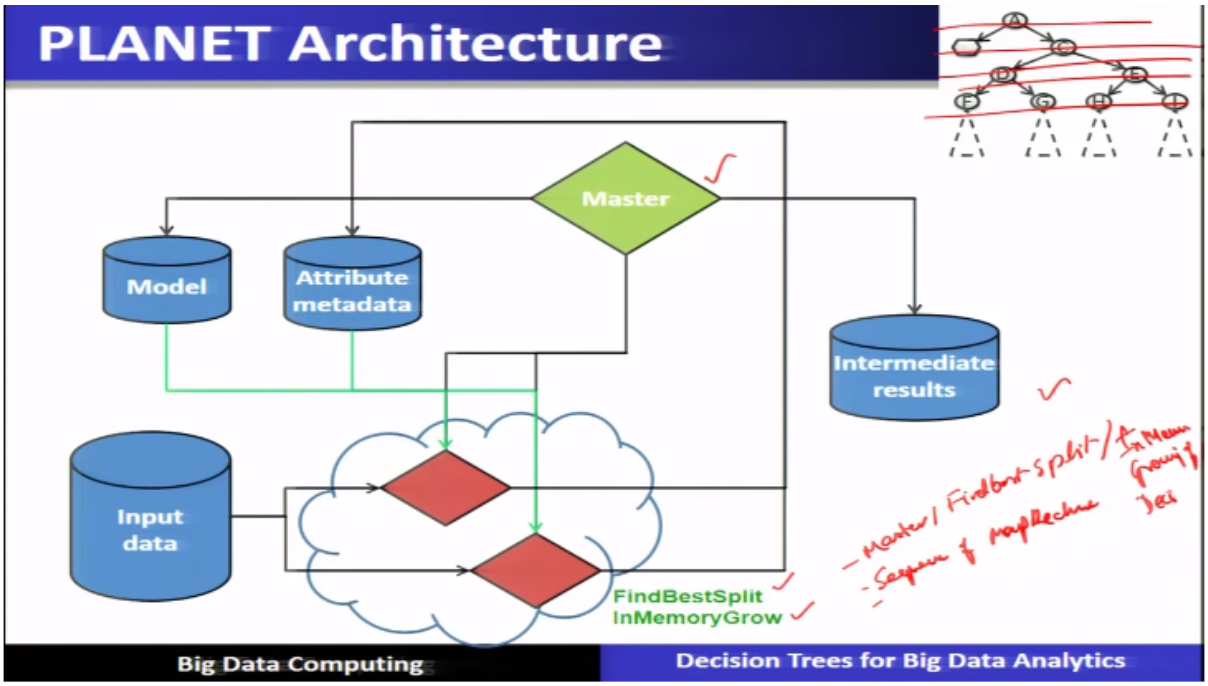
Refer Slide Time (36:47)



Now, this particular implementation of a MapReduce for decision tree, is published in a paper which is called planet by the Google. So, parallel learner for assembling numerous n sampling n symbol trees which we are going to see here as, the MapReduce implementation of a decision tree. So, this is nothing but a sequence of MapReduce job, that will build the decision trees, and here we assume, that

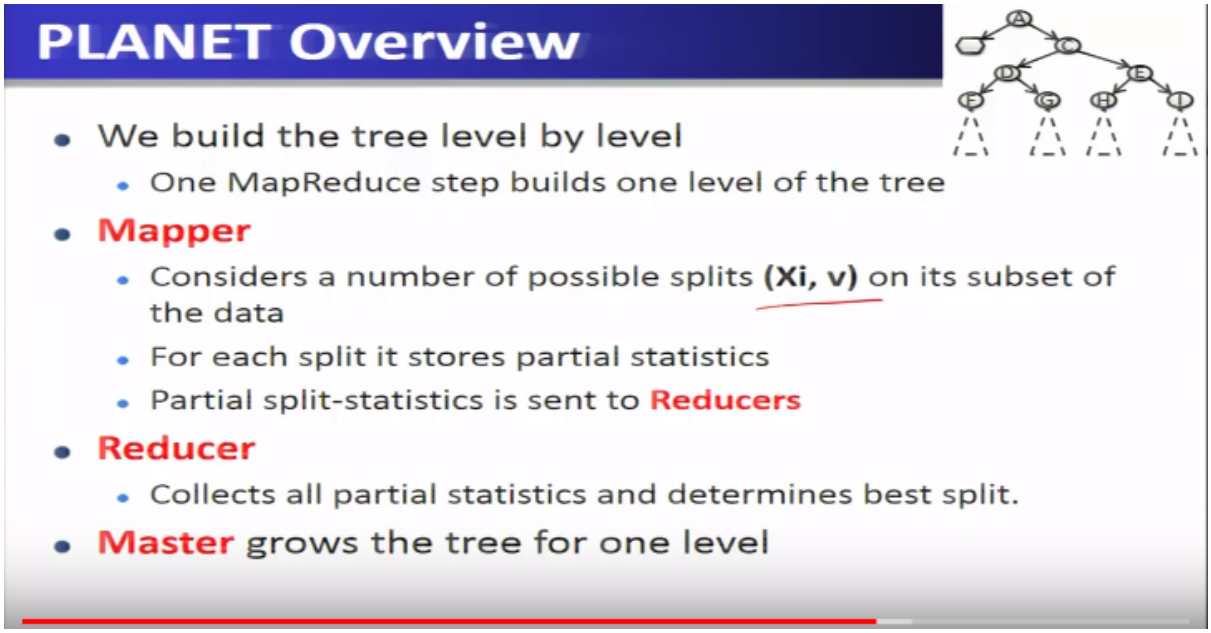
the values which are given in the data set, they are basically hundreds of numerical attributes, and we are going to build the binary tree, which is the decision tree, which is a binary which is a binary tree and this particular tree which we assume, that is small enough to be for each mapper to be kept in the memory and data is too large.

Refer Slide Time (37:45)



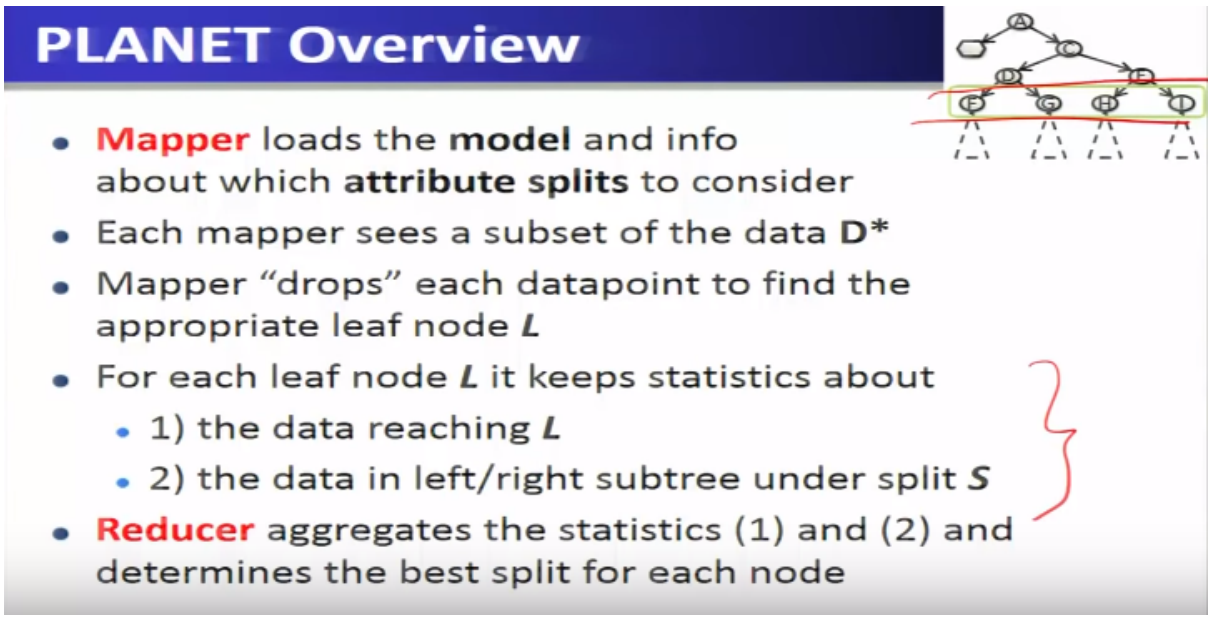
Let us see the different constituents in this architecture. So, here we see, that there will be a master node, and this particular master node in turn will decide about finding the best split and also the in memory growth of that tree at the mapper end, and the intermediate results are also stored, and this all happens in the coordination of the master. So, the tree is built level wise using these different, MapReduce sequence of MapReduce operations. MapReduce operations are being coordinated by the master. There will be a master iron, which will decide, the best is split and also tries to decide about in memory growing of the decision tree. I had the mapper.

Refer Slide Time (39:08)



So, let us see the planet overview. So, here in this particular method we build the tree level by level and one MapReduce step will build one level of a tree. So, if there is a 10 level. So, 10 different MapReduce in sequence has to execute. As far as the mapper is concerned mapper will consider the number of possible splits on the subset of the data, and this comprises of Xi comma V and for each is split. It is stores the partial statistics partial split statistics is, then sent to the reducers, reducers we collect all the partial statistics and determines the best split, and master grows, the tree for one more level.

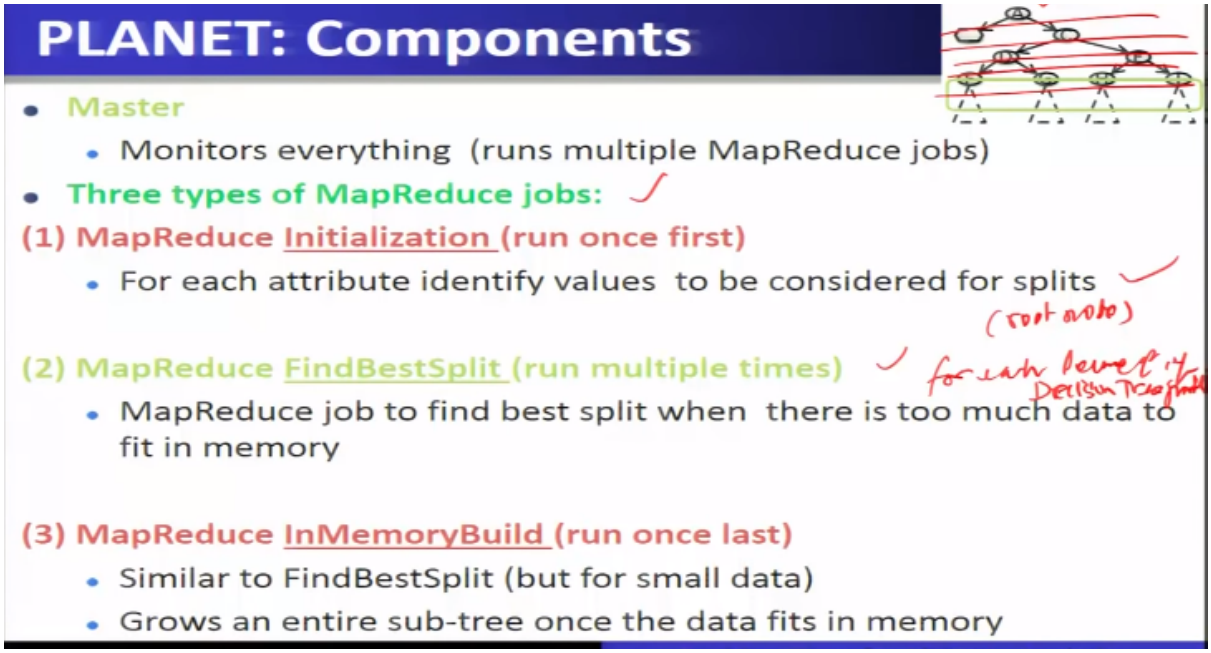
Refer Slide Time (40:05)



So, here the mapper loads the model and information about the attribute splits to consider. So, each mapper sees a subset of the data D star and the mapper drops each data point to find the appropriate

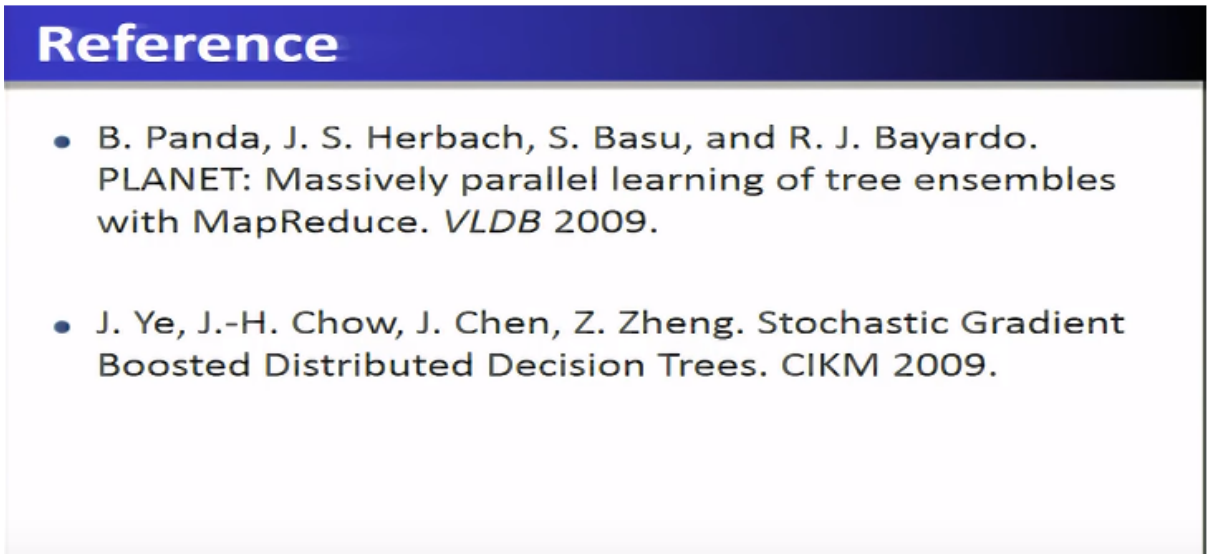
leaf node L, and for each node L it keeps the statistics about, the data reaching L and the data in the left and right subtree under the splits S. This way the mapper. So, each mapper will now, has to deal with a particular level of the nodes and has to decide, whether it has to be further split or not. If it is to be splitted further, then another iteration will go on with the next set of MapReduce job. Reducer aggregates the statistics of one and two and determines the best split at each node.

Refer Slide Time (41:00)



So, overall different components in this MapReduce implementation of decision tree is like this. So, master will monitor everything and runs multiple MapReduce jobs. So, besides master there are three different type of MapReduce jobs, which will be executed. So, first is the initialization of MapReduce, that is it will be run once. So, for each attribute, it will identify the values to be considered, for the split. So, for example the root node so, for the root node decision, has to be done in during the initialization phase, and then it will be a second set of, MapReduce job to find out the best is split, and it will run multiple times for each level of decision tree growth. So, the second step will keep on increasing keep on growing the the the levels of the tree as they are being invoked. So, the number of time is it it is invoked. So, that indicates, that it will be growing in that levels. So, that particular so MapReduce job is to find the best displayed, when there are too much of data to fit in the memory.So, this way this MapReduce will try to find out the best split and keep on splitting and also, it will be then the next step is finally the MapReduce will build thein-memory build, that is sub trees are built at each MapReduce then they will be runs once. So, similar to find the best split for small data. It will grow and entire sub tree once the data fits in the memory. So, these three different MapReduce jobs one is for the root node, that is the initialization. The second one is at the level wise splitting of the nodes, and growing the decision tree, and finally it has to take this subtrees, which are in memory and built during each run. So, it will be done once.

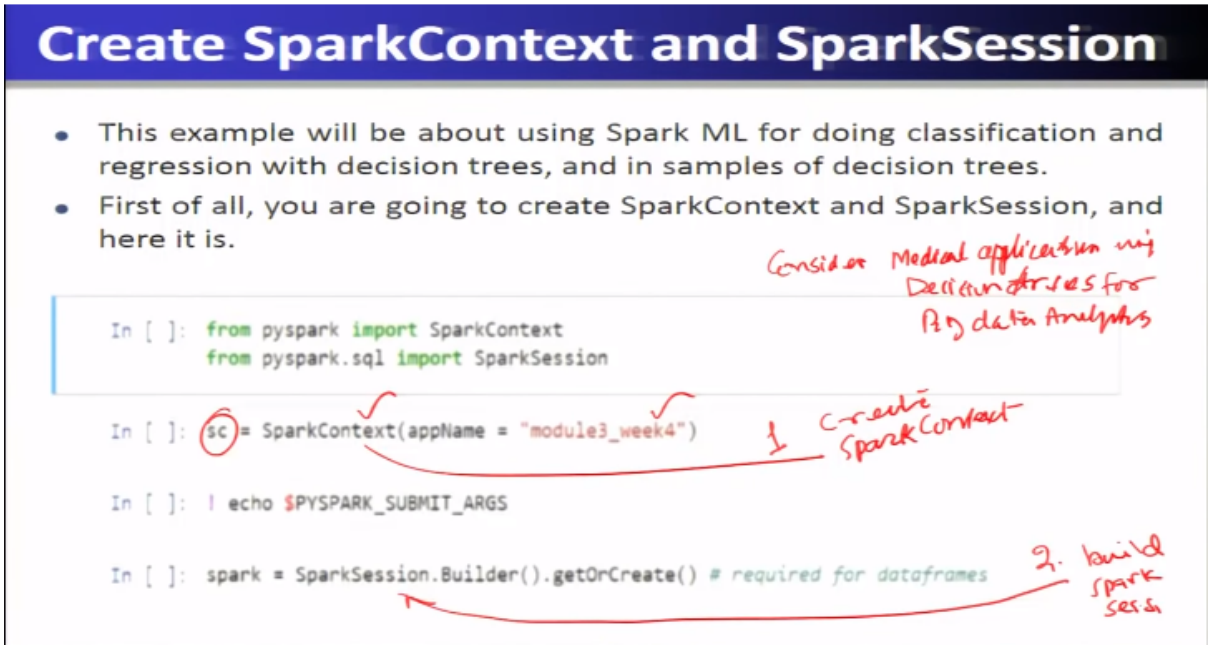
Refer Slide Time (43:33)



These are the references for this discussion, which we have just seen.

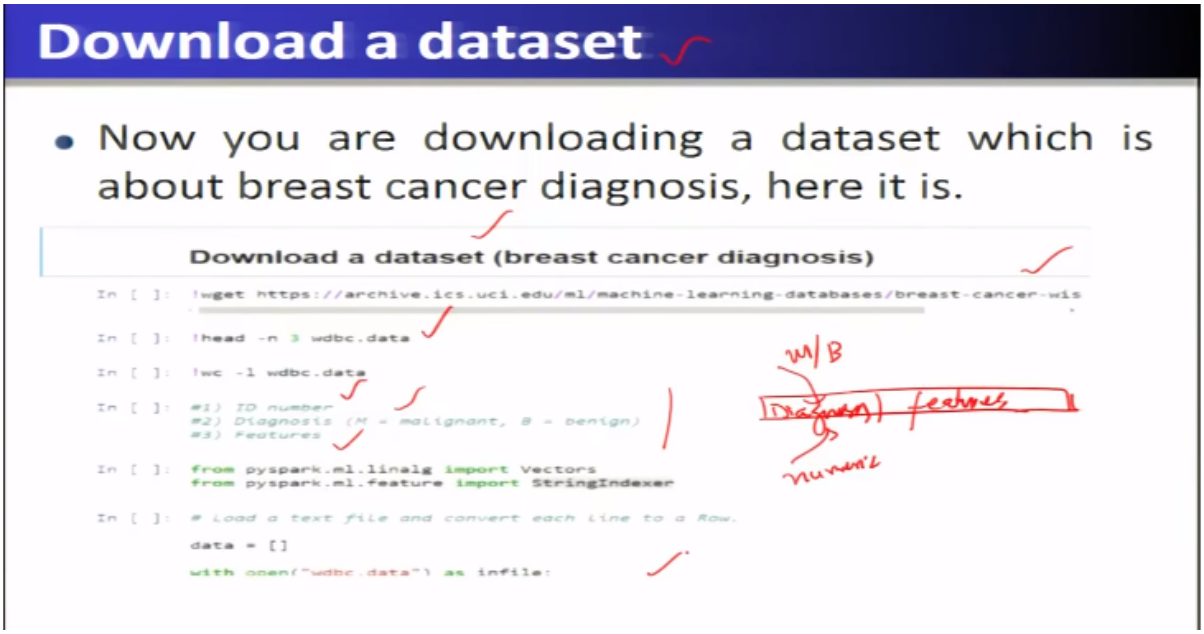
Refer Slide Time (43:43)

Example medical application using decision tree in spark machine learning.
Refer Slide Time (43:50)



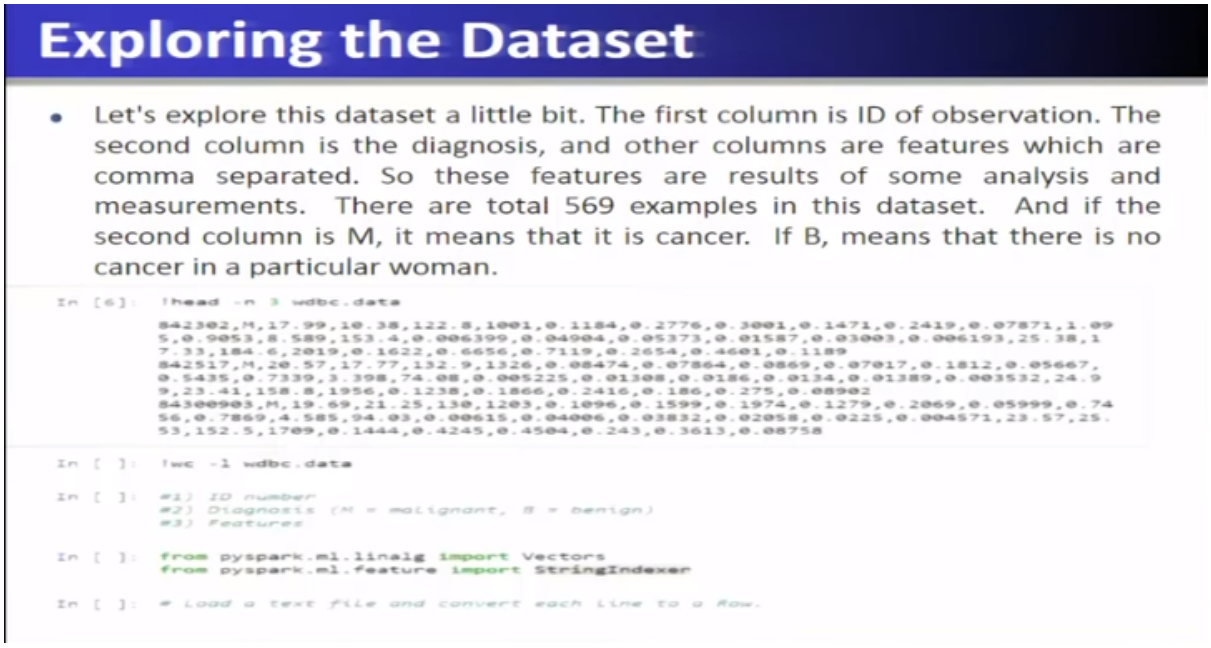
Now, we will consider an example of the medical application of the application of decision trees, for using spark a ML. So, in this example, we will see how the classification using the decision trees is done. So, let us see the steps, for doing this decision tree based analytics, for the medical application. So, here we are going to consider, the medical application using decision trees, for big data analytics. Now, here we have to see, that we have to give the name of the application and now, we have to create the spark context, using this application name. So, we call it as by spark context. We have to create a spark context. After creating the spark context, then we have to build a session that is called a spark session. This is step number one, and this is step number two, which is essential in all such applications, which runs under spark. First, we have to create the context, then a spark session and this is required to build the data frames.

Refer Slide Time (46:10)



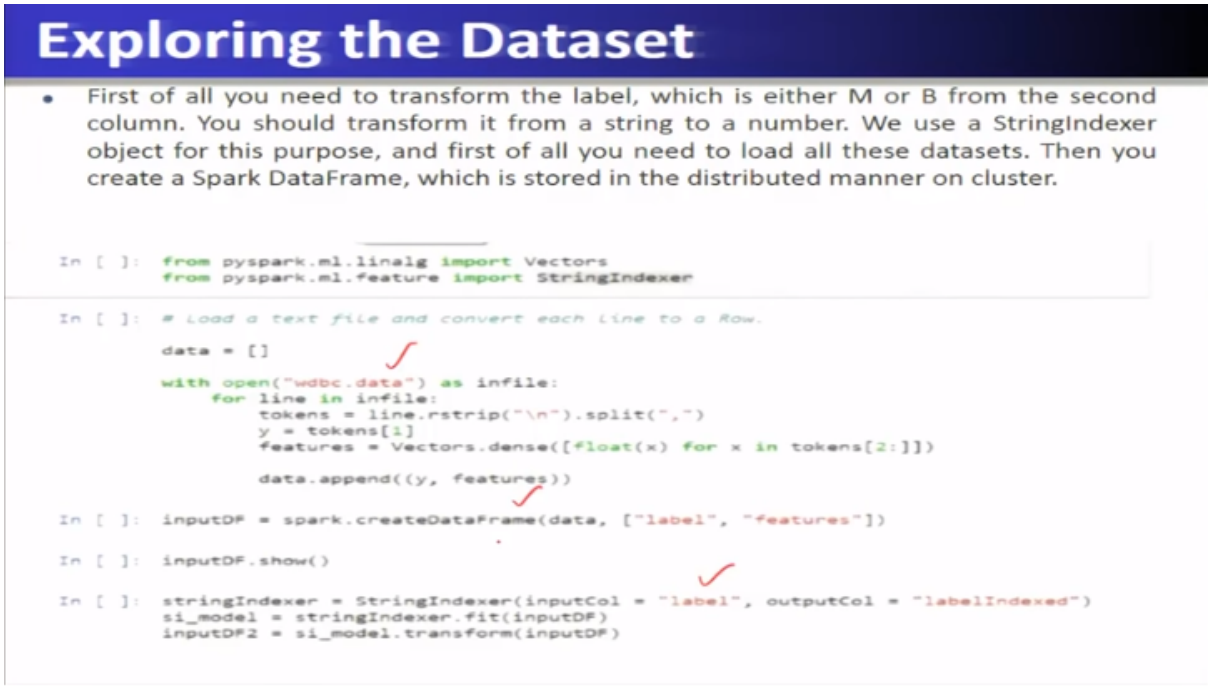
The next step would be to download the data set, which is used here in this example. So, we will download the data set, from some link, and let us, call it as, this data set which is being downloaded. Once the data set is available and is downloaded into the spark system. Then we have to see, that this particular data set has three different parts. That is in the form of key value pair. it has an ID number and, then it will have, the key as, whether it is having a medical problem yes or no means align or having benign and then different features of it. So, the data will be in this particular form. It will be a diagnosis, and then it will be a features. Now, this diagnosis if it is M or B it has to be converted into the numeric form. So, we have to convert, we have to load into file and we have to convert this field into the numeric field.

Refer Slide Time (47:36)

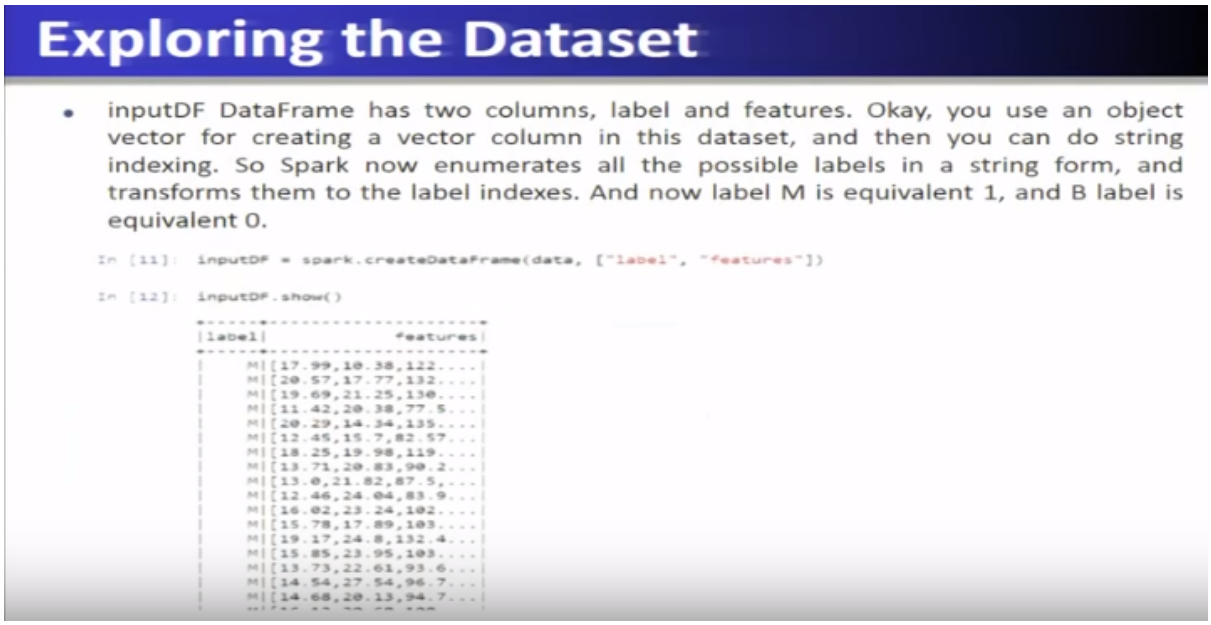


So, out of this particular data set, that is five sixty nine examples in a data set we have to convert in this particular field of diagnosis into the numeric values, and then we have to see this particular explore the data set.

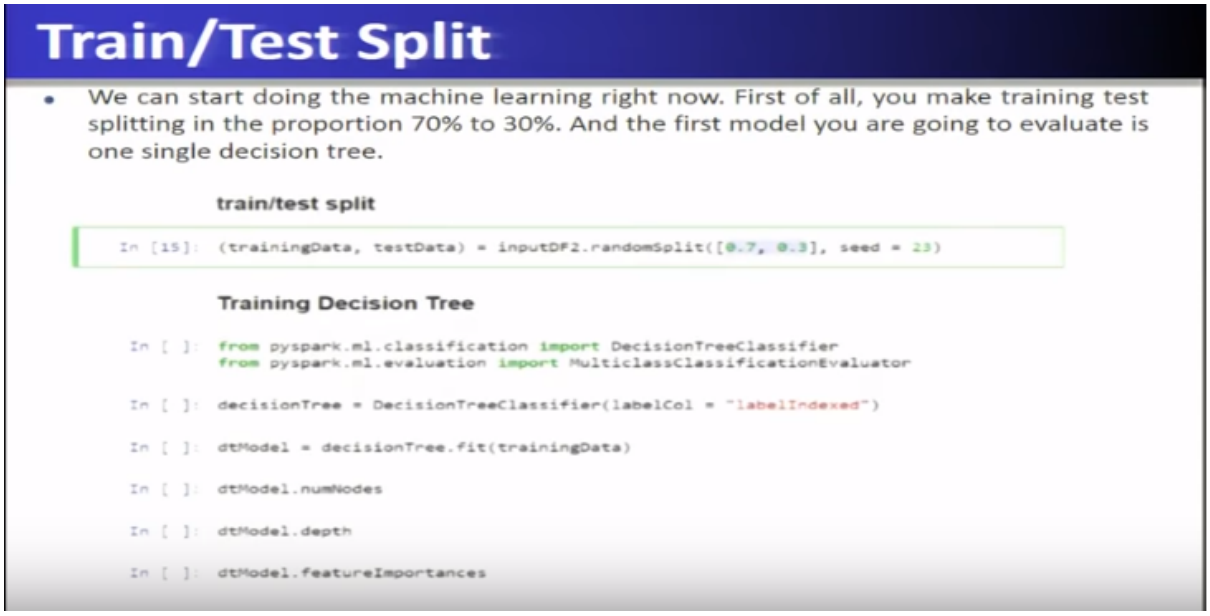
Refer Slide Time (47:48)



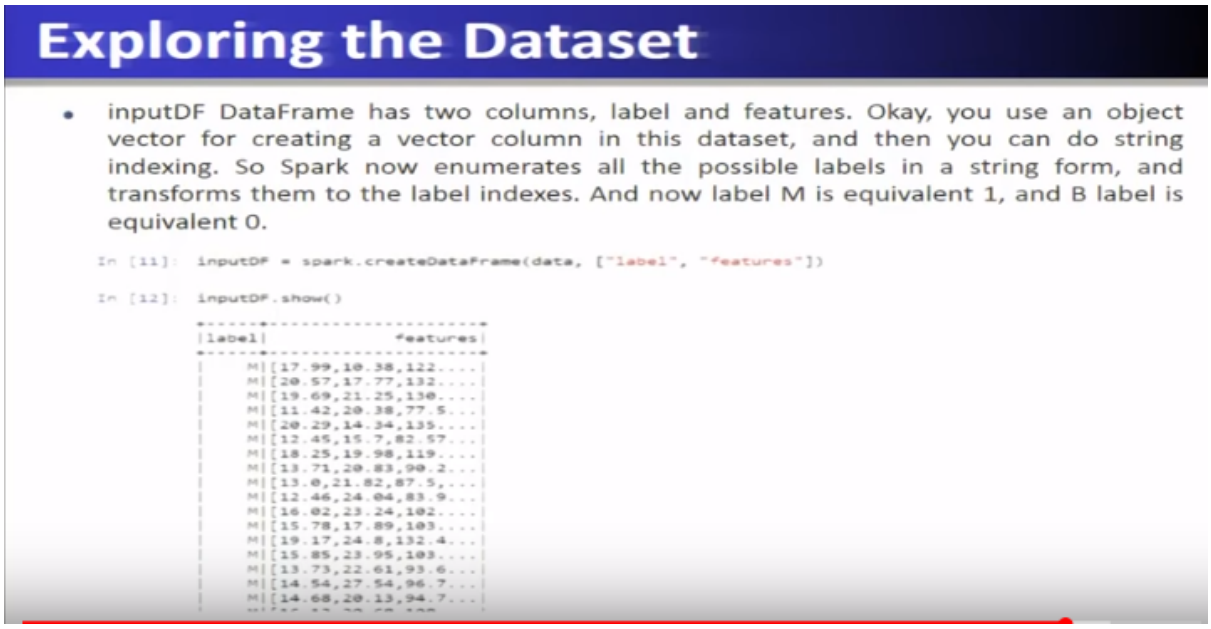
So, after having read this particular data set, which we have downloaded. Now, it will be creating the data frames with the label and the feature, and the label will be converted into into the numeric values,
Refer Slide Time (48:19)



Refer Slide Time (48:24)

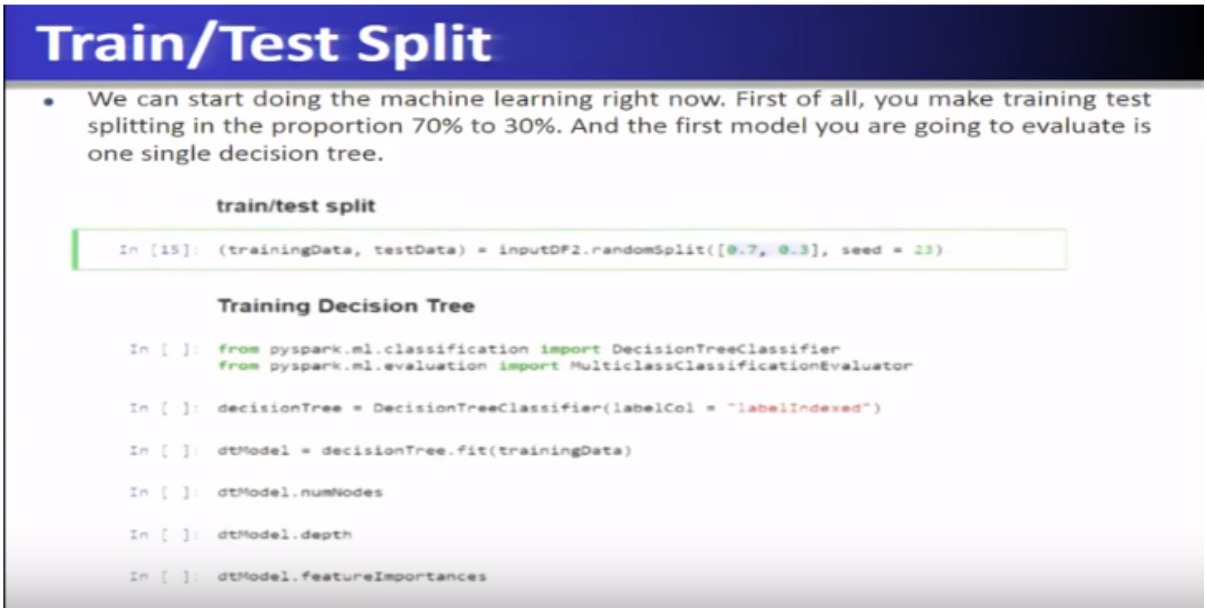


Refer Slide Time (48:27)



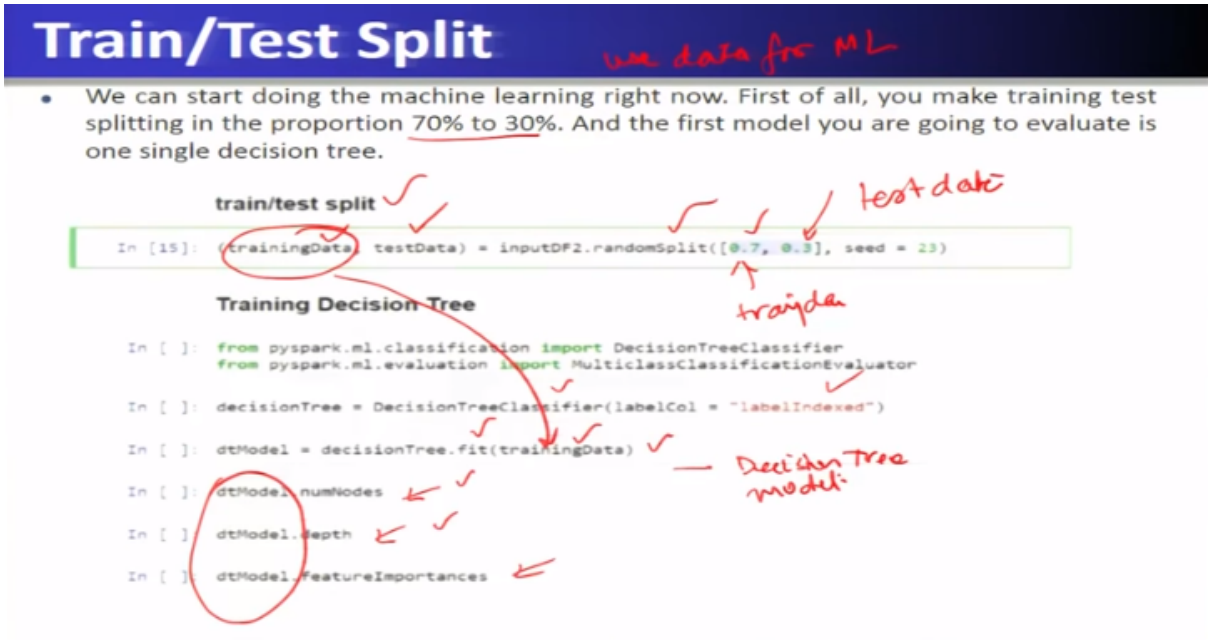
and after that this particular data set, which will having this values, converted into so, these M fields will be converted into the numeric values, and then now after converting it, now it will be ready. So, the conversion will be M is equivalent to one and V is equivalent to zero. So, wherever one is there M is there. So, it will be one and wherever B is there. So, after this conversion now this particular data set will be now, ready for the use of machine learning.

Refer Slide Time (48:58)

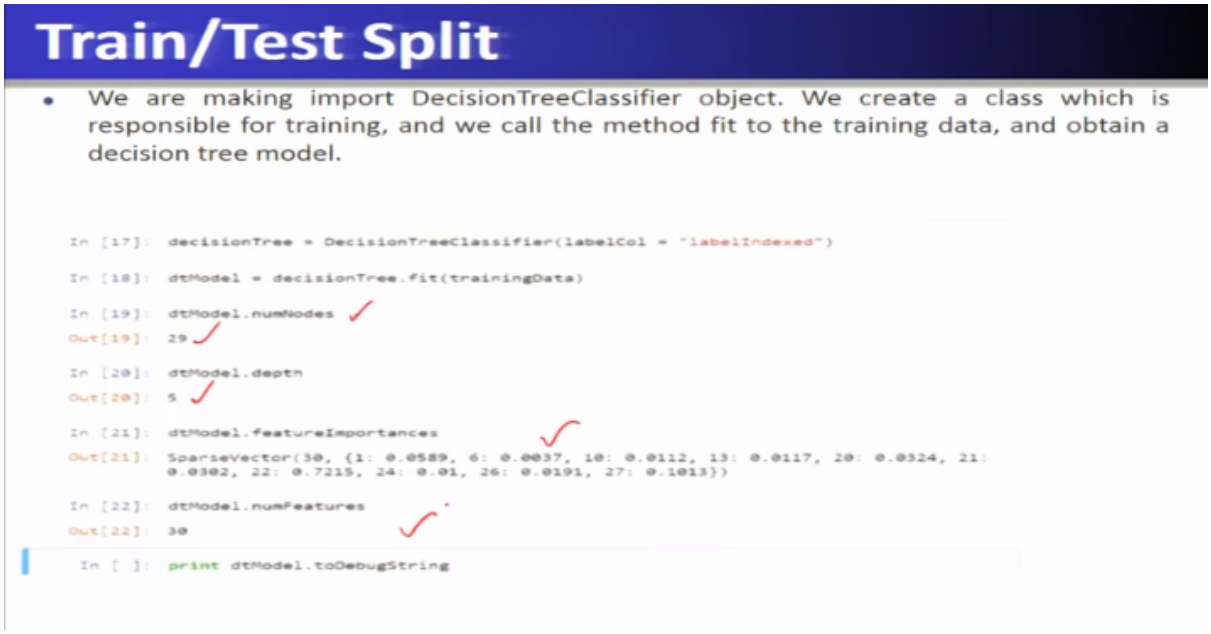


So, the first after preparing the data. Now, we have to use the data for the machine learning, for that we have to split this data into the training data set, and the test dataset, into the proportion of seventy and thirty, So, to split this data set into the training and test data splits. Now, we have to use this particular spark come on random split, that is seventy percent is of training data, and thirty percent is the test data. So, after this splitting of the training data and the test data. Now, we are going to apply this decision tree algorithm. So, we will now use the decision tree classifier and now we will use this label indexed, which we have seen in the previous slide, and then we will do this decision tree fit on the decision on the training data set, which we have already obtained from the the data set is split seventy percent of it. So, after this step we are ready with the the decision tree. Now, we can see out of this decision tree it's different parameters such as, how many nodes are there in the decision tree what is the depth of the decision tree? and also, the what are the features which are of importance? and this will build the model of a decision tree. So, decision tree model, is being built here after this execution.

Refer Slide Time (51:14)

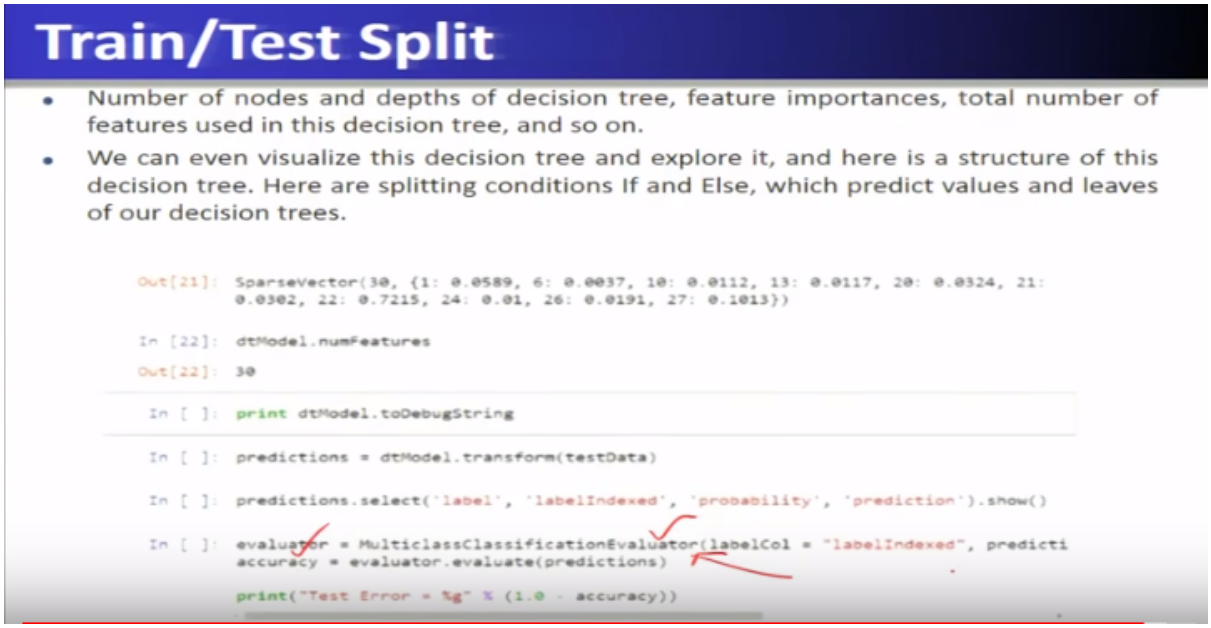


Refer Slide Time (51:13)



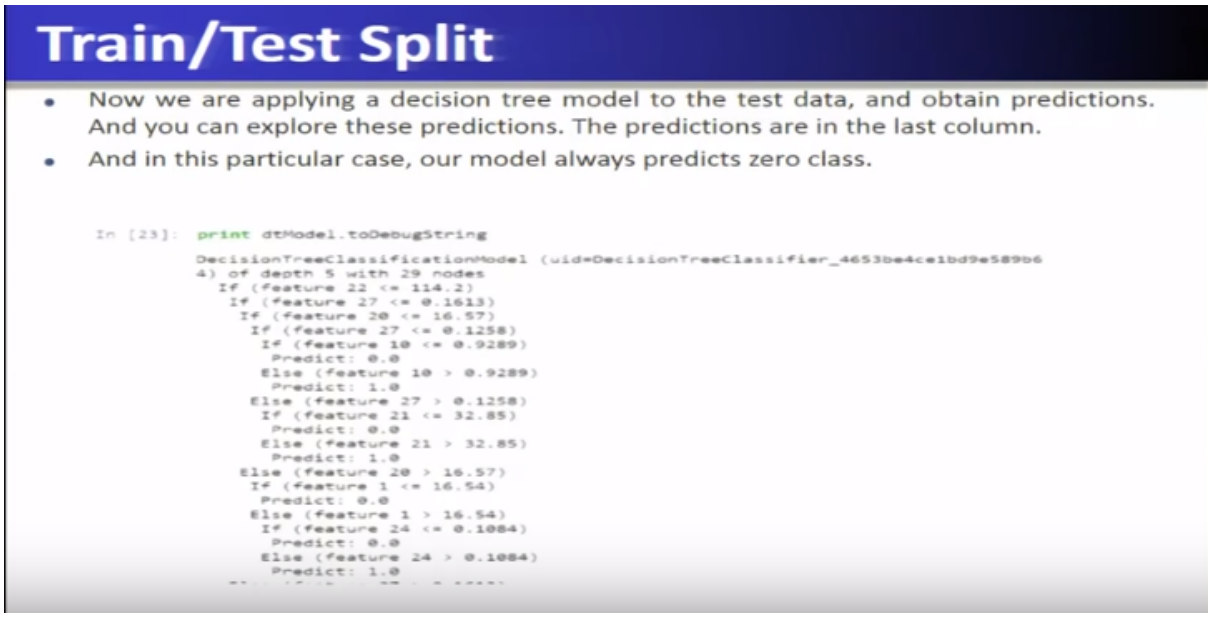
Now, now we can see, that if we find out how many in this particular data set of around six hundred different samples. We have seen, that there are twenty nine and nodes are there, and the number of depth is only five, and different important features, which we have obtained in the form of the graph and number of features are thirty.

Refer Slide Time (51:46)

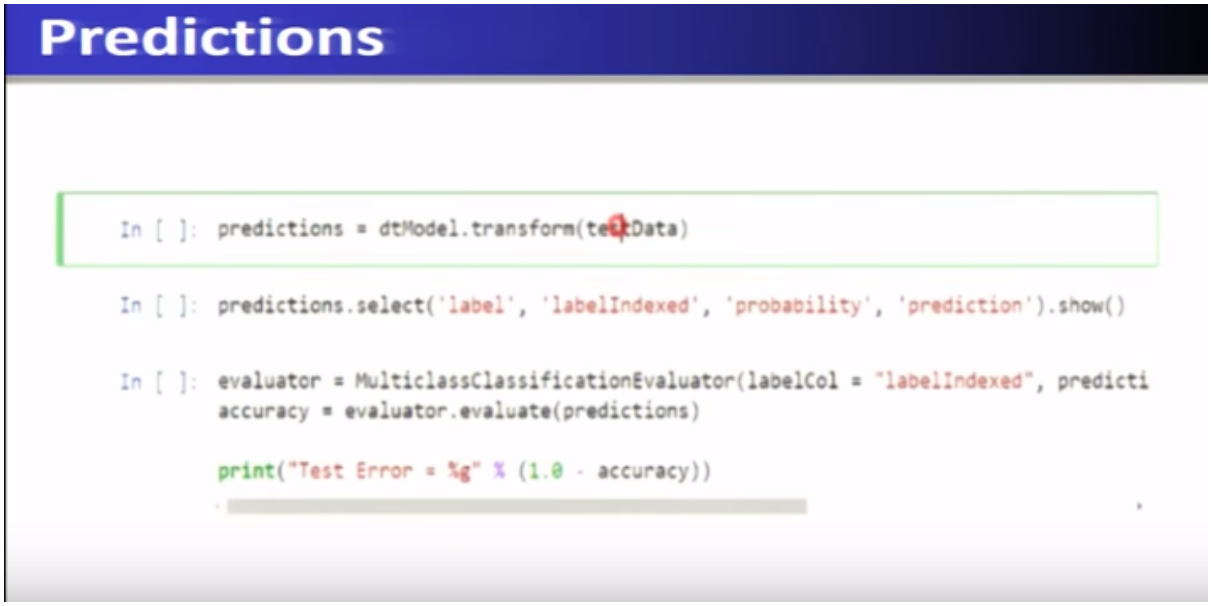


So, after constructing this particular decision tree. Now, we have to see that we have to evaluate this particular decision tree, whether it is the quality wise whether it is good or bad. How it is good for the predictions? So, evaluation is done. So, multiclass classifier multiclass classification evaluator is used for this particular purpose, to evaluate the predictions accuracy.

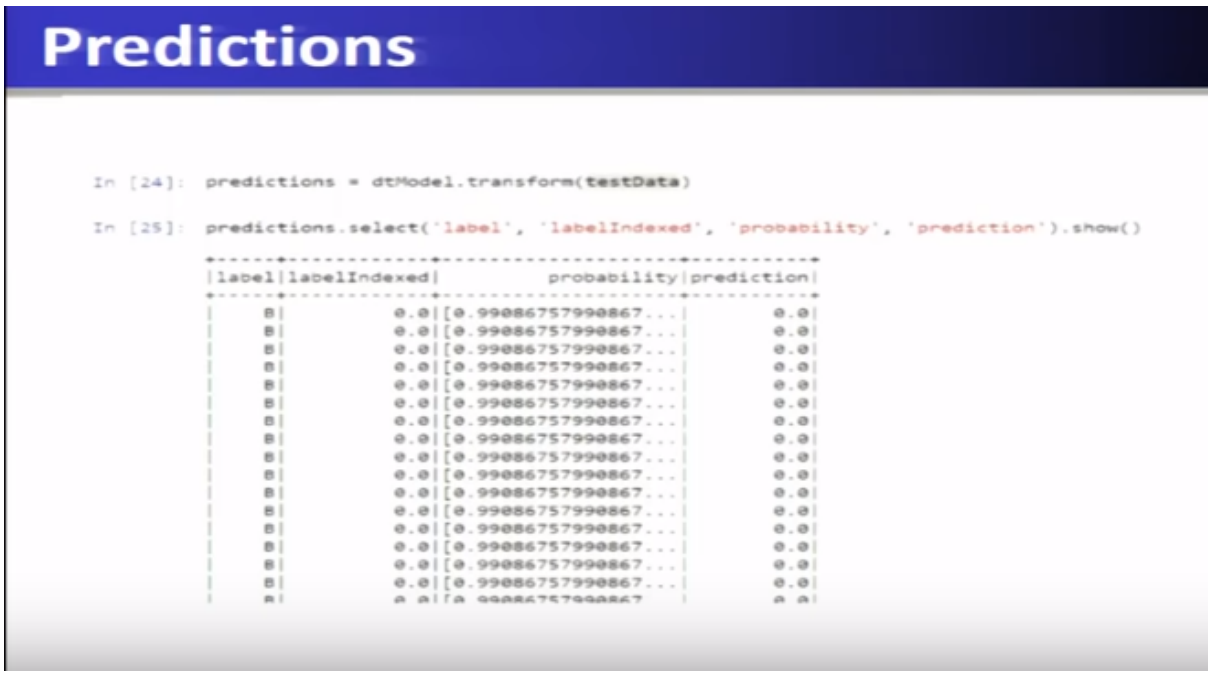
Refer Slide Time (52:20)



Refer Slide Time (52:23)

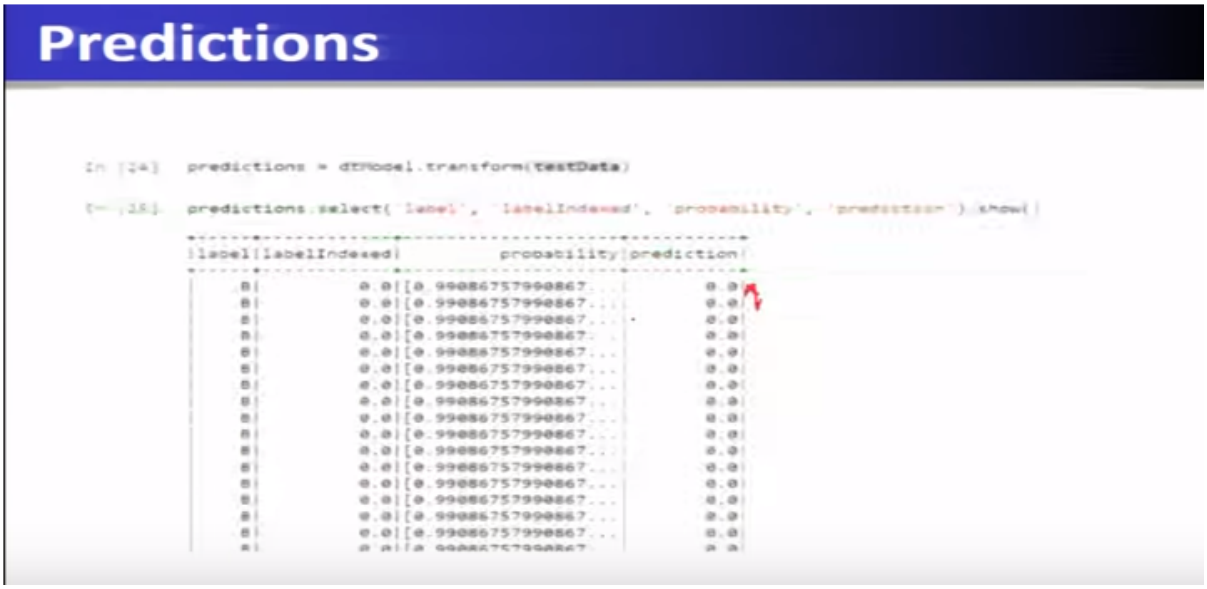


Refer Slide Time (52:29)



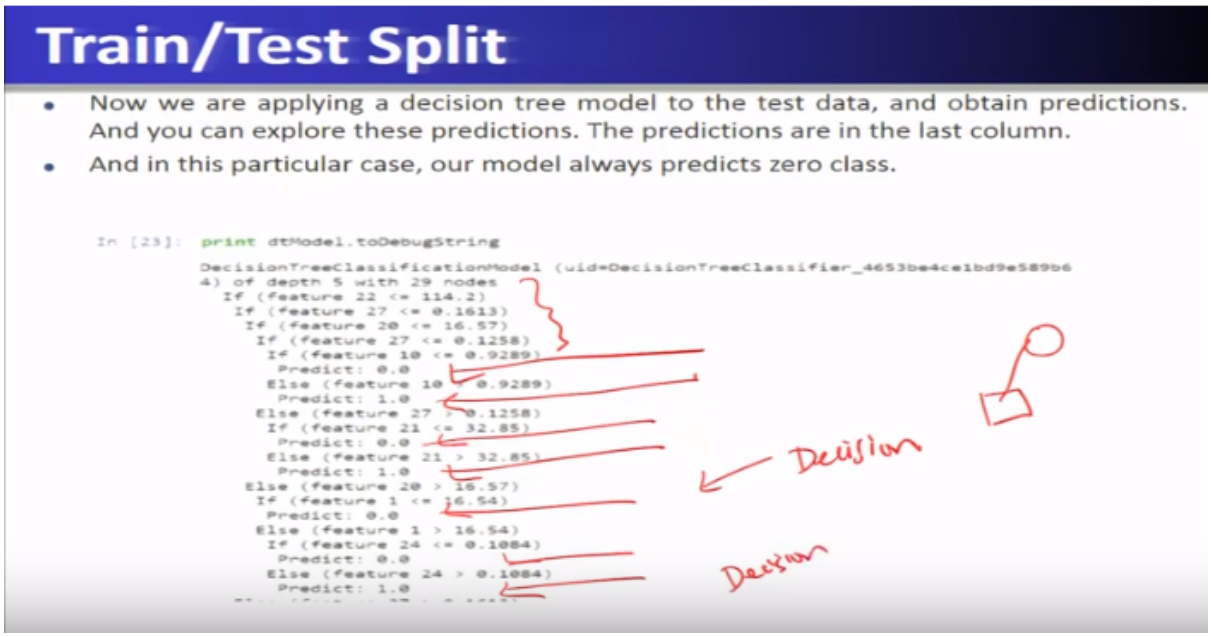
So, we want to evaluate the prediction accuracy and by that by this particular method we have to see that the prediction accuracy, which is calculated for this decision tree construction is around ninety six percent and this is fairly good, which we have generated.

Refer Slide Time (52:41)



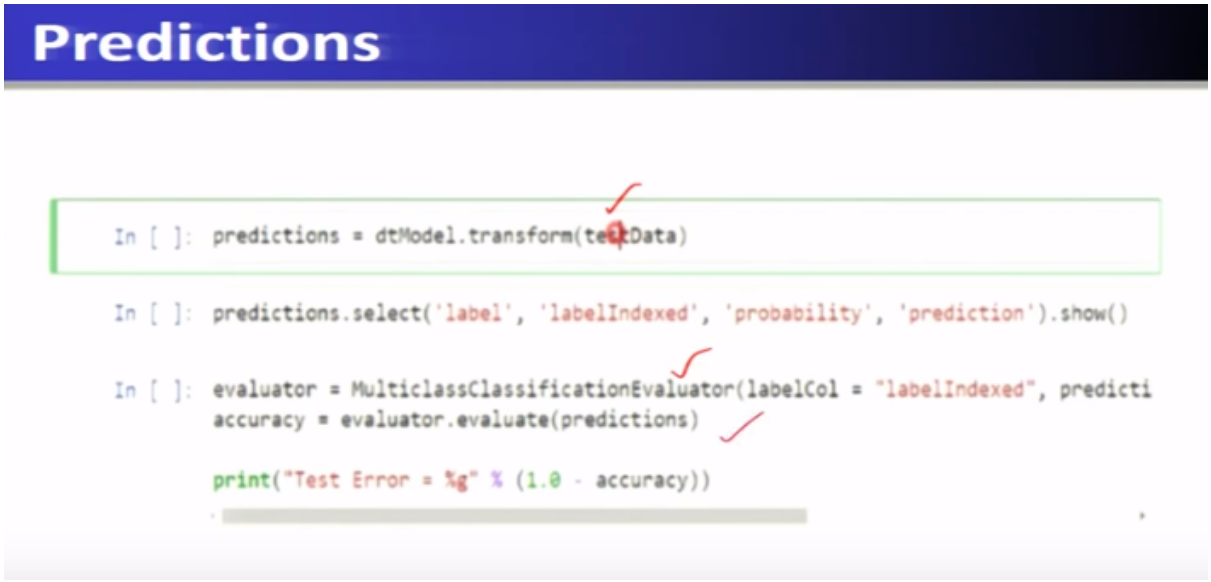
Here we can see that these predictions are shown here in this case. So, this is the decision tree, which is generated which is nothing but here you can see all the predictions, that means predictions are there at the leaf nodes and all are all other nodes are there the decision nodes are there at the internal nodes

Refer Slide Time (52:53)



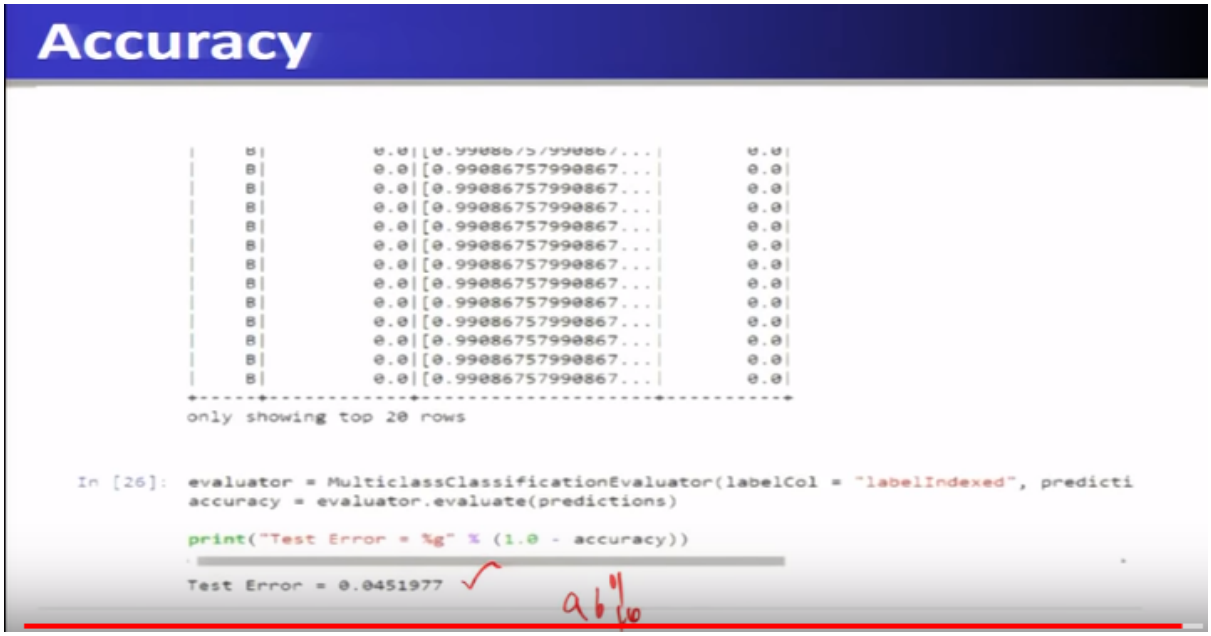
So, this is the prediction node this is also the prediction nodes, this is also the prediction node, and you see that this is the decision tree which is being calculated out of the training data set,

Refer Slide Time (53:32)



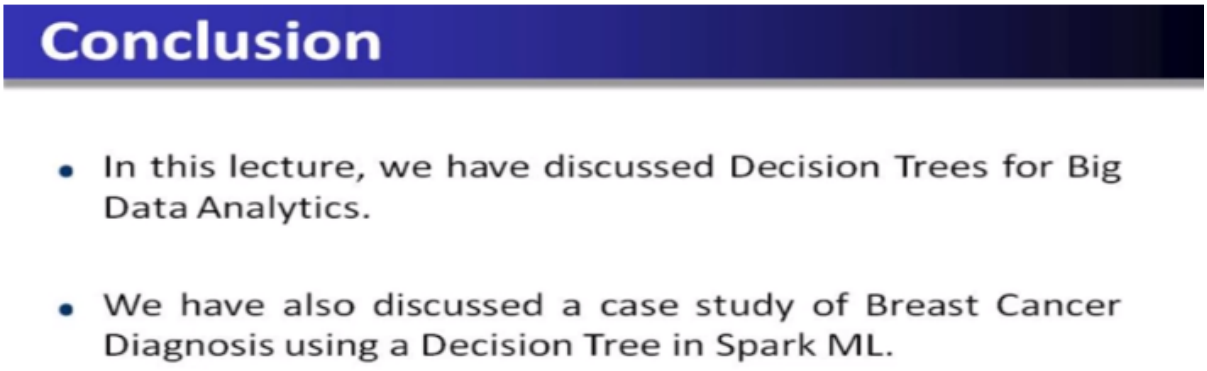
and you and using the test data set we can evaluate this particular model of a training of a decision tree which is being generated based on its prediction accuracy,

Refer Slide Time (53:45)



and we have seen that the prediction accuracy comes out to be ninety six percent which is fairly well.

Refer Slide Time (53:50)



So, conclusion in this lecture we have discussed the decision trees for our bigdata analytics, and we have also seen one case study of a medical application. Wherein we have applied the decision tree based on machine learn is part machine learning. Thank you.