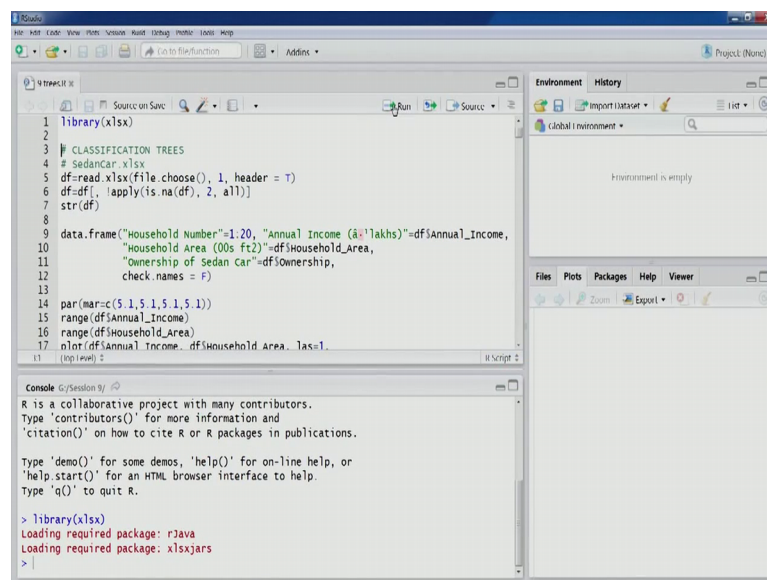


Business Analytics & Data Mining Modeling Using R
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture - 39
Classification and Regression Trees- Part IV

Welcome to the course Business Analytics and Data Mining Modelling using R. So, in the previous lecture we were discussing classification and regression trees. In particular we were doing an exercise in R using a particular data sets sedan car data set that we have been using. So, let us do few of the steps. So, that we are able to resume from the same point, where we ended in the last lecture.

(Refer Slide Time: 00:49)

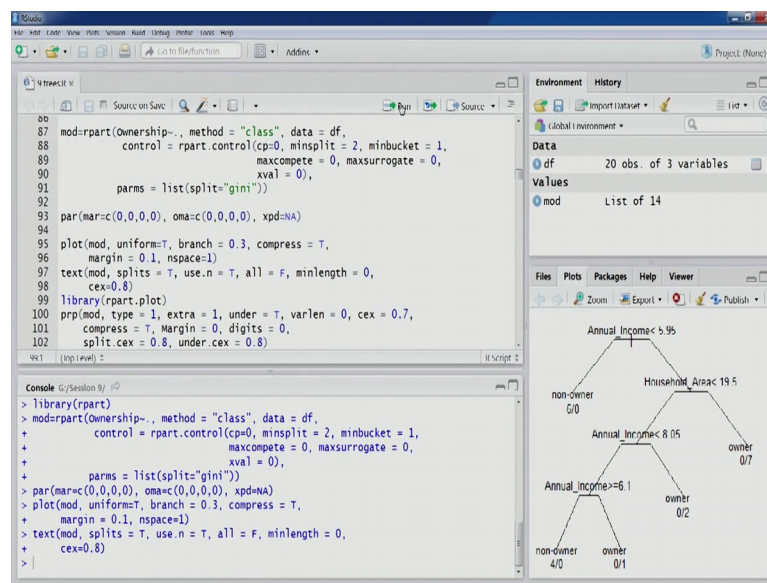


```
1 library(xlsx)
2
3 # CLASSIFICATION TREES
4 # SedanCar.xlsx
5 df=read.xlsx(file.choose(), 1, header = T)
6 df=df[, !apply(is.na(df), 2, all)]
7 str(df)
8
9 data.frame("Household Number"=1:20, "Annual Income (â" lakhs)"=df$Annual_Income,
10           "Household Area (00s ft2)"=df$Household_Area,
11           "Ownership of Sedan Car"=df$Ownership,
12           check.names = F)
13
14 par(mar=c(5,1,5,1,5,1))
15 range(df$Annual_Income)
16 range(df$Household_Area)
17 plot(df$Annual_Income, df$Household_Area, las=1,
18      (top row))
```

The screenshot shows the RStudio interface. The script editor on the left contains the R code for loading the xlsx library and reading a file. The console at the bottom shows the output of the library loading, including messages about required packages like rJava and xlsxjars. The environment pane on the right is empty, indicating that no objects have been loaded into the environment yet.

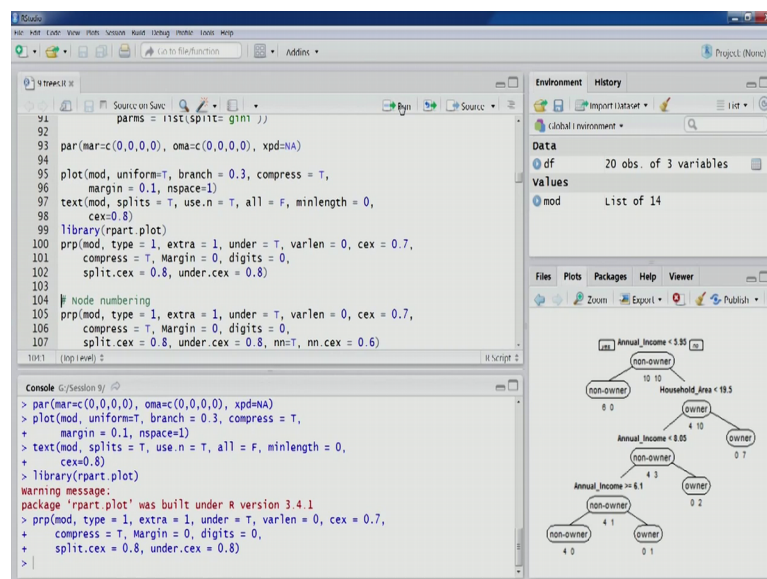
So, let us load the this library let us import the data set, quickly. Now in the last lecture we were we were able to create the model, for this particular data set. So, let us come to that point yes. So, this is the part. So, these were the variables. So, let us reload this library R part

(Refer Slide Time: 01:31)



So, R part is the function that we used last time. So, let us build the model and the plot as well. So, this was the plot that we had created last time, and we also talked about much a nicer version or pretty version of the plot or tree model.

(Refer Slide Time: 01:47)

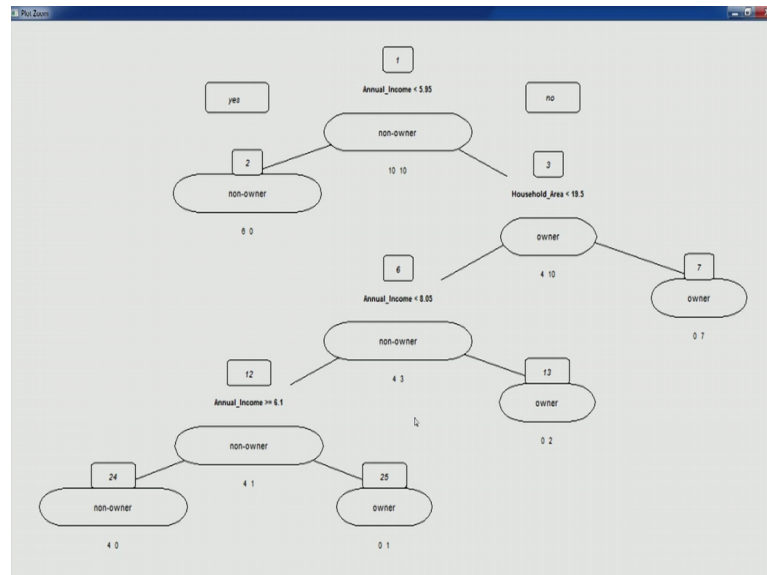


This can we talked about the node numbering, and the requirement of node numbering, if we want to snip off sum up some particular part of the tree right.

So, why we need this it will be more clear as we go on we go along, discuss more in this particular lecture. So, node numbering as we talked about in the previous lecture this n n

argument, we have to specify `nn` as `true`. So, node number would be assigned, and then the character expansion this parameter is also can be used

(Refer Slide Time: 02:26)



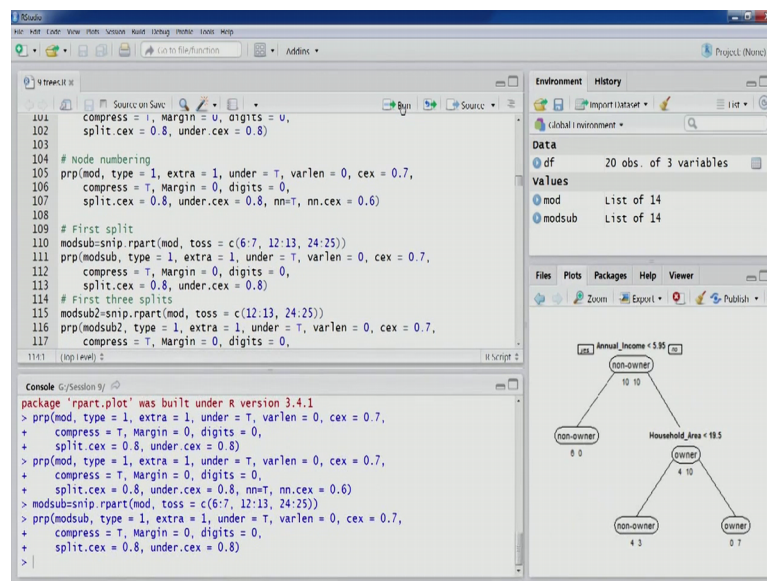
So, once node numbering is done you can see in the plots, you can see in this is the full grown tree using the full data set that we have, and node numbering has also been done. So, in the previous lecture we had done first split, where we just wanted to have the nodes, which we nodes till the point of first split right. So, this is the first split. So, the we would like to have the nodes you know just 1 node, and then the other you know terminal nodes right.

So, in that case as we talked about that we have to from the these node numbers, we have to see if we want to keep these 3 nodes to re-modify our example, we want to just keep these 3 nodes this first decision node then this terminal node, and then other one, this 1 also be decision note and followed by terminal nodes. So, we want to keep the you know you know these 2 decision nodes, and the corresponding terminal nodes as well, then we will have to a snip off you know nodes starting from here 6 7, we talked about the unique node numbering scheme that we have entry models in general and tree algorithms also

So, we have to specify. So, this is the function snipped `r` part. So, `snip dot r` part function will allow us to remove some of the you know unwanted nodes. So, we just want to look focus on the you know first level or first few nodes, then we can remove this the other part other nodes using this `snip r` part function.

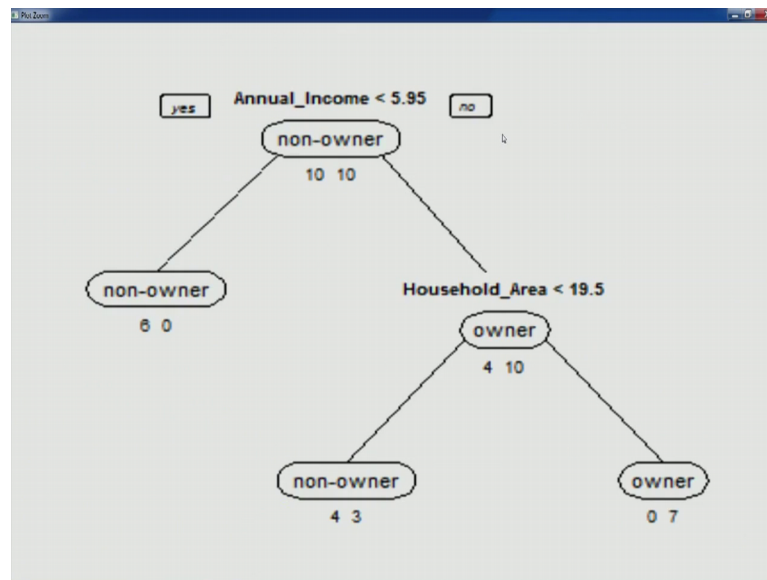
First argument is as we talked about is going to be the model that is `mod` that this object `rpart` part object, and then the second argument is where we specify the node numbers which we want to get rid of. So, you can see 6 and 7 12 and 13 24 and 25. So, these are the node numbers that we want to get rid off. So, we can run this particular function, and you would see that new model you know the sub tree model has been saved in `modsub`. And now we can print this particular tree model.

(Refer Slide Time: 04:41)



Now, you can see this is the new model rather snip snipped off version of the same model right.

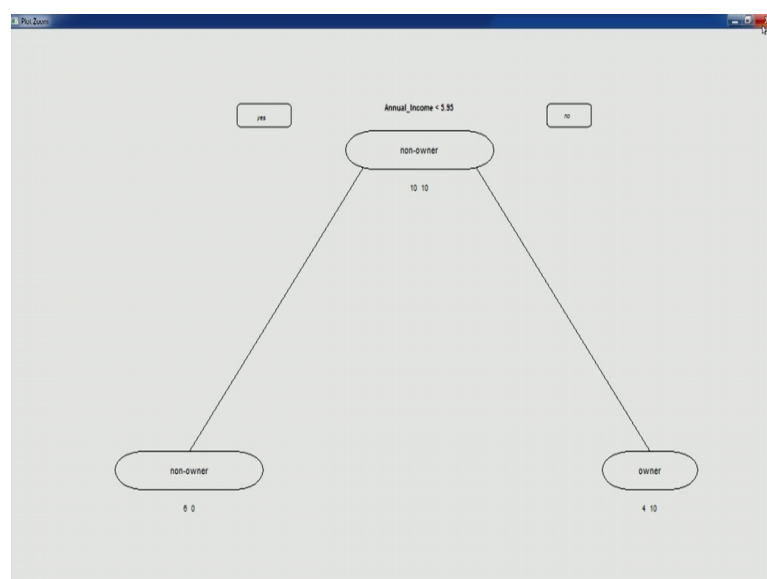
(Refer Slide Time: 04:49)



So, you can see 2 decision nodes here, and the corresponding terminal nodes as well we want to keep just 1 node. So, you know the first split that we talked about. So, we can remove this 1 as well

So, then we will end up with the in just 1 node. So, let us say in the previous this was the full tree right so, in this full tree so, if we will have to remove 3 as well. So, if we put 3 here, we also pass on 3 node number 3 also as the argument here, and then we do our sniping then this we will also be removed, and the new plot would have just 1 root node

(Refer Slide Time: 05:40)



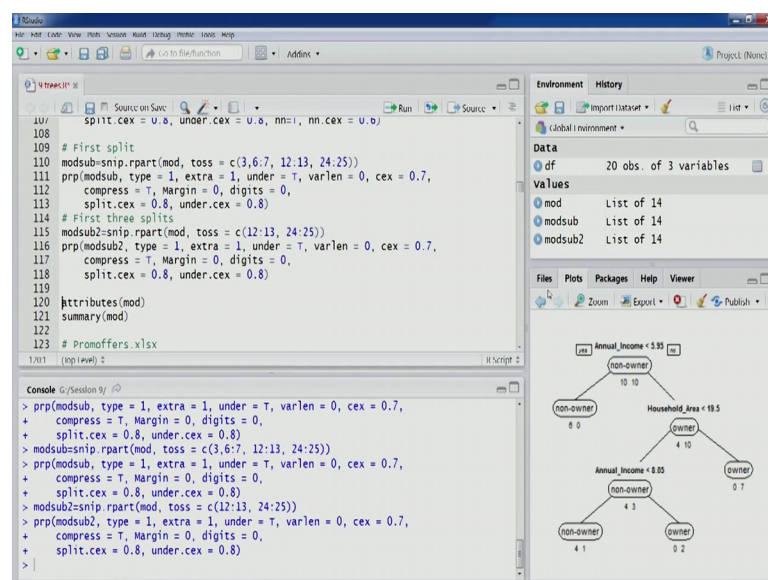
You would see just 1, you know decision node that is the root node, and that is also indicative of the first split, and the others right there would be now as you can see these are the terminal nodes where we get the you know final classification.

So, in this fashion depending on the because as we talked about the full grown tree could be large, and you know if the data set is quite large, and then there are more number of variables that we will see in through 1 more exercise 1 more example, data set that the whole full grown tree could be quite messy and difficult to understand

So, therefore, this exercise will help us in just displaying or plotting the first few levels so, that we can look at the some of the most important variables that are being used to create splits or to create partition. So, similarly if we wanted to have a 3 splits, so as we talked about we can go back to the full grown tree

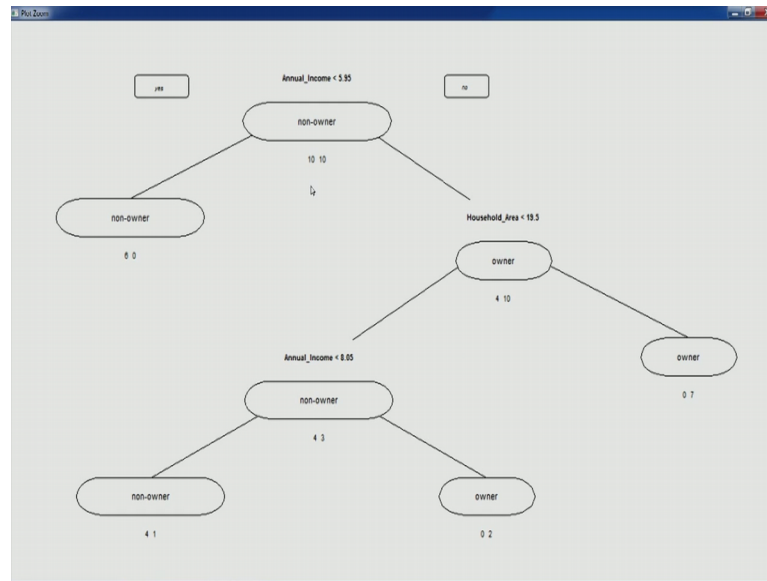
So, this was the tree. So, in this if we want to have this you know 3 splits. So, split 1 then 2 and 3. So, therefore, we will have to remove the other nodes right so, let us go back. So, other nodes would be 12 and 13 and 24, 25. So, we can get rid off these nodes right 12 13. So, appropriately mentioned here so, we can get new sub tree, and then we can plant it.

(Refer Slide Time: 07:14)



Now, you can see, and this new 1 here just 1 2 and 3 3 splits and the corresponding terminal nodes as well.

(Refer Slide Time: 07:21)



So, in this fashion depending on the levels we can snip the tree full grown tree, and the importance of this particular process will realize when we discuss further. We are interested in different attributes that this R part object, R part model object you know actually it is carrying. So, that we can you know do using attributes command. So, this particular is generally command applicable to other techniques as well. So, different models that we have been building in previous lectures while discussing techniques.

(Refer Slide Time: 08:09)

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for creating a recursive partitioning model using `rpart()`. The code includes comments and the following lines:


```
modsub2=snip.rpart(mod, toss = c(3,6,7, 12,13, 24,25))
prp(modsub2, type = 1, extra = 1, under = T, varlen = 0, cex = 0.7,
    compress = T, Margin = 0, digits = 0,
    split.cex = 0.8, under.cex = 0.8)

# First three splits
modsub2=snip.rpart(mod, toss = c(12,13, 24,25))
prp(modsub2, type = 1, extra = 1, under = T, varlen = 0, cex = 0.7,
    compress = T, Margin = 0, digits = 0,
    split.cex = 0.8, under.cex = 0.8)

attributes(mod)
summary(mod)

# Promofers.xlsx
df1=read.xlsx(file.choose(), 1, header = T)
```
- Console:** Shows the output of the `rpart()` function call:


```
> compress = T, Margin = 0, digits = 0,
+   split.cex = 0.8, under.cex = 0.8)
> attributes(mod)
$names
 [1] "frame"      "where"      "call"
 [4] "terms"      "cptable"    "method"
 [7] "parms"      "control"    "functions"
[10] "numresp"    "splits"     "variable.importance"
[13] "y"          "ordered"

$levels
named list()
```
- Environment:** Shows the global environment with 20 observations and 3 variables. The variables listed are `mod`, `modsub`, and `modsub2`, each of type `list`.
- History:** Shows the execution of `rpartObject`.
- Help:** The `rpart` package documentation is open, showing the title "Recursive Partitioning and Regression Trees Object".

So, there also the attributes function can also be used to find out, the different information different you know results and information that is stored in the mod object.

So, this is r part object, and you can see information that is there. So, more details on what kind of what these you know frame is about what these other you know variables are about, you can always go to the help section, and you can always find out more about the r part object, and if you look at the r part object in the help section, and you would see frame first value frame you would see then second value

So, the mode frame is quite big you know it is actually a data frame. So, containing lots of information, then where call and other attributes that you can see here, they can be easily seen and understood. Summary mod function is also there. So, we are interested in looking at the summary. So, what summary function again generic function what it contains is we call the CP table which will discuss again, later in the discussion also.

(Refer Slide Time: 09:22)

```

110 modsub=snp.rpart(mod, toss = c(3,0,7, 12,13, 24,25))
111 prp(modsub, type = 1, extra = 1, under = T, varlen = 0, cex = 0.7,
112 compress = T, Margin = 0, digits = 0,
113 split.cex = 0.8, under.cex = 0.8)
114 # First three splits
115 modsub2=snp.rpart(mod, toss = c(12,13, 24,25))
116 prp(modsub2, type = 1, extra = 1, under = T, varlen = 0, cex = 0.7,
117 compress = T, Margin = 0, digits = 0,
118 split.cex = 0.8, under.cex = 0.8)
119
120 attributes(mod)
121 summary(mod)
122
123 # Promoters.xlsx
124 df1=read.xlsx(File.choose(), 1, header = T)
125 df1=df1[, !apply(is.na(df1), 2, all)]
126

```

Console Output:

```

n = 20

CP nsplit rel error
1 0.60 0 1.0
2 0.15 1 0.4
3 0.10 3 0.1
4 0.00 4 0.0

Variable importance
Annual_Income Household_Area
77 23

```

But since we are in the r part dot object you know page, you can see that there is another description or CP CP table a matrix of information on the optimal pruning based on a comp complexity parameter. So, that is generally displayed in CP table.

Let me also tell you that r part control function specific function that we talked about in the previous lecture right. So, if we open the page for r part control, you would see there

is 1 specific argument for `x_val` that is number of cross validation. So, the `r` part function that we have in this particular package, it also does some cross validation

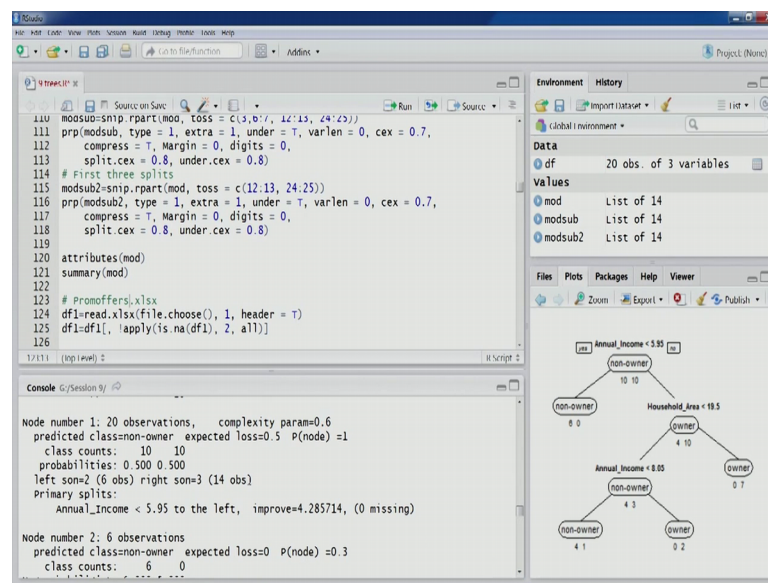
So, some of the as we talked about in the previous lecture, that the some of these observations are reserved for cross validation, within the training partition within the data that is supplied to `r` part function, and that is then later on used to produce the CP table and other information

So, we will discuss them as we go along. So, the coming back to the summary output you would see that later on you would see the variable importance is also mentioned here. So, classification and regression tree you know 1 aspect of with this particular technique is, it can also serve as a variable selection technique or dimension reduction technique. So, some of them we discussed before for example, without we did we have already discussed dimension reduction technique, and variables you know regression based variable selection approaches right.

So, the classification and regression tree is also 1 of those techniques, which help us helps us in terms of identifying the more important variables more important predictors right. So, in this case you can see variable importance is also part of the output annual income is you know 77 percent importance, and household area is remaining 23 part 23 percent

If we had more number of variables, we would have got much larger much bigger list. So, in terms of from this we can say in terms of identifying, in terms of classifying the sedan car ownership, annual income is more important than household area. The same thing you would see is also reflected in the plots as well you can see the first split is based on annual income, then in the summary output that we were discussed discussing right

(Refer Slide Time: 12:02)



So, for every node some information about the split that has been performed on that node, and more information on that node also is always going to be available, and the summary outputs you can see node number 120 observations. So, you can see in the let us go back to the full plot as well.

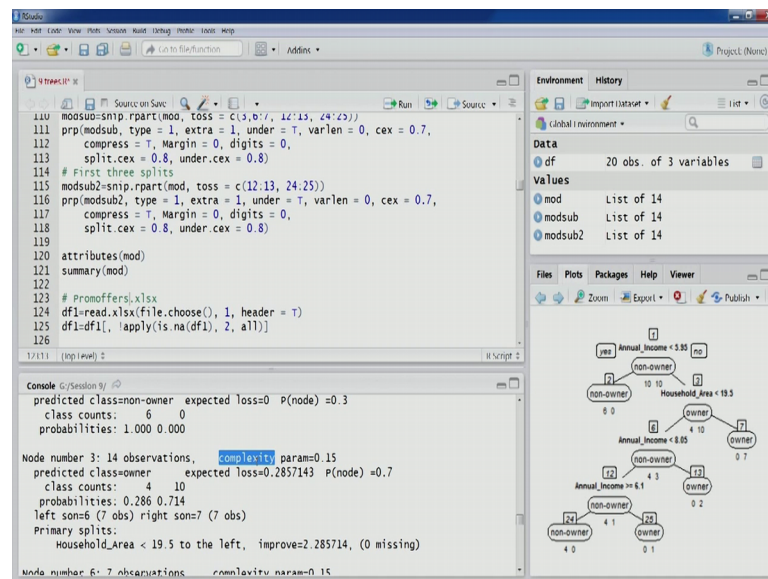
So, this was a full plot. So, you can see 20 observation in the root node all 20, 10 and 10 class counts were also 10 and 10, you would see the expected loss that this value is also there all right, how many observation in the lefts and, how many observers and the rights, and left child or right child so, different terminologies are there. So, you can see 6 observation in this particular node let us zoom in. So, you can see in the left child this is also terminal node, and you can see 6 observations here, and you would see that right son it has fourteen observations right node number 3

So, you can see node number 3 and 14 observation we will have to combine all these 4, 1, 5, 7 and 7 14. So, in this right child is actually a quite you know in it is own it is a sub tree. So, within this sub tree you can see 14 observations are there 6 and 14. So, that was the partition at root node.

So, split is done on based on this particular classification rules. So, that also you can see the improvement value is also there. Now if we if we scroll down further we will if we scroll down further will also see that other nodes node number 2, node number 3 right so, there are also 6 observations 14 observations you know this was terminal node

So, there no. So, this was no split was done here as you can see the information there in the node number 3 the split was done here you can see the split rules classification rule is also mentioned, and the left son number observation returns. So, in this fashion we can always find out the details about these scripts that have been papa performed by r part function.

(Refer Slide Time: 14:13)



(Refer Slide Time: 14:51)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	Income	Spending	Promotion	Age	PIN Code	Experience	Family Size	Education	Online													
1	49	1.60	0	25	110057	1	4	HSC	0													
2	35	2.20	0	45	110092	18	3	HSC	0													
3	10	0.50	0	39	110036	15	1	HSC	0													
4	101	2.71	0	35	110095	9	1	Grad	0													
5	45	1.00	0	35	110081	8	4	Grad	0													
6	21	1.15	0	27	110096	12	4	Grad	1													
7	71	1.03	0	53	110090	27	2	Grad	1													
8	23	0.78	0	50	110017	24	1	PostGrad	0													
9	80	0.28	0	35	110093	10	3	Grad	1													
10	182	9.88	1	31	110053	9	1	PostGrad	0													
11	108	2.57	0	45	110052	39	4	PostGrad	0													
12	47	0.65	0	29	110039	5	3	Grad	1													
13	111	3.80	0	48	110055	23	2	PostGrad	0													
14	41	3.14	0	59	110026	32	4	Grad	1													
15	113	2.22	0	67	110015	41	1	HSC	0													
16	23	2.03	0	40	110039	30	1	PostGrad	1													
17	128	1.18	1	38	110010	14	4	PostGrad	0													
18	80	2.14	0	42	110060	18	4	HSC	0													
19	191	7.34	1	46	110086	21	2	PostGrad	0													
20	22	0.53	0	55	110059	28	1	Grad	0													
21	23	0.60	0	56	110039	31	4	Grad	1													
22	45	2.55	0	27	110018	27	3	PostGrad	1													
23	62	1.20	0	29	110009	5	1	HSC	1													
24	41	0.62	0	44	110046	18	2	HSC	0													
25	152	3.90	0	36	110017	11	2	HSC	0													
26	29	0.50	0	43	110008	19	3	HSC	1													
27	83	0.20	0	40	110042	16	4	PostGrad	0													
28	157	1.67	0	46	110080	20	1	HSC	1													
29	49	2.58	0	56	110053	30	1	PostGrad	1													
30	120	3.53	1	38	110019	13	1	Grad	1													
31	36	1.96	0	59	110094	25	1	PostGrad	1													
32	30	2.97	0	40	110020	16	1	Grad	1													
33	47	1.64	0	45	110041	18	3	BaccGrad	0													

We had used in a previous lecture where we had income spending and 3rd variable from offer. So, where we based on the income and spending, we were we use the this particular for classification all other visual techniques we use these 3 variables. Now this is full data set where we have information on income spending, and the promotion offer is our outcome variable of interest, and then we have information on age pin code, experience family size, education on then online activity status.

So, this is particular this particular information is about you know a particular form with it finding out you know we are trying to build a classifier, whether if a particular whether particular customer is going to respond to their promotional offer or not. So, the promotional offer that particular column that particular variable being the outcome variable, and using the other information other predictors other variables that we have in the data set, they will play the role of predictors income spending age pin code and others

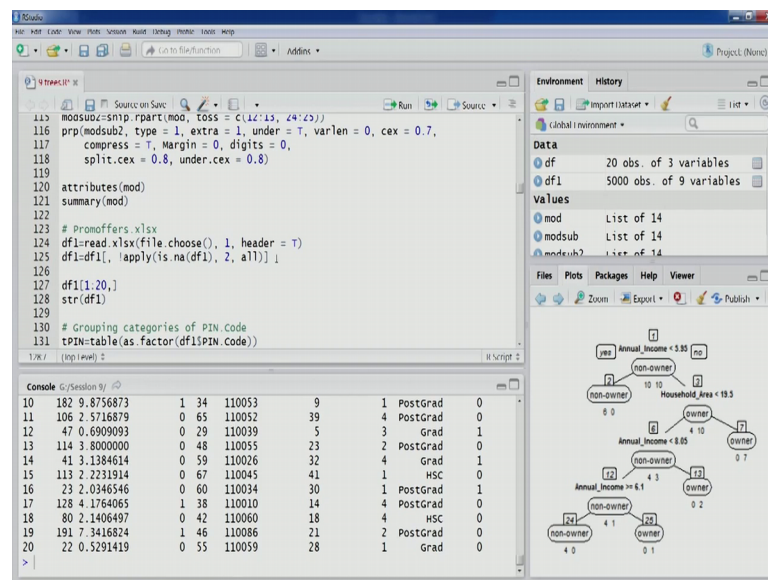
So, using these variables will try to classify a model where we build a classifier where we would like to classify whether particular observation, whether a particular customer is going to respond to the promotional offer or not. So, other variables age is again age of the customer with the responder pin code is the location of that particular customer experience this is the actually the professional experience that you know that of that particular customer, family size of that customer and then we have education and

whether it is twelfth pass h s c or graduate or postgraduate. So, we have some information about the education, and then the online activity. So, whether the customer is online active or not so, 1 representing 1 indicating that the customer is online active, and 0 indicating that the customer is not active online.

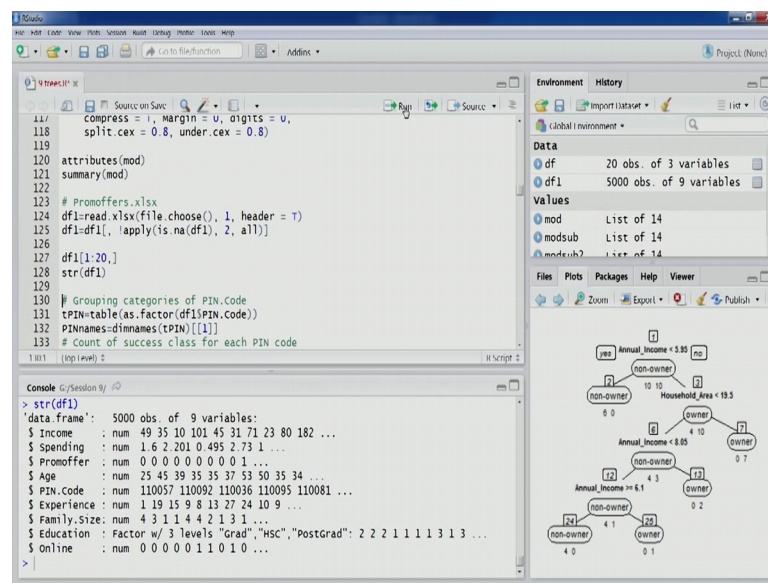
So, with this information we would like to build a classifier to predict the whether the customer is going to accept the promotional offer or not. So, with this background let us come back to R studio, and we will start our classification tree model. So, first let us import this particular dataset, since there are 5000 observations. So, R will takes like bit more time to import this particular file. So, as we have talked about that once the data is imported that it is actually stored in the memory. So, therefore, it takes a bit more time to bring all the observation into memory and therefore, now this is done. So, you can see in the environment section d of 1 data frame 1 5000 observations of 9 variables.

So, let us remove any columns if there are any so, there was none. So, first 20 observation you want to have a look again, you can do this using subset using brackets.

(Refer Slide Time: 18:07)



(Refer Slide Time: 18:22)



can model this particular variable, as a numeric variable right, because the location can also be though even though this is a numeric code, but because the number of categories are many, and location can also be associated with the latitude longitude and therefore, latitude and longitude which are numeric in nature

And these codes in that sense can represent latitude, and longitude, because generally when these codes are assigned. So, there is some method there is some you know some process that is adopted to assign these codes. So, therefore, they might in a way represent the latitude longitude and the they might you know in essence might be the ordinal variable

So, therefore, there are too many categories they can also be treated as a numeric variable however, in this particular exercise that we are going to perform, we will treat them as categorical variable, and I will see how we are able to reduce a large number of categories into few groups few number of categories, and then use them as a factor variable

So, we will see that then the next variable is experience which is numeric. So, just fine family sized numeric which is also fine, education is factor 3 levels so, appropriately mentioned again. So, no need to change then online, So, this is mentioned as numeric, but we will have to change it whether it is factor variable with 2 categories active or non active. So, let us start with our pin code variable

So, what will what we plan to do is grouping will try to group the categories of pin code. So, how do how do we go about grouping these categories, as you can see in this our classification task, we are trying to classify the value of we are trying to classify, or predict the class of promotional offer variable that outcome variable of interest and therefore, with respect to keeping in mind this being the supervised approach supervised algorithm. So, everything is driven by the outcome variable. So, essentially the whole focus is in decreasing the overall error for classification of this particular variable, promotional offer and therefore, the pin codes they can be grouped with respect to this particular variable promotional offer.

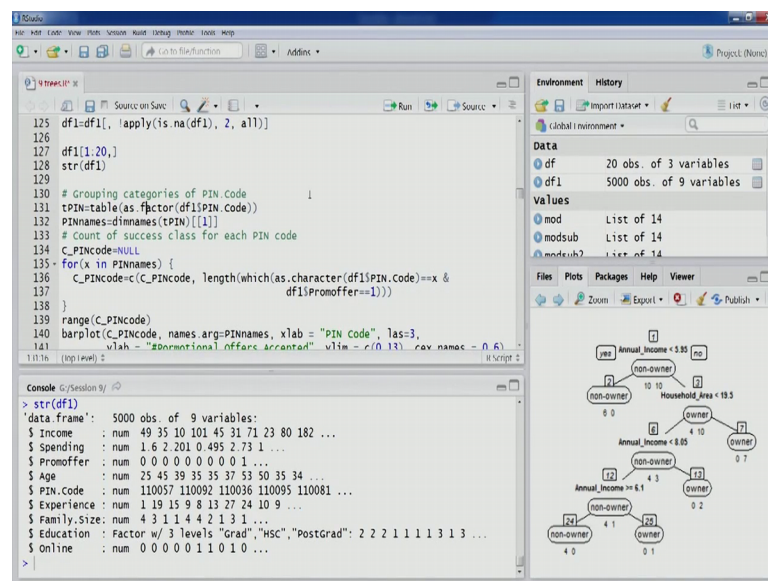
So, what we can actually do is the areas, which have similar sort of acceptance rate, or similar sort of rejection rate, they can actually be grouped together right. So, because the task is prediction of this particular variable promotional offer so, in terms of the you

know predictability or importance of this particular variable pin code, the areas are pin code or cities or towns having similar acceptance rate label, can be grouped and I all having similar rejection levels can be grouped together

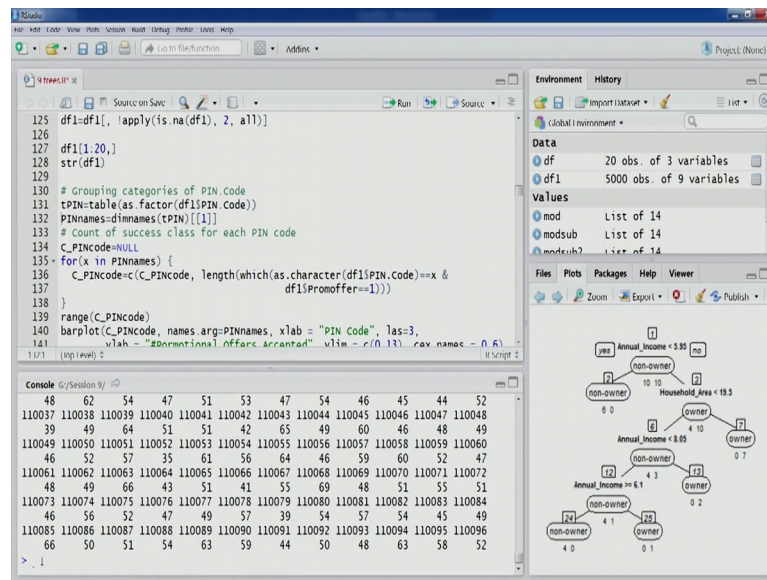
Because you know having them separately as individual category, and grouping them and grouping the locations with similar success rate would not you know impact much to the model, another will end up with fewer number of dimensions. So, we would be able to reduce the dimensionality by clubbing you know the similar types of you know locations, where the acceptance or rejection rate is similar.

So, it seem this kind of exercise we had all also done in visualisation technique, but that was limited to the you know generating bar plot. So, will go through the in an actual modeling exercise

(Refer Slide Time: 23:57)



(Refer Slide Time: 24:28)



So, we can do this, so you can see for each of these pin codes, you can also see the frequency. So, for example, the pin code 1 1 triple 0 1, this is 54 times it has appeared in the data set. The other pin codes also the similar numbers are being displayed there right.

So, the frequency for different pin codes we can easily see. Now pin names for these for these you know pin codes, we can easily extract. So, we want to extract the you know pin codes, and so we can do this using dim name dimension names function dim names function, and we can pass on this table object here t pin, and the first you know first element

So, this is actually as a list. So, first element of which is going to give us the pin names. So, we can store this here so, that we can later on use it when we generate the bar plot or for other things. So, what we are going to do as we talked about, because we want to group these categories as per their success rate as per their acceptance rate, or rejection rate of the promotional offer. So, we will have to count for each of these pin codes will have to count. So, that we can compute the acceptance or rejection proportion or rate right.

So, let us initialize this variable count pin code C, underscore pin code. Once this is a slice you can see I am running a loop here. So, for all the pin names which we have just computed for all of those pin names, we are going to run this loop, and in this loop you

can see within this combined function. The pin code and we are converting it into a character vector.

So, that it could be compared with x which is character vector, because pin names is the you know character vector, and here in this case same thing you can also immediately understand by looking at the environment section, you can see pin names is stored as character vector. So, therefore, pin code which is essentially the you know numeric vector the way it is stored right now

So, we will convert it into this. So, that comparison could be performed, and once for a particular value you know x that particular pin code, then when you look at whether how many of the observation for example, let us say pin code this 1 1 double 0 double 8. So, 54 times it is appearing in the data set. So, out of those 54 times how many times the customer has accepted the promotional offer.

So, we would like to count you know that we would like to count that the number of acceptance right. So, in this fashion so, once all the you know all the observations with that particular pin code are selected, and within that the promotional offer when that promotional offer was accepted. So, those observation will get the indices of those observation using the which function, and then length when we apply the length function on those that indices vector will get the count

So, for each pin code the success count the acceptance count, number of acceptance that will get through this codem, and then that is going to be stored here, in this particular c pin code variable. So, what is going to happen is for each of the pin codes which are there in pin names right. So, for each of them this kind of count of acceptance would be performed using this particular for loop.

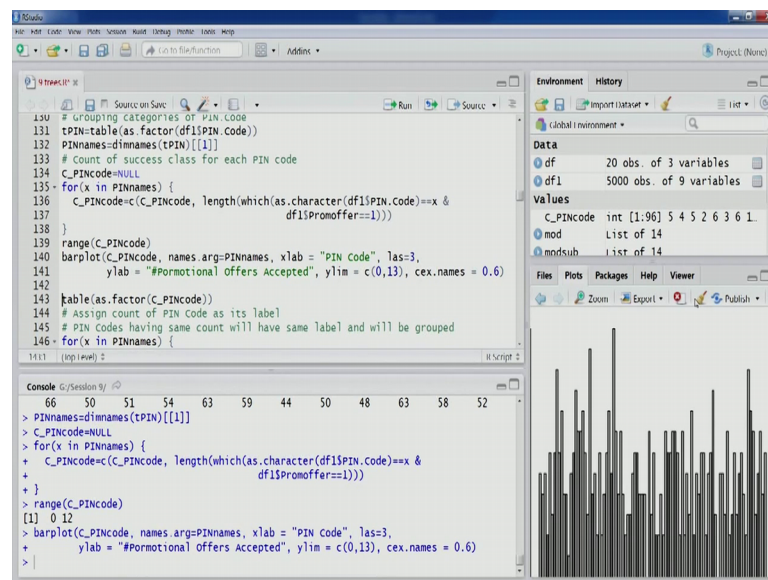
So, let us execute this code, now you would see in the environment section we have created C underscore pin code variable here, this is an integer variable all the all the values that are there they are counts actually. So, those all of them are integers we can see some of the values, you can see here in the environment section itself right C for different so, 96 pin codes were there right

So, for all those 96 pin codes we have the count of success, you can see in pin names the character vector that we had created 96 you know pin names were there. So, 96 pin code

for there, and for all those 96 pin codes 96 locations, we have been able to count the number of observations, which actually number of customer which who actually accepted the promotional offer

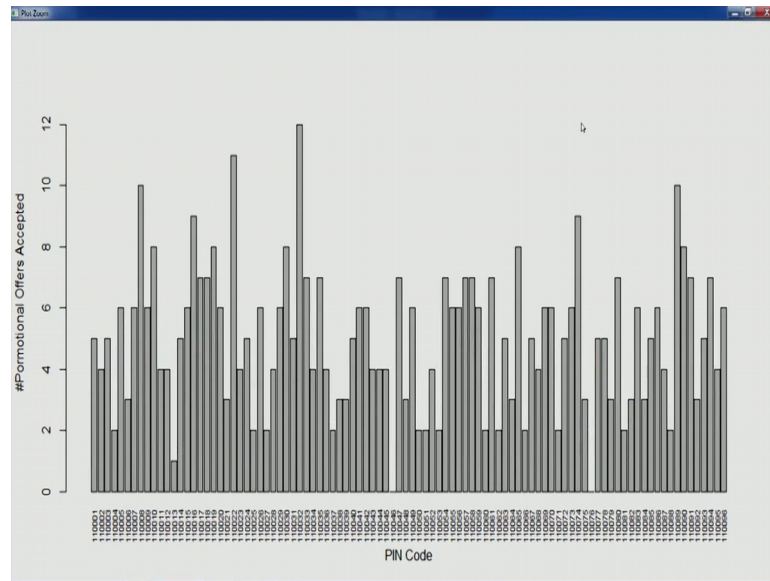
So, once we have this information. Let us look at the range before we keep generate the plot. So, you can see the maximum number of acceptance is 12 right, and the minimum is 0

(Refer Slide Time: 29:28)



So, with this information when we have this information, we can actually generate a bar plot and analyze visually what the what has been the you know acceptance, and the rejection proportion and other things. So, let us create this bar plot you can see the limits on y axis are appropriately specified 0 to 13. So, the range is within this limit, let us generate this plot. So, first we need to correct the graphics setting. Now so, again let us generate this plot we get the plot in the desired format.

(Refer Slide Time: 30:07)



So, you can see on the y axis we have number of promotional offers accepted. So, this is 0 to 12, and on the x axis as you can see the different pin codes have been mentioned here. So, all the pin codes that we have 96 of them, so all the pin codes for different pin code we can see the bars right.

So, from here we can further analyze, and identify which of the groups can be clubbed and whether there are any groups by looking at this plot, which can actually be you know clubbed or grouped. So, at this point we will stop here, and we will our discussion on the classification regression trees.

Thank you.