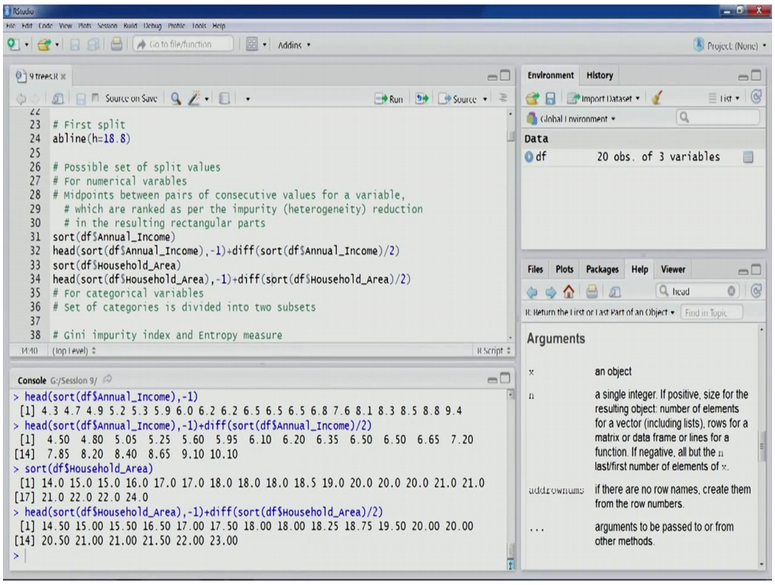
Business Analytics & Data Mining Modeling Using R
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture – 37
Classification and Regression Trees - Part II

Welcome to the course Business Analytics and Data Mining Modeling is Using R. So, in the previous lecture we started our discussion on classification and regression trees. So, we talked about two steps recursive partitioning and pruning and we started our discussion on recursive partitioning and we also started our exercise on the same and in the previous lecture. So, we were discussing about the possible set of split values what could be those values and how we can compute them, how we can get an idea about those split values using R.

So, in the previous lecture we talked about if the variable is numerical the predator is numerical then what could be the possible set of split values. So, we talked about annual income and also household area that sedan car dataset that we are using for this exercise. So, we also computed midpoint values for these two variables and we talked about we have two variables and 19 midpoint values for each of them twenty observation we have in total. So, about 38 predictor value combination will have and out of these 38 predator combination if the algorithm the implementation of that algorithm if it follows this process and out of these 38 combination we will have to select one optimal one which is going to reduce the impurity; that means, the heterogeneity.

(Refer Slide Time: 01:51)



```
# First split
abline(h=18.8)

# Possible set of split values
# For numerical varables
# Midpoints between pairs of consecutive values for a variable,
  # which are ranked as per the impurity (heterogeneity) reduction
  # in the resulting rectangular parts
sort(df$Annual_Income)
head(sort(df$Annual_Income),-1)+diff(sort(df$Annual_Income)/2)
sort(df$Household_Area)
head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
# For categorical variables
# Set of categories is divided into two subsets

# Gini impurity index and Entropy measure
```

```
> head(sort(df$Annual_Income),-1)
 [1] 4.3 4.7 4.9 5.2 5.3 5.9 6.0 6.2 6.2 6.5 6.5 6.5 6.8 7.6 8.1 8.3 8.5 8.8 9.4
> head(sort(df$Annual_Income),-1)+diff(sort(df$Annual_Income)/2)
 [1]  4.50  4.80  5.05  5.25  5.60  5.95  6.10  6.20  6.35  6.50  6.50  6.65  7.20
[14]  7.85  8.20  8.40  8.65  9.10 10.10
> sort(df$Household_Area)
 [1] 14.0 15.0 15.0 16.0 17.0 17.0 18.0 18.0 18.0 18.5 19.0 20.0 20.0 20.0 21.0 21.0
[17] 21.0 22.0 22.0 24.0
> head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
 [1] 14.50 15.00 15.50 16.50 17.00 17.50 18.00 18.00 18.25 18.75 19.50 20.00 20.00
[14] 20.50 21.00 21.00 21.50 22.00 23.00
>
```

Data

df 20 obs. of 3 variables

R: Return the First or Last Part of an Object

Arguments

x an object

n a single integer. If positive, size for the resulting object: number of elements for a vector (including lists), rows for a matrix or data frame or lines for a function. If negative, all but the n last/first number of elements of x.

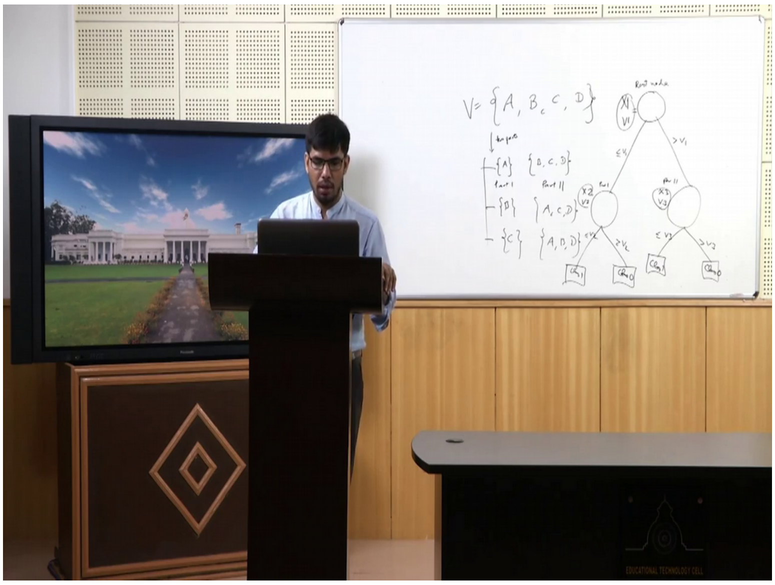
addrownums if there are no row names, create them from the row numbers.

... arguments to be passed to or from other methods.

That could be there in the resulting partition. So, resulting partition having the least impurity; that means, more you know homogenous partition so that particular value combination would actually be selected for (Refer Time: 02:06).

So, what if the variable if our variable is categorical? So, in that particular case the set of categories that we have they are divided into two subsets, for example, if we have a particular variable.

(Refer Slide Time: 02:29)

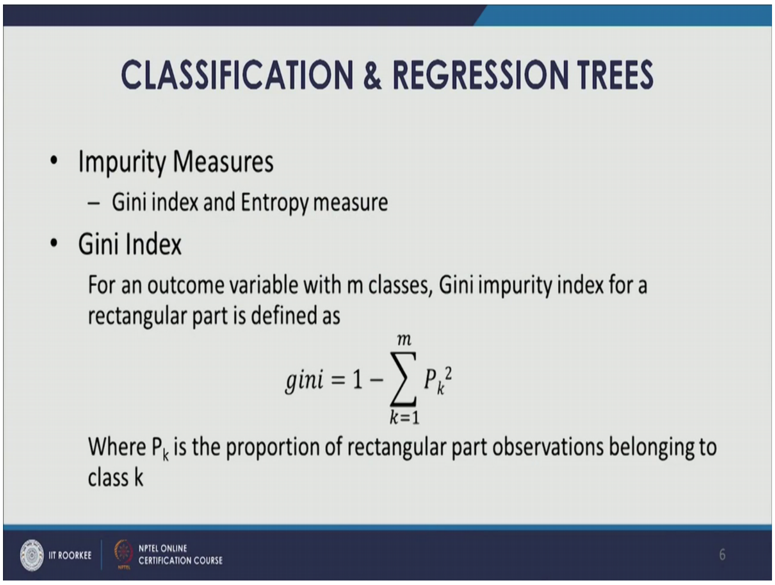


Let us say we have this variable. So, our values on the categories that are there, they could be this. So, from this we have to we can have many midpoint many set of possible candidates here right. So, there could be different you know value there different options here for example, you know this could be one. So, we have to create two parts from here. So, one category will go into one part the other categories will go into the other part right part 1 and part 2. So, in this fashion there could be various other candidates it could be B and others could be here then similarly it could be you know C and the others could be here. So, in this fashion there could be many combinations of these splits. So, there could be many split value the predictor and split value combination in this case also.

So, for categorical variable this is how we can create you know different combination of variable and split value. So, two subsets, for each the variable 4 categories A B C D, so all you know two subsets combination could be the different values that can be used as the possible set of candidate.

Now, let us talk about the impurity measures that we could be using for in this in this in this particular algorithm classification and regression tree. So, impurity measures that we are going to cover is two measures mainly a gini index and entropy measure. So, let say start our discussion on gini index.

(Refer Slide Time: 04:45)



CLASSIFICATION & REGRESSION TREES

- Impurity Measures
  - Gini index and Entropy measure
- Gini Index

For an outcome variable with m classes, Gini impurity index for a rectangular part is defined as

$$gini = 1 - \sum_{k=1}^{m} P_k^2$$

Where P_k is the proportion of rectangular part observations belonging to class k

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
6

So, for an, so both these majors whether gini index or entropy measures, they are in a sense major the impurity. So, for impurity for the original, original rectangle, original group in our in our data and then once we create partitions. So, two parts part one and part two for each of those parts we can further compute the impurity using these matrix. So, then later on we can compare that the after we have done the particular partition after we have done a particular split whether there has been a decrease in impurity. So, how do we measure that impurity of different partitions? So, these are the two matrix which can be used gini index and entropy measure.

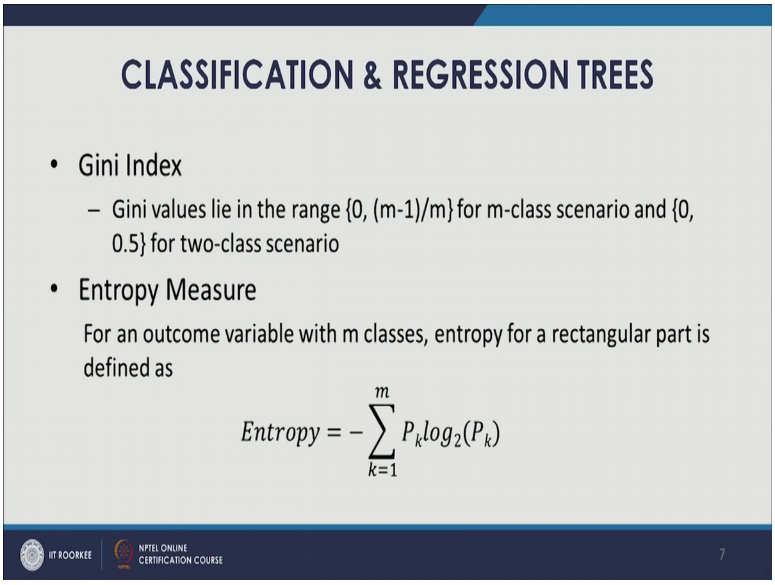
So, let us talk about the gini index first. So, for an outcome variable with m classes, gini impurity index for a rectangular part is defined as this gini 1 minus summation over k one to m because we have m classes and then P k square, where P k is the proportion of rectangular part observation belonging to class k. So, for each class if we have we have m classes, for each class will have to compute the proportion of observations belonging to that class in that particular rectangular part.

So, for example, if we had the full original rectangle all the observations and. So, we can compute the you know for each class, class 1 to m c 1, c 2 up to c m for each class we will have to compute the proportion values right proportion of observations belonging to class one in that particular rectangular part. Portion of observation belonging to class c 2 again in that same rectangular, in this fashion for all classes c 1 to c m we will have to compute the this proportion values P k and then square and summation of this. So, this will actually represent, this will actually the summation of this once we subtract this value from one this is actually going to represent the impurity right.

So, this will give us the impurity index for the rectangular part and once we create partition once we do a split we will have two more parts. So, for those two parts again we can use the same formula to compute their impurity value and these two parts we can add these two, we can add the impurity values of these two parts and then we can compare it with the original rectangular partition and see how much impurity has been reduced because of the partitioning alright. So, this is one particular metric that we can use.

Let us talk about the second metric entropy measure. So, before that let us understand the values gini values range. So, gini values lie in this range 0 m minus 1 divided by m.

(Refer Slide Time: 08:04)



CLASSIFICATION & REGRESSION TREES

- Gini Index
  - Gini values lie in the range {0, (m-1)/m} for m-class scenario and {0, 0.5} for two-class scenario
- Entropy Measure

  For an outcome variable with m classes, entropy for a rectangular part is defined as

$$Entropy = -\sum_{k=1}^{m} P_k log_2(P_k)$$

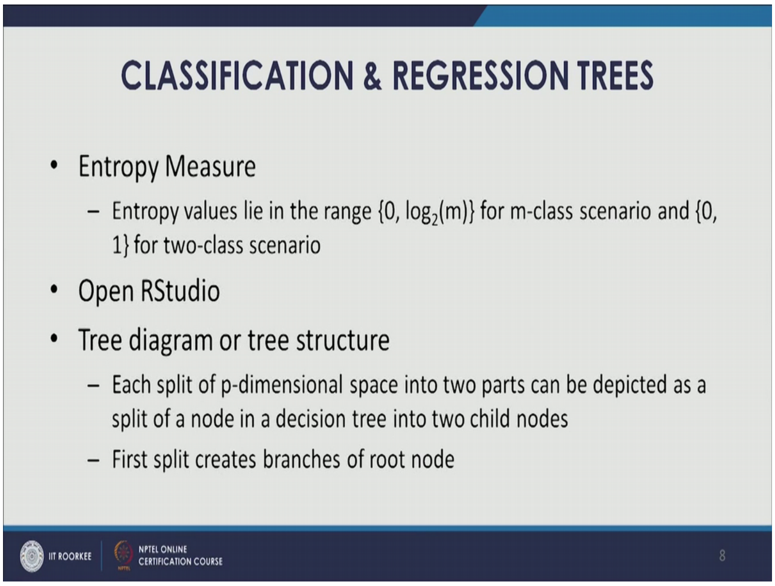
IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
7

So, if there are m classes. So, this is going to be the range for gini index and if there are if there are just two classes so the range is going to be 0 to 0.5. So, for 0 to 0 0.5, when the how we compute these two range? When in a two class scenario if the representation of both the classes is equal right, in that case the proportion would be 0.5 and 0.5 for both the classes. Now, if you go back to the expression here 1 minus summation over P k square. So, we have use 0.5 and 0.5 for both the classes. So, you would get that value right. So, the value that you get and then that is going to be 0.5 so that is going to be the highest value. So, when we have the equal representation from all the classes the value the gini index value is going to be the highest because there that is the situation where the impurity is, where the impurity is highest because the observations belonging to different classes they are equal. If there in a particle rectangular partition if most of the observations belong to one particular class then of course, impurity is less because very few observation would be belonging to other class.

If this you know this particular ratio keep on decreasing and becomes equal where you know the different classes the observation belong different classes they are in equal proportion then of course, the impurity is going to be the highest and that is also you know indicated in this particular range. So, m class scenario the value is going to be 0 m minus one divided by m and 2 class scenario the value the range is going to be 0 to 0 0.5.

Let us talk about the next metric that is entropy measure. So, for an outcome variable with m classes and entropy for a rectangular part is defined as this entropy minus summation over k equal to 1 to m and P k log and log of P k base 2. So, this is how we compute the entropy value. So, as we discussed for gini index right same thing P k stands for the same thing proportion of class k members in the rectangle in the rectangular part. So, then we compute that value then we take log of it log base 2 of it and then multiply these value and then sum it over and classes and the minus of that is going to be the entropy value.

(Refer Slide Time: 10:53)



CLASSIFICATION & REGRESSION TREES

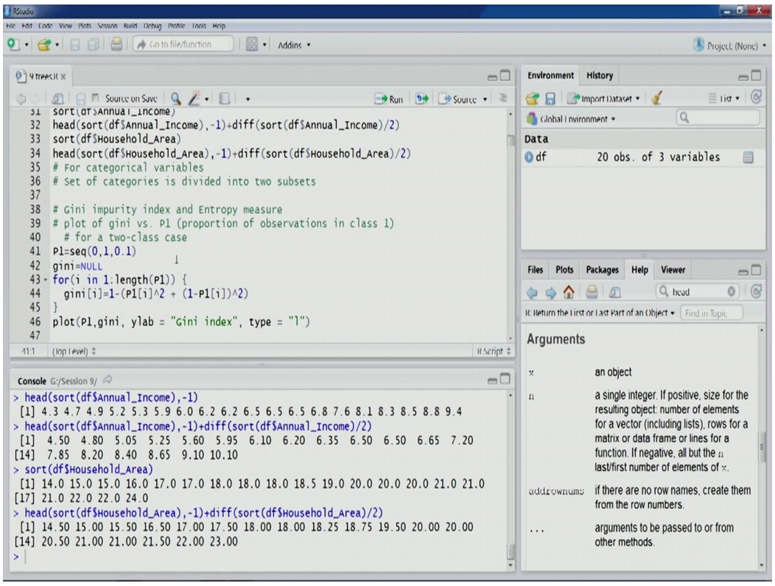
- Entropy Measure
  - Entropy values lie in the range {0, $\log_2(m)$} for m-class scenario and {0, 1} for two-class scenario
- Open RStudio
- Tree diagram or tree structure
  - Each split of p-dimensional space into two parts can be depicted as a split of a node in a decision tree into two child nodes
  - First split creates branches of root node

So, the range for entropy value, here it is going to be 0 and a log m base 2 for m class scenario and 0 and 1 for 2 class scenario, how?

So, for example, for two class scenario the highest impurity is going to be in this situation when the members belonging to each of those two classes they are in equal proportion they are in equal numbers. So, in that case the P k value is going to be 1 by 2 or 0.5. So, if the P k value is 1 by 2 you can plug in that value in this particular expression and you will get that log base 2 of 1 by 2 is going to be you know minus 1. So, that minus they will cancel out and then P k is there then that you will get the 1 by 2 and then for the second the other class also it will compute this value and once you sum it 1 by 2 plus 1 by 2 is going to be 1. So, this is how the range is.

So, highest impurity highest impurity scenario is when the all the classes they have equal proportion they have equal representation in a particular rectangular part right. So, that is when the highest impurity is going to be there and that will also give us the range for entropy values and also for gini index. So, what we will do? To understand more about these two particular matrix will do a simple exercise in R. So, let us go back.

(Refer Slide Time: 12:38)



```
sort(df$Annual_Income)
head(sort(df$Annual_Income),-1)+diff(sort(df$Annual_Income)/2)
sort(df$Household_Area)
head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
# For categorical variables
# Set of categories is divided into two subsets

# Gini impurity index and Entropy measure
# plot of gini vs. P1 (proportion of observations in class 1)
  # for a two-class case
P1=seq(0,1,0.1)
gini=NULL
for(i in 1:length(P1)) {
  gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
}
plot(P1,gini, ylab = "Gini index", type = "l")
```

```
> head(sort(df$Annual_Income),-1)
 [1] 4.3 4.7 4.9 5.2 5.3 5.9 6.0 6.2 6.2 6.5 6.5 6.5 6.8 7.6 8.1 8.3 8.5 8.8 9.4
> head(sort(df$Annual_Income),-1)+diff(sort(df$Annual_Income)/2)
 [1]  4.50  4.80  5.05  5.25  5.60  5.95  6.10  6.20  6.35  6.50  6.50  6.65  7.20
[14]  7.85  8.20  8.40  8.65  9.10 10.10
> sort(df$Household_Area)
 [1] 14.0 15.0 15.0 16.0 17.0 17.0 18.0 18.0 18.0 18.5 19.0 20.0 20.0 20.0 21.0 21.0
[17] 21.0 22.0 22.0 24.0
> head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
 [1] 14.50 15.00 15.50 16.50 17.00 17.50 18.00 18.00 18.25 18.75 19.50 20.00 20.00
[14] 20.50 21.00 21.00 21.50 22.00 23.00
> 
```

Data

```
df          20 obs. of 3 variables
```

Arguments

x an object

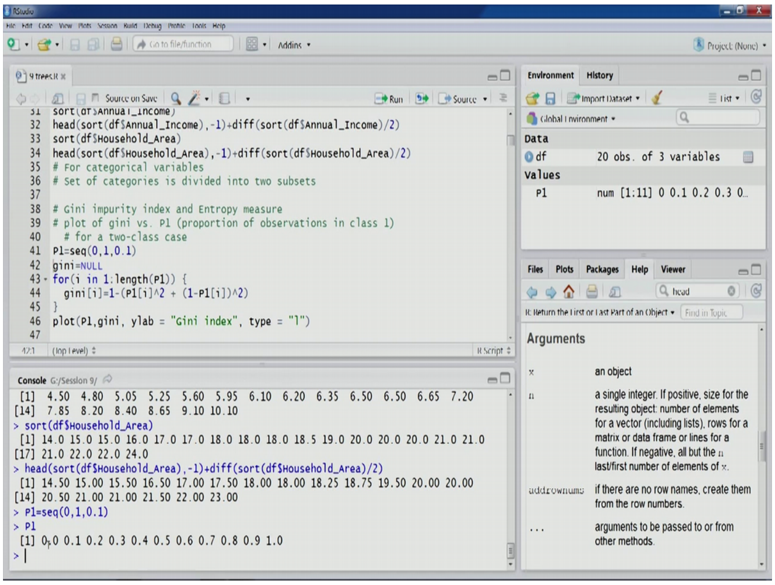
n a single integer. If positive, size for the resulting object: number of elements for a vector (including lists), rows for a matrix or data frame or lines for a function. If negative, all but the n last/first number of elements of x.

addrownums if there are no row names, create them from the row numbers.

... arguments to be passed to or from other methods.

So, let us first understand the plot of you know gini values versus P 1 this which is proportion of observations in class 1 now this is for a 2 class. So, let us understand how the plot is going to be depending on how we vary the proportion of observation belonging to class 1. So, let us say that P 1, this is our, sequence is the function that we can use to generate different proportions. So, let us compute this you can see P 1 has been created as you can see in the environment section and if you are interested in looking at the specific values. So, the proportion can range from 0 to 0.1 to 0.2 to 0.3 up to 0.9 and then 1.

(Refer Slide Time: 13:20)



```r
sort(df$Annual_Income)
head(sort(df$Annual_Income),-1)+diff(sort(df$Annual_Income)/2)
sort(df$Household_Area)
head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
# For categorical variables
# Set of categories is divided into two subsets

# Gini impurity index and Entropy measure
# plot of gini vs. P1 (proportion of observations in class 1)
  # for a two-class case
P1=seq(0,1,0.1)
gini=NULL
for(i in 1:length(P1)) {
  gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
}
plot(P1,gini, ylab = "Gini index", type = "l")
```

```
 [1]  4.50  4.80  5.05  5.25  5.60  5.95  6.10  6.20  6.35  6.50  6.50  6.65  7.20
[14]  7.85  8.20  8.40  8.65  9.10 10.10
> sort(df$Household_Area)
 [1] 14.0 15.0 15.0 16.0 17.0 17.0 18.0 18.0 18.0 18.5 19.0 20.0 20.0 20.0 21.0 21.0
[17] 21.0 22.0 22.0 24.0
> head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
 [1] 14.50 15.00 15.50 16.50 17.00 17.50 18.00 18.00 18.25 18.75 19.50 20.00 20.00
[14] 20.50 21.00 21.00 21.50 22.00 23.00
> P1=seq(0,1,0.1)
> P1
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> |
```

Data
df   20 obs. of 3 variables
Values
P1   num [1:11] 0 0.1 0.2 0.3 0...

R: Return the First or Last Part of an Object

Arguments

x — an object

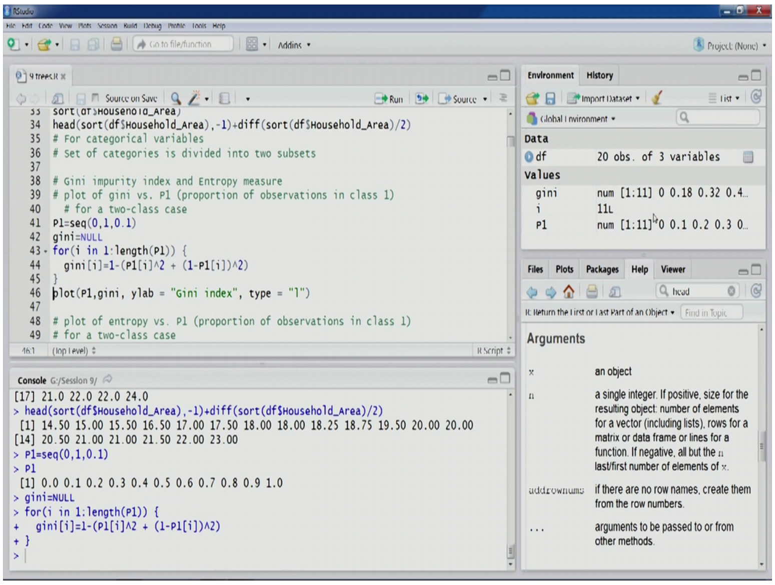
n — a single integer. If positive, size for the resulting object: number of elements for a vector (including lists), rows for a matrix or data frame or lines for a function. If negative, all but the n last/first number of elements of x.

addrownums — if there are no row names, create them from the row numbers.

... — arguments to be passed to or from other methods.

So, against these proportion values P 1 values we are going to compute gini index values and then we are going to plot them. So, as we are already familiar with gini index formula. So, let us first initialize this gini variable. So, let us do the initialization and then we are going to run this loop i in 1 to length of P 1 that is eleven values in total. So, for each of those values for each of those proportion values we are going to compute the gini index. So, this was the, this is how we can express the gini index formula here in (Refer Time: 00:00), 1 minus and within parenthesis we have for each proportion we first, we use the proportion values and they take a square of it and then we do a sum and then we add all these values for the all the classes.

(Refer Slide Time: 14:32)



```
sort(df$Household_Area)
head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
# For categorical variables
# Set of categories is divided into two subsets

# Gini impurity index and Entropy measure
# plot of gini vs. P1 (proportion of observations in class 1)
  # for a two-class case
P1=seq(0,1,0.1)
gini=NULL
for(i in 1:length(P1)) {
  gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
}
plot(P1,gini, ylab = "Gini index", type = "l")

# plot of entropy vs. P1 (proportion of observations in class 1)
# for a two-class case
```

```
[17] 21.0 22.0 22.0 24.0
> head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
 [1] 14.50 15.00 15.50 16.50 17.00 17.50 18.00 18.00 18.25 18.75 19.50 20.00 20.00
[14] 20.50 21.00 21.00 21.50 22.00 23.00
> P1=seq(0,1,0.1)
> P1
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> gini=NULL
> for(i in 1:length(P1)) {
+   gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
+ }
> 
```

```
Data
df        20 obs. of 3 variables
Values
gini      num [1:11] 0 0.18 0.32 0.4...
i         11L
P1        num [1:11] 0 0.1 0.2 0.3 0...
```

R: Return the First or Last Part of an Object

Arguments

x — an object

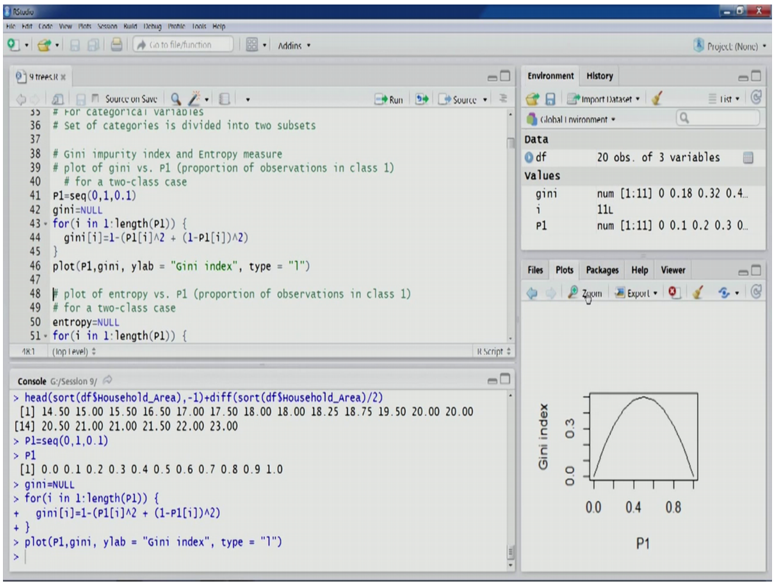
n — a single integer. If positive, size for the resulting object: number of elements for a vector (including lists), rows for a matrix or data frame or lines for a function. If negative, all but the n last/first number of elements of x.

addrownums — if there are no row names, create them from the row numbers.

... — arguments to be passed to or from other methods.

So, let us compute this. We would see that a gini vector numeric vector has been has been created again 11 value. So, 11 gini index values corresponding to different proportion values right. So, let us plot this.

(Refer Slide Time: 14:47)



```
# For categorical variables
# Set of categories is divided into two subsets

# Gini impurity index and Entropy measure
# plot of gini vs. P1 (proportion of observations in class 1)
  # for a two-class case
P1=seq(0,1,0.1)
gini=NULL
for(i in 1:length(P1)) {
  gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
}
plot(P1,gini, ylab = "Gini index", type = "l")

# plot of entropy vs. P1 (proportion of observations in class 1)
# for a two-class case
entropy=NULL
for(i in 1:length(P1)) {
```

```
> head(sort(df$Household_Area),-1)+diff(sort(df$Household_Area)/2)
 [1] 14.50 15.00 15.50 16.50 17.00 17.50 18.00 18.00 18.25 18.75 19.50 20.00 20.00
[14] 20.50 21.00 21.00 21.50 22.00 23.00
> P1=seq(0,1,0.1)
> P1
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> gini=NULL
> for(i in 1:length(P1)) {
+   gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
+ }
> plot(P1,gini, ylab = "Gini index", type = "l")
> 
```

```
Data
df        20 obs. of 3 variables
Values
gini      num [1:11] 0 0.18 0.32 0.4...
i         11L
P1        num [1:11] 0 0.1 0.2 0.3 0...
```

And this is the plot gini index versus proportion. Now, from here you can clearly understand as the proportion increases from 0 to 0 0.5 somewhere here you would see that gini index, index value is highest and it is 0.5 as we talked about right and as we further increase the increase the proportion right then again because P 1 this proportion keeps on increasing again then the gini index value this will start decreasing right. So, this will keep on decreasing and again when the proportion is 1 this will go to this will become 0. So, this is how the values are going to be for gini metric, gini index.

The same thing we can do for entropy measure as well. So, let us plot and let us plot a graph entropy versus P 1 that is proportion of observation class 1. So, P 1 for we have already defined. So, let us initialize the entropy here now within for loop you can see how we have written the code for calculation of entropy value. So, you can see for each class we have one expression and each expression we have proportional and then multiplied by log base 2 value of that proportion and once we sum all these expression and we take a minus of it. So, let us run this loop to find out the entropy values you can see 11 values have been created right. You would see that first particular value that is n a n it is showing as n a n, this is mainly because the proportion value for 0 here and log of 0 is not defined. So, because of that we have got this particular value.

So, let us plot. So, here you would see that in the plot function we are also using a spline function which will smooth smoothen the plot that we generate. So, let us see how what is going to happen. So, this is the plot here. So, you can see this particular plot is much more smoother than the plot that we had created for gini index. So, again here also as we move from 0 to 0 0.5 you would see that entropy measure is this particular value is maximum the value is 1 at 0.5 and as this proportion P 1 increases further this value goes down up to 0. So, this was about the two matrix, two matrix the gini index and entropy measure. So, let us talk further about our technique classification and regression trees.

So, next important point is the t diagram or t structure that we create. So, as we talked about the recursive partitioning steps. So, let us understand the t diagram what is how this is going to be built. So, for each split of P dimensional space into two parts, so that is of course, the part of recursive partitioning, can be depicted as a split of a node in a decision tree into two child nodes. So, we can have a root node right, we can have a root node, let us this is our root node and this is the original party partition then the each split that we perform it can be denoted using two nodes here, right. So, this is one part 1, this is part 2. So, in this fashion they split that we are talking about can be created.

So, P dimensional space if it is P dimensional space we will start with the root node and this is going to be partition two parts are going to be created. So, this can be represented in this fashion decision node having two child nodes. Now, once we have these two parts these two child nodes then again the same process would be applied on these two parts till you know, so the tree will start growing till the point we have created homogeneous partitions or homogeneous groups. So, first split creates branches of root node. So, as we can see. Now, two types of nodes in tree structure first one is a decision node. So, that is depicted with a circle here and then the second one is terminal or leaf node that is typically depicted using rectangle right.

So, these terminal nodes they typically they correspond to final rectangle parts. So, when we talk about just the recursive partitioning step where we build the full grown tree; that means, we get pure homogeneous parts. So, in that case we are going to have you know terminal nodes. So, for example, if this was you know root node and we created two partitions and once we created two partition we were able to achieve the homogeneous rectangles right. So, let us say further partitioning of this leads to homogeneous rectangles right. So, we will have, we can represent those nodes because they are going to be the terminal nodes leaf nodes using these rectangles right. So, these are decision nodes right. So, predictor and predictor value combination are going to be applied on these decision nodes and then the terminal nodes would indicate the actual class because this is now pure homogeneous group. So, it is going to be either class 1 or class 0, class 1 or class 0. So, in this fashion the tree structure could be there.

So, two types of node decision nodes and terminal node. So, decision nodes are the one where we apply the predictor value combinations and create a split and the terminal nodes or leaf nodes are the one where we finally, end up with pure homogeneous part homogeneous group and therefore, we can label it with the class name class 1 or class 0 if it is a two class case.

Now, let us understand the steps to classify new observations, new observation using tree based models. So, for a new observation once the tree has been built. So, new observation to be classified can be dropped down the tree. So, it can be dropped down from root node and then depending on the different comparison it will take different branches and the finally, it will end up with the terminal node or leaf node.

So, first step new observation to be classified is drop down the tree is starting from root node and at each decision node which also root, root node, root node is the first decision node. So, at each decision node the appropriate branch is taken until we reach a leaf node right. So, for example, this is a variable, variable V 1 and you know let us say X 1, this is X 1 and then the corresponding value for this particular you know variable is V 1 and the split is created right. So, values less than V 1 they go this side values greater than V 1 they go this side two parts alright. So, in this fashion here again we will have another variable X 2 and the value V 2 here we will have X 3 and value V 3 right and then the observation having value less than V 2 will go here greater than V 2 will come here similarly for here.

So, in this fashion we will continue till we till the new observation reach the terminal node or leaf node where then finally, it is going to be classified as per the class of that particular terminal node. So, finally, at leaf node majority class is assigned to the new observation. So, now, this is going to be when we do not have any special class of interest where we are trying to maximize the overall accuracy or trying to minimize the overall misclassification error, but when we have a special class of interest as we have been talking about in previous lectures for other techniques the steps are going to change a bit. For example for a class of interest scenario proportion of records belonging to the class of interest is compared with the user specified cut off value for the same right.

(Refer Slide Time: 23:52)

CLASSIFICATION & REGRESSION TREES

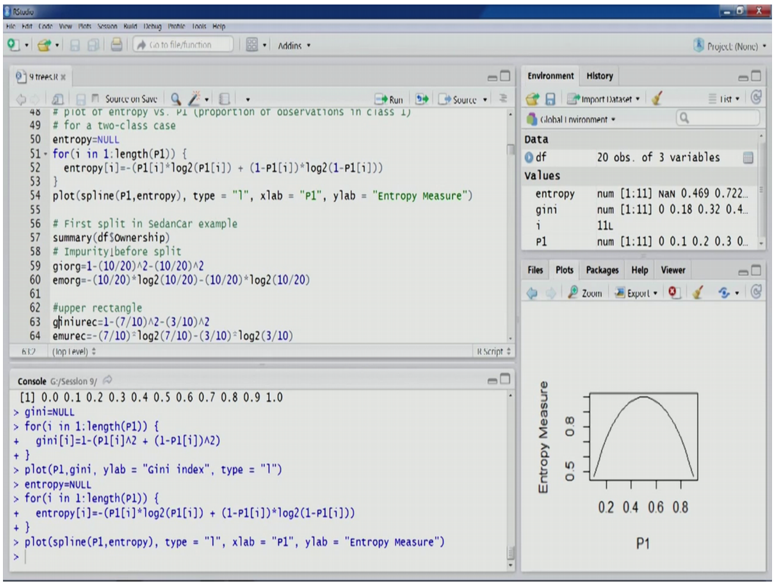
- Steps to classify a new observation using tree based models
  - At leaf node, majority class is assigned to the new observation
    - For class of interest scenario, proportion of records belonging to the class of interest is compared with the user specified cut off value for the same
- Open RStudio

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 10

So, for the once you once we reach the leaf node typically you know when we talk about the recursive partitioning it is going to be a purely homogeneous partition. So, there is going to be no such problem, but if the tree is not fully grown tree it has been pruned back pruning will discuss in coming lectures. So, in that case the partition the leaf terminal node might not be homogeneous and there could be some observation belonging to other classes. So, therefore, how do we decide? So, for when we try to, when we do not have any special class of interest and when we are looking to maximize overall accuracy in those situation we can just look at the majority class in the terminal node and assign that class to the new observation.

But when we have a class of interest we will compute the proportion of records belonging to that class of interest and then compare this particular proportion value to the user specified cut off value because that is the class of interest. So, we would like to identify more observations belonging to that class one even if it comes at the expense of miss identifying more observation belonging to other classes. So, the step is, this step final step is going to change depending on whether we have a class of interest or not.

(Refer Slide Time: 25:45)



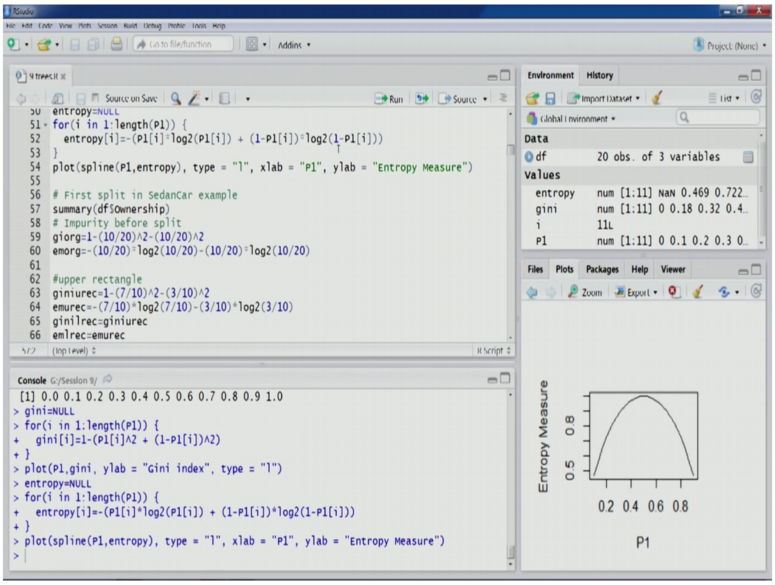
```
# plot of entropy vs. P1 (proportion of observations in class 1)
# for a two-class case
entropy=NULL
for(i in 1:length(P1)) {
  entropy[i]=-(P1[i]*log2(P1[i]) + (1-P1[i])*log2(1-P1[i]))
}
plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")

# First split in SedanCar example
summary(df$Ownership)
# Impurity before split
giorg=1-(10/20)^2-(10/20)^2
emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)

#upper rectangle
giniurec=1-(7/10)^2-(3/10)^2
emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
```

```
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> gini=NULL
> for(i in 1:length(P1)) {
+   gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
+ }
> plot(P1,gini, ylab = "Gini index", type = "l")
> entropy=NULL
> for(i in 1:length(P1)) {
+   entropy[i]=-(P1[i]*log2(P1[i]) + (1-P1[i])*log2(1-P1[i]))
+ }
> plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")
> 
```

```
Data
df          20 obs. of 3 variables
Values
entropy     num [1:11] NaN 0.469 0.722…
gini        num [1:11] 0 0.18 0.32 0.4…
i           11L
P1          num [1:11] 0 0.1 0.2 0.3 0…
```

So, what we will do? I will go through a simple exercise in R. So, let us go back to R, but before that let us also go through and this exercise where we compute the impurity using two matrix that we talked about.

(Refer Slide Time: 26:03)



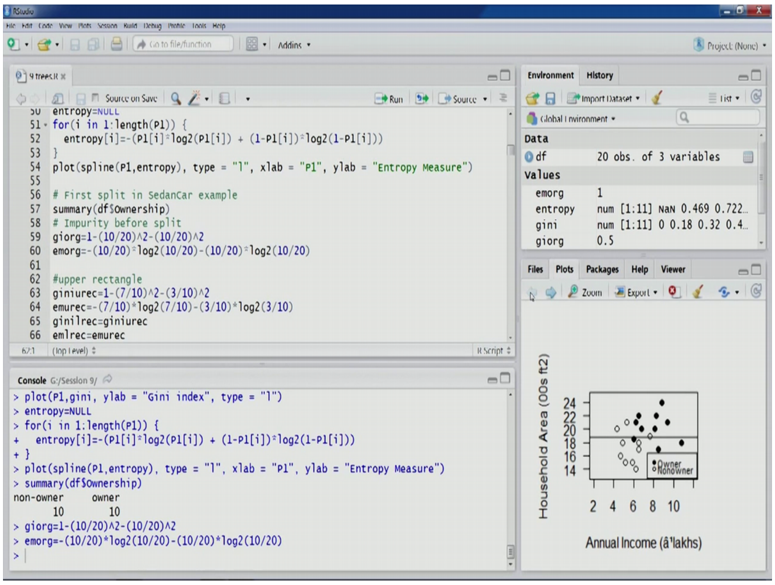
```r
entropy=NULL
for(i in 1:length(P1)) {
  entropy[i]=-(P1[i]*log2(P1[i]) + (1-P1[i])*log2(1-P1[i]))
}
plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")

# First split in SedanCar example
summary(df$Ownership)
# Impurity before split
giorg=1-(10/20)^2-(10/20)^2
emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)

#upper rectangle
giniurec=1-(7/10)^2-(3/10)^2
emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
ginilrec=giniurec
emlrec=emurec
```

```
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> gini=NULL
> for(i in 1:length(P1)) {
+   gini[i]=1-(P1[i]^2 + (1-P1[i])^2)
+ }
> plot(P1,gini, ylab = "Gini index", type = "l")
> entropy=NULL
> for(i in 1:length(P1)) {
+   entropy[i]=-(P1[i]*log2(P1[i]) + (1-P1[i])*log2(1-P1[i]))
+ }
> plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")
> 
```

```
Data
 df          20 obs. of 3 variables
Values
 entropy     num [1:11] NaN 0.469 0.722...
 gini        num [1:11] 0 0.18 0.32 0.4...
 i           11L
 P1          num [1:11] 0 0.1 0.2 0.3 0...
```

So, sedan car example that we have discussed before, let us look at the summary of this particular ownership variable. So, we have 10 observation belonging to non owner category and then observation belong to owner category. Now, the different matrix that we talked about the impurity index how we can compute. So, for gini index and entropy value for the original partition, original rectangle we can compute in this fashion you can see 1 minus because 10 observation belong to the non owner category out of 20. So, in this fashion we can compute the gini index for other classes as well. So, this would be the gini value. So, entropy value also we can compute in this fashion you can see 10 observation belong to owner non remaining 10 of belong to non owner. So, in this fashion we can compute the entropy value.

(Refer Slide Time: 26:50)



```
entropy=NULL
for(i in 1:length(P1)) {
  entropy[i]=-(P1[i]*log2(P1[i]) + (1-P1[i])*log2(1-P1[i]))
}
plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")

# First split in SedanCar example
summary(df$Ownership)
# Impurity before split
giorg=1-(10/20)^2-(10/20)^2
emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)

#upper rectangle
giniurec=1-(7/10)^2-(3/10)^2
emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
ginilrec=giniurec
emlrec=emurec
```

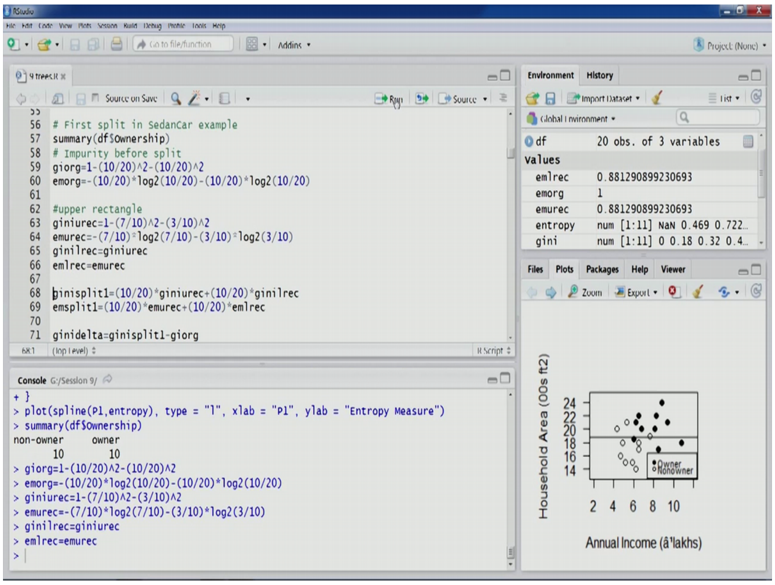
```
> plot(P1,gini, ylab = "Gini index", type = "l")
> entropy=NULL
> for(i in 1:length(P1)) {
+   entropy[i]=-(P1[i]*log2(P1[i]) + (1-P1[i])*log2(1-P1[i]))
+ }
> plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")
> summary(df$Ownership)
non-owner     owner
       10        10
> giorg=1-(10/20)^2-(10/20)^2
> emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)
> 
```

Data	
df	20 obs. of 3 variables
Values	
emorg	1
entropy	num [1:11] NaN 0.469 0.722…
gini	num [1:11] 0 0.18 0.32 0.4…
giorg	0.5

Now, once the first split that we had created earlier let us look at the graph. So, this was the graph you can see here we had created the first split at you know household area value of 18.8 and from this using this let us compute the gini entropy and entropy major values. So, from this let us zoom into this particular plot. So, in the upper rectangular part you can see we have 7 observations belonging to the owner class and 3 observations belonging to the non owner class. So, it is 7 out of 10 to owner and 3 out of 10 non owner for upper rectangular part. So, gini for upper rectangular is going to be 1 minus 7 divided by 10 and that is square of that then 3 by 2 divided by 10 square of that. So, in this fashion we can compute the gini value for upper rectangular. Similarly for the entropy value for the upper rectangular also we can compute using similar approach. So, let us compute these two values.

Now, if we look at the graph again you can see that lower rectangular part this is symmetric to the upper rectangular part in terms of proportion. So, portion of observations belonging to the owner and non owner. So, you know upper rectangular is dominated by owner lower rectangular is dominated by non owner, but the proportion they are very symmetric. So, the values for gini index and entropy measure they are going to be seen. So, why not assign the same values for lower rectangular as well. So, gini value is going to be same as follow a rectangular is going to be same as upper rectangular. Similarly entropy value is going to be a follow rectangular, is going to be same as that for upper rectangular.

(Refer Slide Time: 28:51)



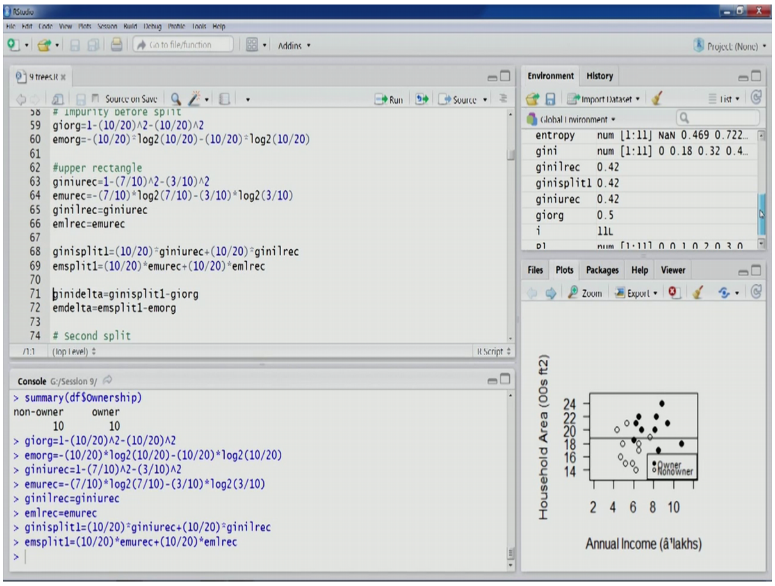
```
# First split in SedanCar example
summary(df$Ownership)
# Impurity before split
giorg=1-(10/20)^2-(10/20)^2
emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)

#upper rectangle
giniurec=1-(7/10)^2-(3/10)^2
emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
ginilrec=giniurec
emlrec=emurec

ginisplit1=(10/20)*giniurec+(10/20)*ginilrec
emsplit1=(10/20)*emurec+(10/20)*emlrec

ginidelta=ginisplit1-giorg
```

```
+ }
> plot(spline(P1,entropy), type = "l", xlab = "P1", ylab = "Entropy Measure")
> summary(df$Ownership)
non-owner     owner
       10        10
> giorg=1-(10/20)^2-(10/20)^2
> emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)
> giniurec=1-(7/10)^2-(3/10)^2
> emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
> ginilrec=giniurec
> emlrec=emurec
> 
```

```
df          20 obs. of 3 variables
Values
  emlrec    0.881290899230693
  emorg     1
  emurec    0.881290899230693
  entropy   num [1:11] NaN 0.469 0.722…
  gini      num [1:11] 0 0.18 0.32 0.4…
```

Once this is done, so for a split 1 we can compute the gini index value. So, we will add these two values for upper rectangular and lower rectangular. So, you can see we are also multiplying these value by their proportion here. So, 10 out of 20 observations in the upper rectangular, 10 out of 20 observation in the lower rectangular, this will give us the impurity index after first split and for entropy values of (Refer Time: 29:28) split.

(Refer Slide Time: 29:27)



```
# impurity before split
giorg=1-(10/20)^2-(10/20)^2
emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)

#upper rectangle
giniurec=1-(7/10)^2-(3/10)^2
emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
ginilrec=giniurec
emlrec=emurec

ginisplit1=(10/20)*giniurec+(10/20)*ginilrec
emsplit1=(10/20)*emurec+(10/20)*emlrec

ginidelta=ginisplit1-giorg
emdelta=emsplit1-emorg

# Second split
```

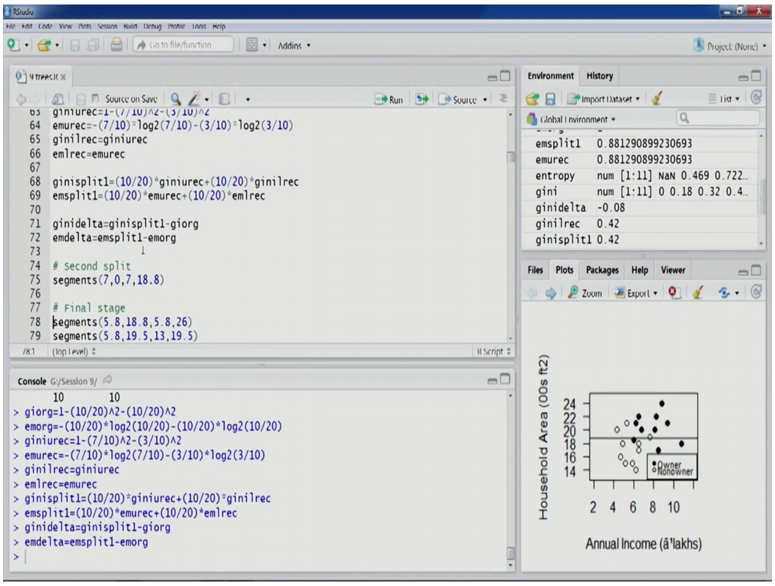
```
> summary(df$Ownership)
non-owner     owner
       10        10
> giorg=1-(10/20)^2-(10/20)^2
> emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)
> giniurec=1-(7/10)^2-(3/10)^2
> emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
> ginilrec=giniurec
> emlrec=emurec
> ginisplit1=(10/20)*giniurec+(10/20)*ginilrec
> emsplit1=(10/20)*emurec+(10/20)*emlrec
> 
```

```
  entropy    num [1:11] NaN 0.469 0.722…
  gini       num [1:11] 0 0.18 0.32 0.4…
  ginilrec   0.42
  ginisplit1 0.42
  giniurec   0.42
  giorg      0.5
  i          11L
  P1         num [1:11] 0 0.1 0.2 0.3 0…
```

So, you can see in the environment section. So, values have been created you can split one around 0.88 and again you split on around 0.42, and the original values also you can see original value is 0.5 g i o r g and e m o r g 1. So, now, we can compute the difference between you know that the delta that deduction that has happened in impurity. So, that is gini delta we can compute and e m delta. So, you can see e m delta minus this one minus 0.11 around minus 0.12 and gini delta is minus 0.08. So, if we can see there is a reduction in impurity. So, therefore, these two the first split is force is help us in achieving more, help us in achieving more homogeneous parts which is also very clearly visible from the clots as well.

So, in this fashion we can keep on continuing creating partition.

(Refer Slide Time: 30:38)



```
giniurec=1-(7/10)^2-(3/10)^2
emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
ginilrec=giniurec
emlrec=emurec

ginisplit1=(10/20)*giniurec+(10/20)*ginilrec
emsplit1=(10/20)*emurec+(10/20)*emlrec

ginidelta=ginisplit1-giorg
emdelta=emsplit1-emorg

# Second split
segments(7,0,7,18.8)

# Final stage
segments(5.8,18.8,5.8,26)
segments(5.8,19.5,13,19.5)
```

```
emsplit1    0.881290899230693
emurec      0.881290899230693
entropy     num [1:11] NaN 0.469 0.722…
gini        num [1:11] 0 0.18 0.32 0.4…
ginidelta   -0.08
ginilrec    0.42
ginisplit1  0.42
```

```
       10        10
> giorg=1-(10/20)^2-(10/20)^2
> emorg=-(10/20)*log2(10/20)-(10/20)*log2(10/20)
> giniurec=1-(7/10)^2-(3/10)^2
> emurec=-(7/10)*log2(7/10)-(3/10)*log2(3/10)
> ginilrec=giniurec
> emlrec=emurec
> ginisplit1=(10/20)*giniurec+(10/20)*ginilrec
> emsplit1=(10/20)*emurec+(10/20)*emlrec
> ginidelta=ginisplit1-giorg
> emdelta=emsplit1-emorg
> 
```

So, I will stop here and the other partition and the values gini values and the other excises and discussion will continue in the next lecture.

Thank you.