Business Analytics & Data Mining Modeling Using R

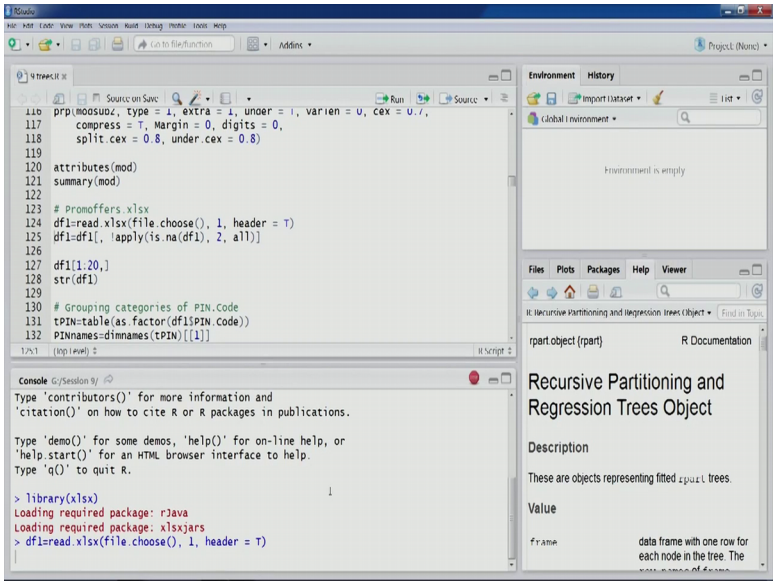
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture – 41
Classification and Regression Tress- Part VI

Welcome to the course business analytics and Data Mining Modelling Using R. So, in the previous lecture we were discussing classification trees and specifically we were doing an exercise in R. So, let us start from the point we left in the previous lecture. So, we were we were building classification tree model using the promotional offer data set. So, let us redo few points.

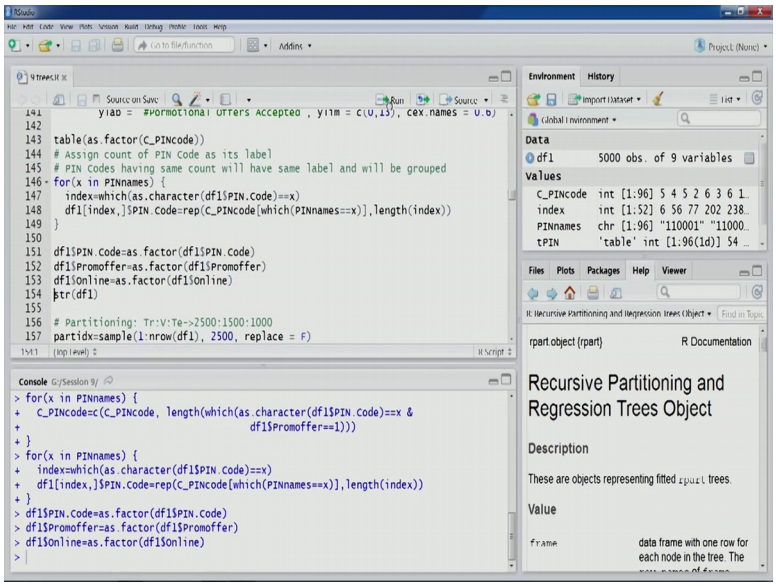
So, that we are able to resume from the same point where we left in the previous lecture. So, let us reload the data set that we used in the previous lecture for promotional offer. So, this is the data set let us import it into the R environment. So, as we saw in the previous lecture this is a quite large data set 5,000 observation. So, it will take slightly more time.

(Refer Slide Time: 01:27)



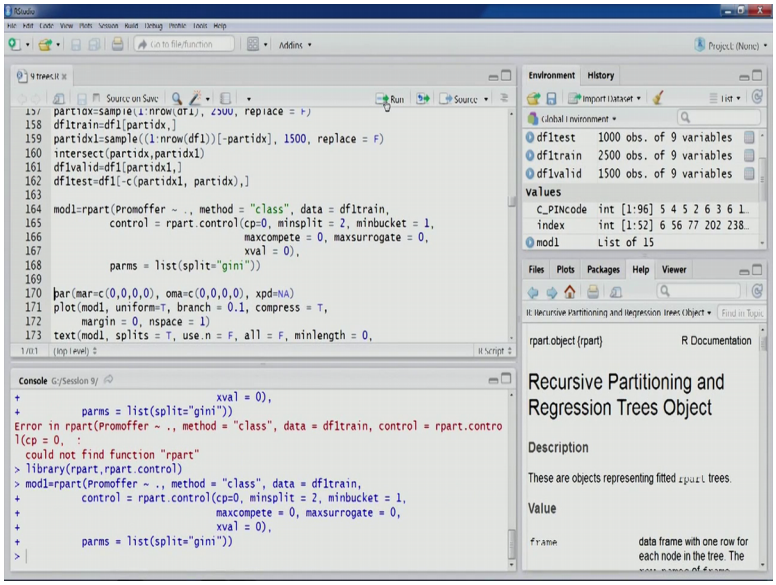
So, you also discussed about the pin code, and how we could? How we were actually using pin code? How we were grouping different categories of pin code? So, let us repeat some of those steps. So, that we are able to reach to the same point. So, pin code promotional offer.

(Refer Slide Time: 02:00)



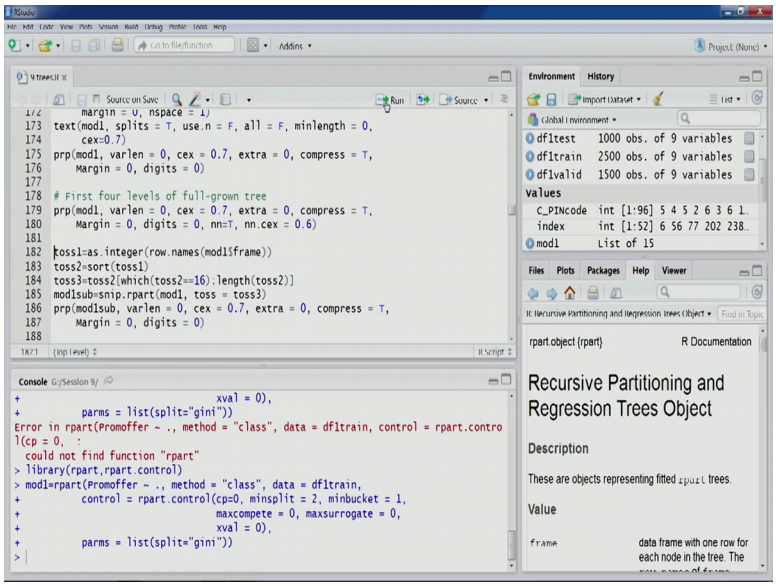
Online all these have to be converted fact available as we did in the previous lecture partitioning and then we also build the model.

(Refer Slide Time: 02:11)



We need to load this protocol library part library. So, once it is loaded, then we can build the model. We had also saw the tree diagram using a different function the plot function as well as prp function.

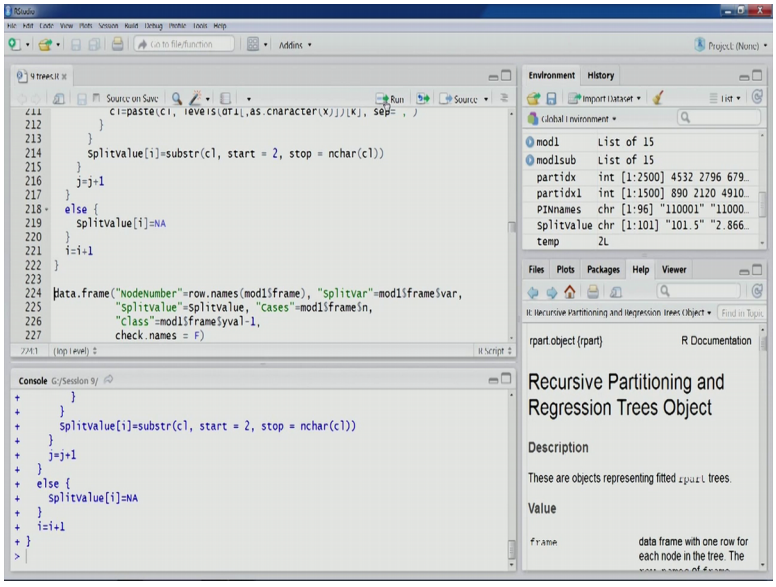
(Refer Slide Time: 02:44)



Then we started our exercise about finding the split values or finding the split values for different partitions.

So, let us go back. So, this was where we stopped. So, we were calculating the computing the split values.

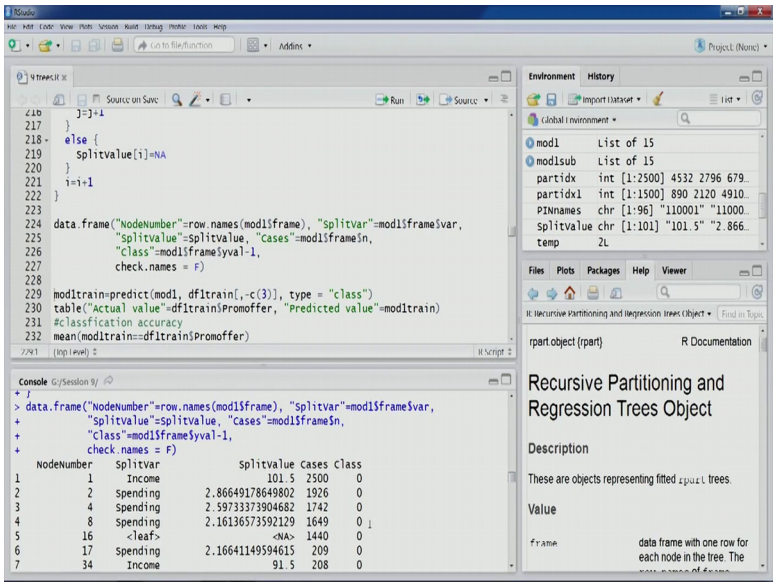
(Refer Slide Time: 03:07)



So, this discussion we had already done. So, once for every split variable the variable that were used to develop the full grown tree; once we have computed this split values extracted; rather extracted this split values for all the split variables. So, let us. So, this is

the data frame which we will a get give us some useful information. So, the previous the split value that we had computed, now we can see here the same thing which we saw in the.

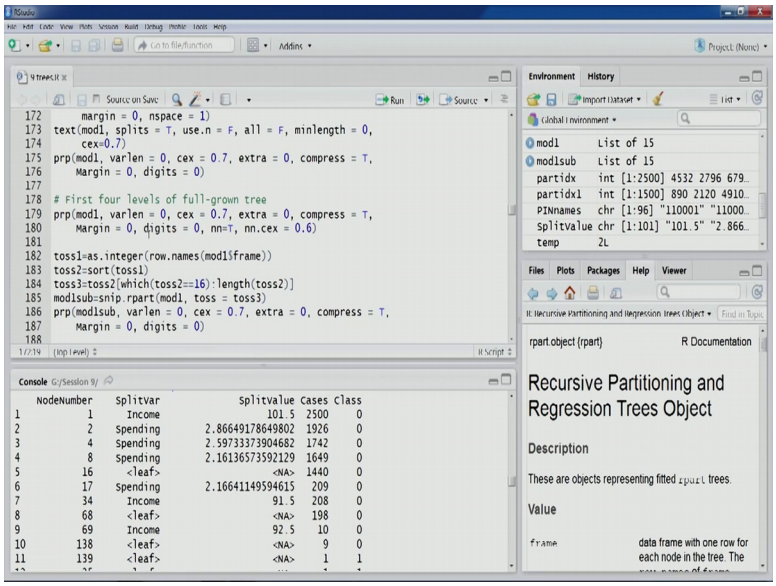
(Refer Slide Time: 03:42)



The same thing which we saw in the full grown tree and where we could actually follow the root node and what was the split variable and value combination then the other splits and the; predictor and value combination all right. So, the same thing is represented in this a tabular format in the here and, that is why be required to extract this split value. So, that we could create this tabular format you can see node number. The unique node number as we discussed about the tree diagrams, and they split variable that has been used in that particular node number.

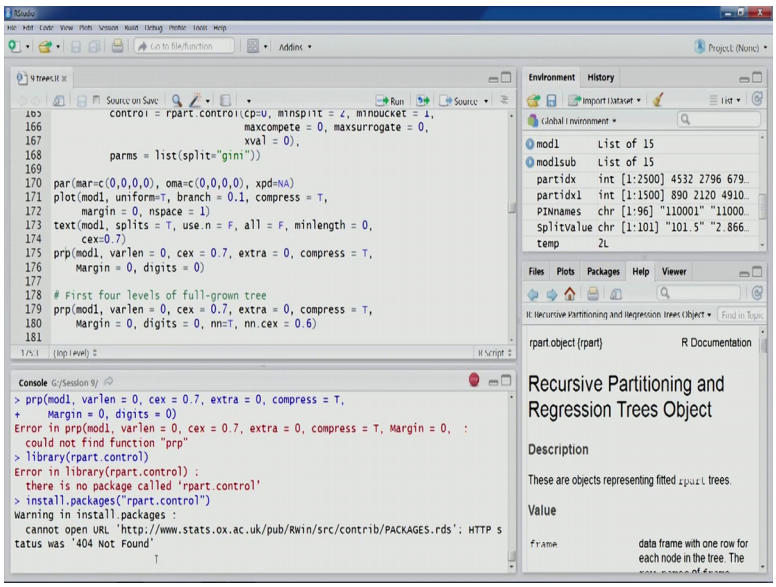
And the corresponding is split value that has been used to create a split and number of cases and the class for that particular node right. So, so, in this fashion we can look at each of the each you know different nodes that are their node number two the split variable was spending. So, this is with respect to full grown tree as we saw in the previous lecture. So, this is the split value for spending and the number of cases that are there in that particular node. So, we want to have a look at the folder diagram again. So, that was quite a.

(Refer Slide Time: 04:56)



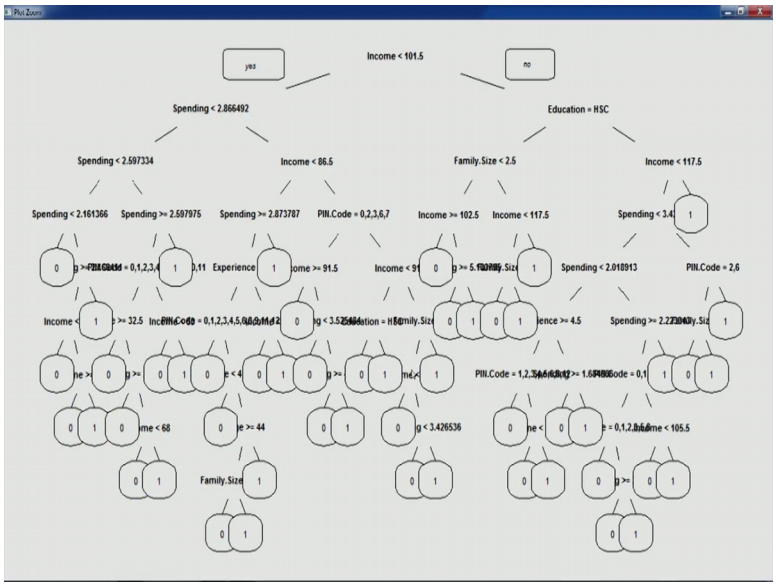
That was quite a big tree model and so, this will this prp function will give us the tree diagram.

(Refer Slide Time: 05:12)



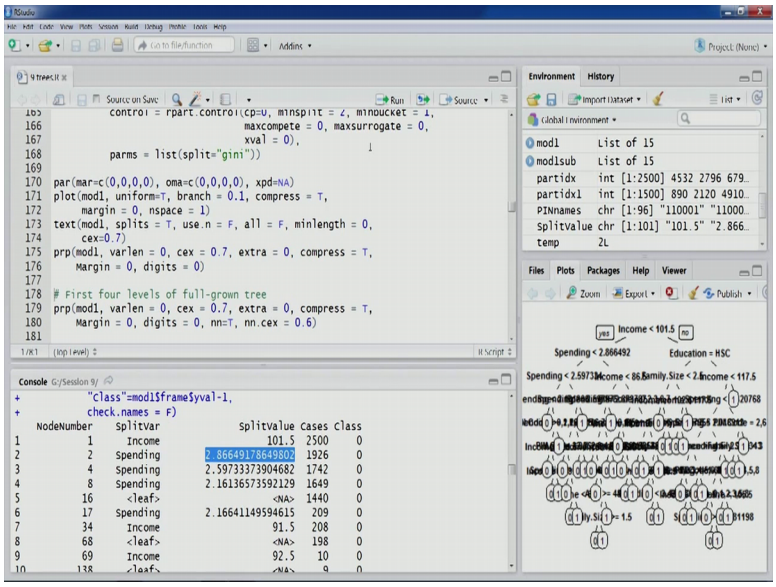
So, we need to load this particular library to be able to use the, this particular function; so once this particular package installed will be able to use rpart this function. So, let us reload the library the name was different. So, that was the problem. So, it was actually rpart dot plot. So, prp function is there. So, we have to use this. So, now, this function would be available to us. So, let us plot this you can see this was quite a big plot.

(Refer Slide Time: 06:33)



Tree diagram and the table that we had generated.

(Refer Slide Time: 06:38)



For split variable and value combination, this is the table that we are generated. Now from here and looking at the tree model you can get a better sense of, what happened in our tree diagram? So, you can see first node, root node income 1, 0 less than 101.5 and, that is the node number and the same value is here you can see in the table as well. Similarly, they say this is the second node expanding. So, the same thing is presented here you can see the value is also same.

So, the tree diagram the information that we have is specifically focusing on the split variable and split value, that we can also present in this tabular format and understand; what are the important predicted value combinations? This is especially important if we have quite a big tree we are developing a full grown tree and. So, in those situation this tabular format might be more useful. And you can see the next node number is 4. So, the 4 is will come here. So, this one is the split variable value combination is spending less than 2.16 right 2.59. So, there must be one mole this is the 1, 4 and then this is the node number 8. So, in this fashion you can see.

So, the; these are this is the table for split variable value combination. So, let us move forward. So, once we have built this model. So, let us look at the performance of this model on training parties itself; since this model has been built you know this has be this 3 model is a full grown tree. So, therefore, what we expect is that? All the observation would be correctly classified, because once we develop full grown tree.

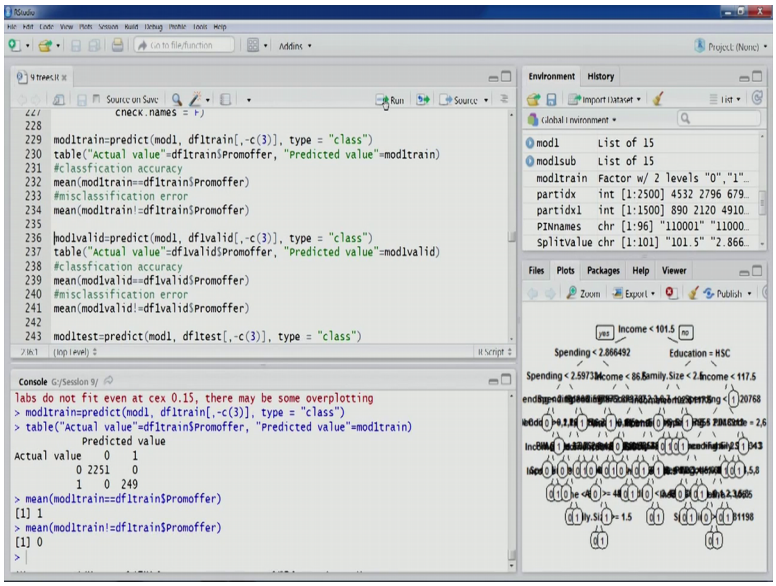
So, the; we keep on creating partitions, we keep on doing splits till we are able to reach till we are able to create pure homogeneous groups or rectangles or partitions right. So, predictor is the function as we have been using in previous techniques as well. So, this can be used to score the a particular data set. So, the model mod 1 and we are again trying to score off the training partition itself and minus c 3.

So, we are not including the; dependent variable outcome variable in this particular data set and the type is class. So, with this we can look at the performance and you would see that, this is the classification matrix you can see actual value and predicted value. So, all the 2251 actual value have been correctly classified as; correctly classified as class 0 and you can see 249 actual class one records they have been correctly classified right.

So, the if we look at the we find the classification accuracy and error you would see it is 100 percent right 1 and 0 here. So, you can see the performance is 100 percent because the tree model is all fitting the data completely. So, the whole data the data set the training partition has been completely fit with 100 percent accuracy and 0 percent error.

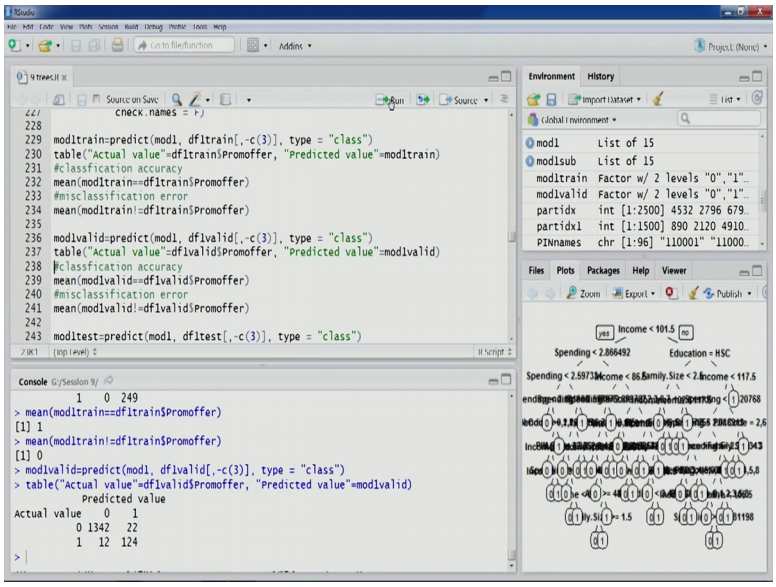
Now, let us let us look at the performance of this full grown tree model on validation partition and test partition. So, we had already we have already created these two partitions starts apply the same function.

(Refer Slide Time: 10:15)



Predict let us look at the classification matrix, now here you would see few errors.

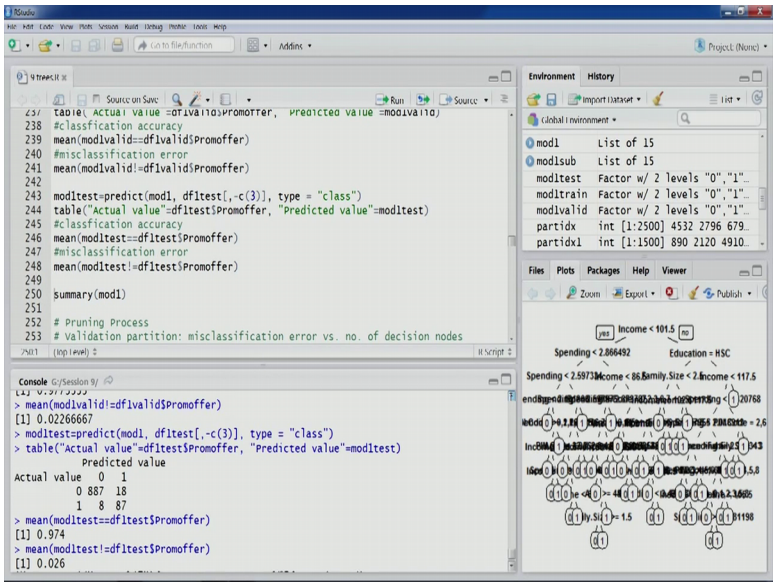
(Refer Slide Time: 10:21)



Right 22 class 0 records have been incorrectly classified as class 1 and 12 class 1 records have been incorrectly classified as class 0. So, let us look at the accuracy and misclassification error number. So, you can see 0.97. So, 97.70 percentage, that is; the accuracy and then we have 2.27 percentage that is the error right. So,. So, when we apply the model.

So, there is this much difference in the performance of the model. So, it is of course, this is expected; because it is not possible for the; the over fitted 100 percent over fitted model to perform well on new data. However, the 97 this performance is also quite good; however, this is affected that it would be lower than the training partition. Now, let us look at the test partition and the performance. Now, let us look at the accuracy and misclassification error numbers we can see 97.4 in this case.

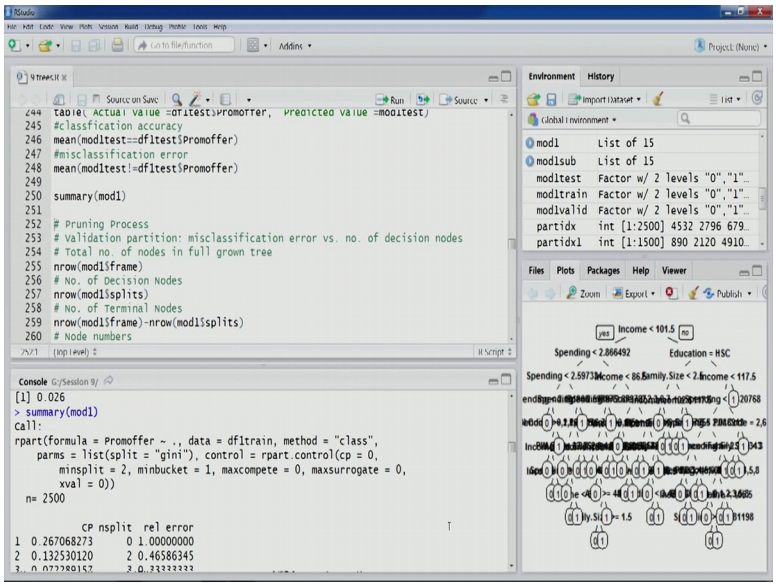
(Refer Slide Time: 11:29)



So, you if we compare from the previous performance; so would see then that in the validation partition is 97.7 and here it is that decreased 797.4. So, if we compare from the training to validation and test; so 100 to 97.7 to 97.4; so the performances though the performance is good for all the partition, but it is decreasing.

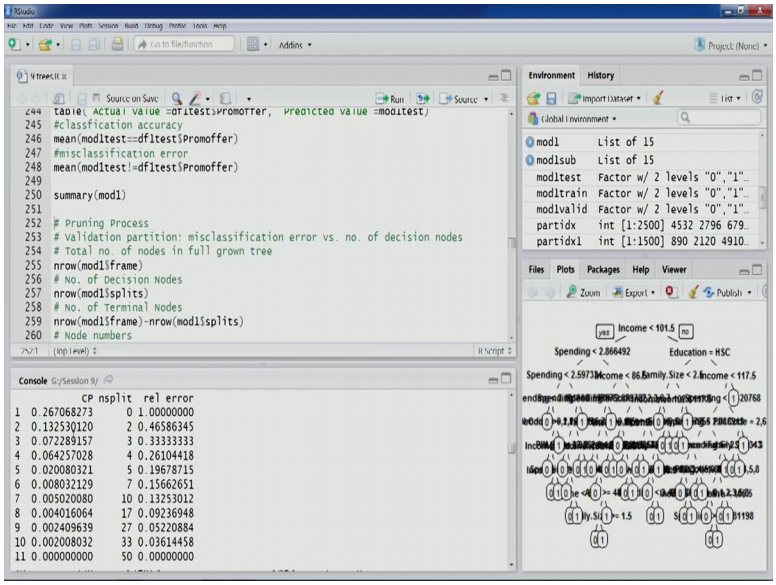
If we are interested in looking at the information; the way R output is presented. So, we can use the summary function as we have been doing for other techniques as well we can look at the details here. So, this is quite big output region being that, there are too many node numbers, because this is full grown tree. So, you can see in the summary output.

(Refer Slide Time: 12:25)



The call of this particular function the details the number of observation then this is the CP table.

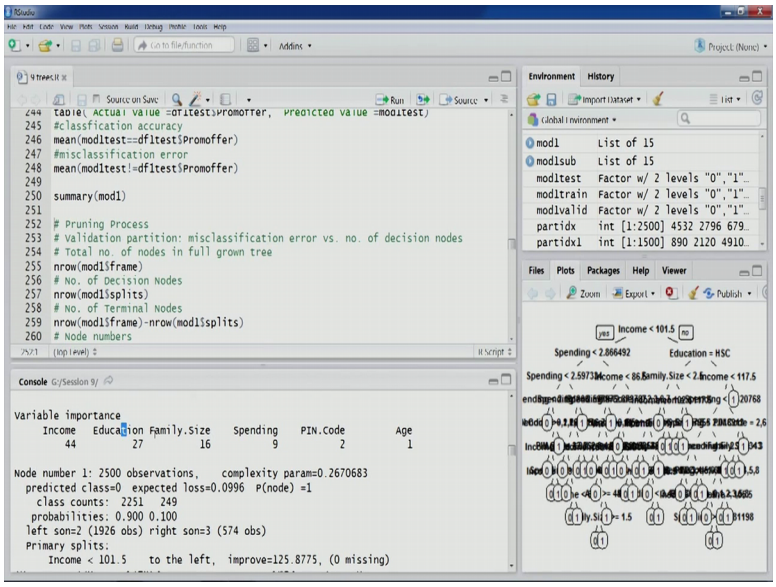
(Refer Slide Time: 12:32)



The CP table this is complexity a complexity parameter values. We will discuss this particular, what these values are about? In detail later on and split is the number of splits and then the relative error. So, I will discuss all these three all these three terms later on. Now let us look at the variable importance. So, for if we compare this to the previous data set that we had used the (Refer Time: 13:03) card data set we had just two variables

right. So, now, in this case we have you know quite a few variable here and. So, you can look at the variable importance income comes out to be the most important variable 44 percentage importance, then followed by education.

(Refer Slide Time: 13:19)



And then followed by family size, then followed by spending then pin code and age. So, from this we can also understand. So, this as we talk about that classification and regression tree can also work as a variable selection approach or dimension reduction approach. So, from here you can see that we can have a rank ordering a list of the; variables which are more important and so, from that list again we can base on about domain knowledge and expertise and looking at these numbers we can again understand which variables are important which variables to retain in the model.

Now, if we look at the splits to the partitions that are how they are being created. So, node number one we can look at this. So, some specific details complexity parameter value and other details we can clearly see predicted class expected loss and all those expected loss is nothing, but the node root node it is classified as class 0 right; because majority of the members belong to that class 0. So, what would be the a loss that is proportion of you know records which belong to the class 1 which would be incorrectly classified.

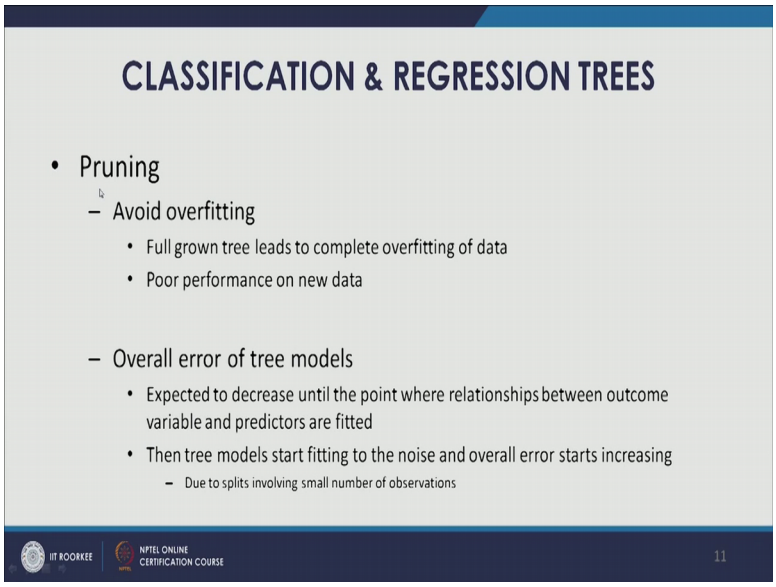
So, that is the loss here in this case node probability of this node this node probability is 1, right this being the root node. So, the class counts are also there the probabilities and

that, that is also there as you can see here you can see left son 19 1926 observation will go to the left son the right son the remaining 574 observation.

So, the same thing you can visualize in the tree diagram itself right. So, here; however, the node num node details this is not given here, because of the space problem in the tree diagram. Now, the primary split the split that has been performed is income less than 101.5 which we can see in the tree diagram as well the same information, how the improvement value that you can also see here after the split is performed. So, this is the improvement that is expected if this is split that is done right. Similarly, if you for node number two; we can look at other details 9 1926 observations and the complexity parameter value and then other details as we saw in the root node.

Now, look at this split spending less than 2.86 right improvement 3.31. So, node number 2. So, this is the node number 2. So, this split is performing in this fashion; using this predicted value combination you can look at the improvement and other things. So, similarly for all the node numbers in the summary output for all the node numbers, this detail we can find out right for all the node numbers this detail is available. So, further importance of some of these numbers that we have been talking about for example, complexity parameter and other things as we said we will discuss in the during the lecture itself. Now let us now that brings us to our next part that is pruning. So, let us go back to the slides.

(Refer Slide Time: 16:53)



So, as we have talked about we have discussed in the in some of the starting lectures of classification regression trees that there are two primary steps in cart procedures. In cart algorithm first one is recursive partitioning the second one is pruning. So, till now our discussion has revolved around the recursive partitioning, were we tried to create where we have tried to create the full grown tree we have tried to achieve your homogeneous subgroups right.

Now as we saw that this particular once we develop full grown tree it over fits the whole data and. So, how do we avoid this? So, pruning is the next step that can help us in avoiding over fitting. So, the idea is to avoid over fitting why we perform pruning? The idea is to avoid over fitting. So, full grown tree leads to complete over fitting of data as we have been saying poor performance on new data. So, because of the over fitting on new day new observation the performance will further decrease.

Now, if we if we look at the; the way tree models or in general other models that we have been talking about the focus is on typically the focus is on overall error. So, we always look to minimize overall error of classification models and also for tree models as well. So, what happens in when we try to you know minimize overall error? So, this particular error for any particular technique and also for tree models so, this is expected to decrease until the point where release relationships between outcome variables and predictors are fitted.

So, so, till the time the relationship between outcome variable and the set of predictors; that is being captured; that is being tapped in by the model. So, the till that time the overall error will continue to decrease right as we keep on you know building our model using that you know relationship using the information, that is there in those predictors. So, till the point the predictors information is being used to build the model the overall error will decrease; however, there is going to be a one point where the all the all that predictor information that could have been captured, that could be captured depending on the different strengths and weaknesses of different technique and for tree models.

As well at some point the models will start fitting to the noise once the all the predictors information; that could be useful for that classification tasks prediction task. In general, once that has been done the model will start fitting to the noise and overall error will start increasing, once we start fitting to the noise the overall error will start increasing. Now,

why specifically we talk about the tree models? Why that would happen? And in general also due to as you can see in the slide as well due to splits involving a small number of observations.

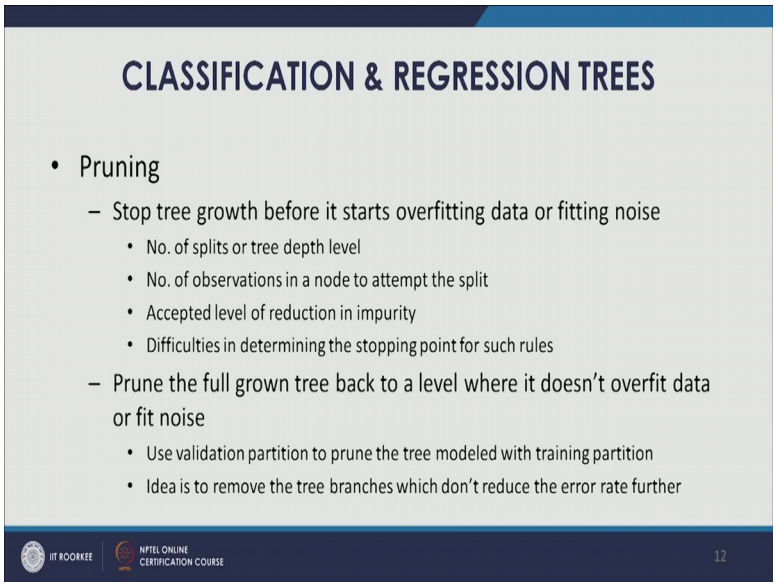
So, as we keep on doing splits using predictor value combinations. So, initially for every split will get you know much more reduction in overall error, but as we go along and most of the you know predictors information has been built into the model and once we start fitting to the noise we will also reach to the point where we are actually dealing with a small number of observation. So, we are further creating more and more splits most of the observation have already been put into their right rectangles right partitions. Now, further we are looking for few observations which are still to be correctly classified.

So, because those observations as we did through an exercise using (Refer Time: 20:51) cost data set the partition that we kept on creating using the graphic right. So, finally, we were also changing even if there is one point; that is, incorrectly classified in a particular group we also created a partition for even that one single point in that exercise right. So, you would see that essentially we are dealing with a small number of observation as we do to you know you know as we do too many splits right.

So, because of this the model tree models in general other models also they start fitting to the noise after some point once they predict information has been modelled. So, the overall error will start increase and will start fitting to the noise. So, the same thing happens in the when we develop full grown tree we keep on splitting we are we keep on splitting, we are keep on getting partition until we reach to the it will be generate pure homogeneous groups. So, some of the you know last splits some of the you know last split which have been created based on few observations. So, they are leading to they are actually fitting to the noise.

So, how do we overcome this how do we overcome this complete over fitting update of the data. So, there are two approaches. So, one is stop tree growth before it starts over fitting data or fitting noise all right.

(Refer Slide Time: 22:13)



So, we have to find out you know you know that point where the tree has stopped you know building the predictor. You stop using the predictors information to build the model, rather it has started the model as the technique has started over fitting the data or noise. So, how that can be done? So, number of is number of splits or tree depth level. So, if we can find out that optimum number of splits or tree depth level, where this actually can happen where this is actually going to happen the over fitting of data are fitting to the noise you know the number of split after which this starts to happen, if we are able to find out that number.

Then, probably we can stop tree growth there and we will get the optimized tree second approach could be number of observations in a node to attempt this split. So, how many observations could be there before and split is attempted. So, we are able to understand for a particular problem and the data set keeping everything in mind, if we are able to determine that and then also we can stop tree growth accepted a level of reduction in impurity. So, once we create split we are expecting certain reduction in the impurity the different two important impurity matrix we talked about gini index entropy measure. So, what is the level of reduction that we expect once we create this one split once we create partition and different subgroups.

So, if we are able to determine a; you know accepted level of you know impurity reaction in impurity. So, that can also be used to stop tree growth. So, this is another

approach now these three things that we talked about, whether it is number of a splits that can be performed optimized number of splits or optimal number of observation in a node to attend the split or the accepted level of reduction in impurity if. So, you would see with all these three you know approaches it is difficult to determine these points and how do we determine the number of splits or number of observation.

It is slightly difficult more number of experimentations would have to be performed right, it is going to take lot of time to build the model and to fill find the and the optimum you know these points and optimum values of some of these approaches. So, because of this difficulty this particular approach is you know not that popular and there is another approach; that is, generally tried which is actually the pruning approach that is the focus of this discussion this particular exercise also.

So, prune the second approach is prune the full grown tree back to a level where it does not over fit or fit noise. So, that is the second approach. So, once first we can develop the full grown tree and then we start pruning it back to the level where it does not over fit data or fit noise. So, that is one approach how it can be performed we can use validation partition. So, we can you know build our tree we can construct our tree using the training partition and, then we can start pruning it back using the validation partition.

So, this is one particular technique. So, we talked about in the starting you know lectures of this course, we talked about the importance of partitioning the training partition validation partition and also the test partition, but if we look at the some of the techniques that we have tried we have just been creating two partitions training partition and test partition.

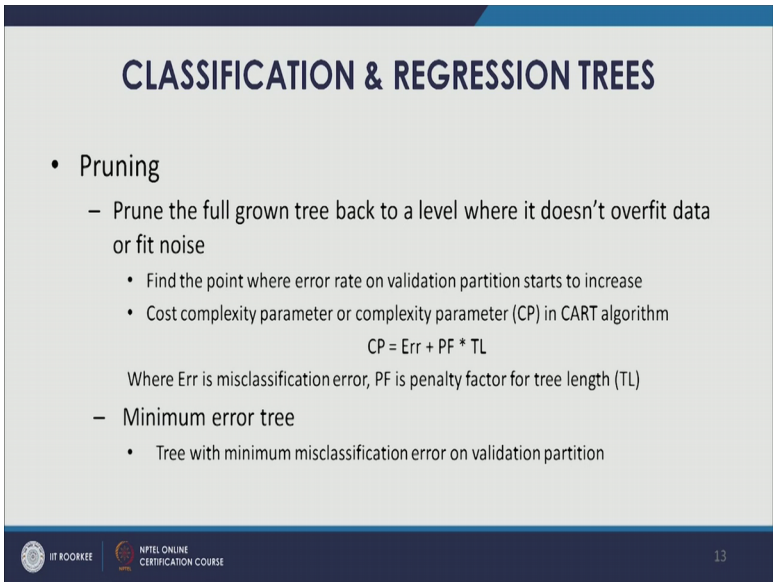
Now, in this particular technique you would see the importance of validation partition training partition is being used to build the tree that is full grown tree validation partition is being used to define, the model to fine tune the model how to prune back the tree to the level where it is stops fitting to the noise or stops over fitting to the data. So, validation partition also is going to be in a way part of the modelling process. And therefore, the importance of test partition which would act as a new data partition and where we can apply our model and check it is performance.

So, in this particular technique you we see that the training partition and the validation partition they both become part of the modelling process. So, in this approach the

pruning approach; that we just discussed the idea is to remove the tree branches. When we talk about removing the; you know pruning that we back to a certain level. So, the idea is to remove the tree branches which do not reduce the error rate further right. So, the error rate is not you know some of the branches which are not able to decrease the error the overall error then probably we should remove those branches.

So, that is the main idea. So, so, to implement this particular approaches we need to we need to be able to understand to identify or determine the branches or the nodes which are not in decreasing the overall error further and then remove them off slip them off so with this so, this so, a few other things in pruning. So, as we talked about.

(Refer Slide Time: 27:54)



I find the point where error rate on validation partition starts to increase. So, we talked about the point where you know construct the full grown tree and use the using retaining partition and the prune impact using the validation partition.

So, first we need to find out the point where error rate on validations we can build the model using partition and then apply, then a score the validation partition and check the error rate and find out at which point the error rate starts to increase; that is, the point that is the level at which till which we have to prune the tree. So, another concept related to the; this pruning is cost complexity parameter or complexity parameter CP; that is that is generally used in cart algorithm.

So, the complexity parameter values are typically used to perform the pruning in many implementations of cart. So, what is a complexity parameter? So, if you look at the definition here this complexity parameter that is CP is error; that is nothing, but the overall error that we talk talking about the misclassification error. So, CP is error plus PF into TL. PF is the penalty factor and TL is the tree length or tree size.

So, for tree size we introduce a penalty factor and error plus this penalty factor. So, that will give us the CP value. So, you know if there are two particular tree models two candidate tree models and they both have the same misclassification error, but if particular you know they have both these tree models have different tree sizes different number of nodes.

So, it is the smaller tree model that would be selected that would be based on the complexity parameter, because there is going to be penalty on this on three sides. So, the model having bigger tree length or high figure tea tree size would have additional you know additional value additional value added to this error component and therefore, it is error the CP value will would be more than the tree having similar having same misclassification error, but of a smaller size.

So, (Refer Time: 30:24) parameter or cost complexity parameter in a sense can be used to control the size of the tree to control the length of the tree. So, a complexity parameter can also be used to prune back the tree and as we have talked about that validation partition; that can be also used error rate and we can also identify from, where it starts to increase or the second approach is use the compute the complexity values and complexity parameter values and use that to find out the optimum length of the tree now.

So, we will stop at this point. And we will continue our discussion on pruning in the next lecture.

Thank you.