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

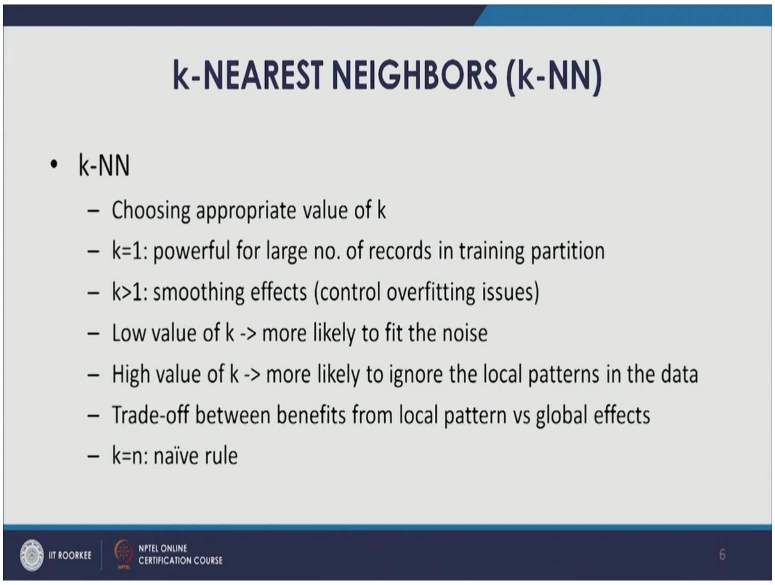
# Lecture - 29
# Machine Learning Technique K-Nearest Neighbors(K-NN)- Part II

Welcome to the course business analytics and data mining modeling using R. So, in the previous lecture we started about discussion on another technique K nearest neighbors or K Nn. So, we discussed most of the theoretical part, the discussion what K NN is about, the difference of K NN from K NN being the data mining technique, the difference of it from other technique like statistical technique, multiple linear regression.

So, we discussed all those things, we also did an exercise in R where we use the sedan car or x l s data set and did few did the partitioning normalisation and then the modeling. So, we are we also looked at the results and. So, the next important aspect of K NN is that the selection of this value of k. So, appropriate value of K, the optimal value of K how do we find out that.

So, in the previous lecture we in our small excise that we done, we had used the value of K as 4 for the illustration purpose, but the optimal value of K could be different. So, how we find out that? So, first we need to understand certain thing about this particular aspect of K NN, the selection of choosing a an appropriate value of k. So, for example, if we take K value as 1.

(Refer Slide Time: 01:54)



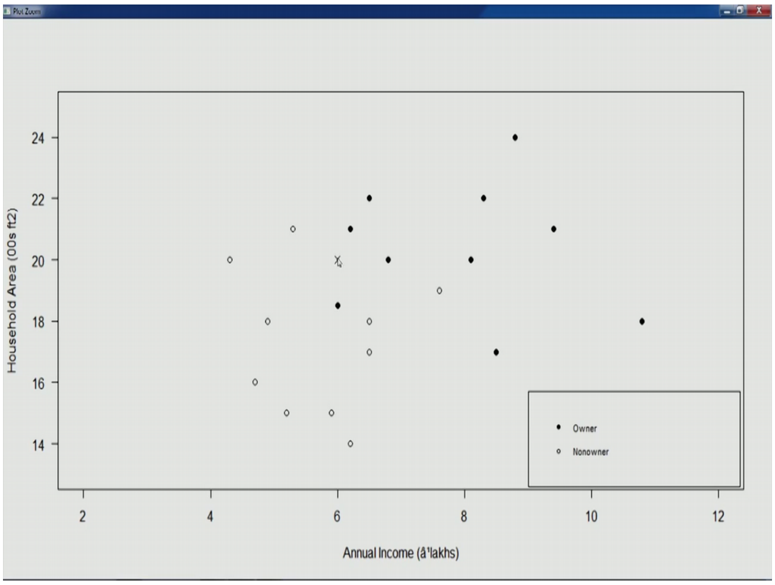
## k-NEAREST NEIGHBORS (k-NN)

- k-NN
  - Choosing appropriate value of k
  - k=1: powerful for large no. of records in training partition
  - k>1: smoothing effects (control overfitting issues)
  - Low value of k -> more likely to fit the noise
  - High value of k -> more likely to ignore the local patterns in the data
  - Trade-off between benefits from local pattern vs global effects
  - k=n: naïve rule

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
6

Value of K as 1, then this going to be a powerful technique especially for large number of records in any partition because K being 1 fewer number of computation, distance based computation would have to be performed and because the value of K is 1 so; that means, we are essentially dealing with just 1 neighbour. And therefore, using this just 1 neighbour the class of once we identify the closest neighbour, the class of a that neighbour itself is going to be the predicted class of new observation.

(Refer Slide Time: 02:44)



So, for example, the plot that we had generated in the previous lecture we are able to use that plot then this one. So, for example, if this is the new observation represented by x symbol and if we are able find the, if we are require to find just the 1 a closest neighbor of this new observation then it would be probably this point, this point seems to be the closet point. So, the class of this closest point would actually be assigned to this particular new observation.

So, if the value K is 1, then the computation number of computation would just we want distance between a you know number of computation would be much less and the easily we can classify new observations and assign the same, if the value of if the value of K is greater than 1. So, as we move towards higher value of K the smoothing effect will start to takes place, smoothing effect in the sense because the, if the value of K is more than 1 lets says 4, 5, 6, 7 even 10. Then the different, more number of neighbours would actually be used to classify the record and therefore, it would in a way control over fitting is used. When we have the value of K low value of K or let us say K called if value of K is 1, then we might in the fitting to the noise. So, therefore, if you have larger value then that is smoothing effect with will in a way control some of the over fitting is used arise arising due to noise.

So, the next point exactly talks about this that low value of K it might lead to you know more chances of fitting the noise, if we opt for a high value of K than more likely to ignore the local patterns in the data. If we have going for high value of K then we might ignore the local patterns because the smoothening a would be heightened and therefore, those global effects would actually be, you know the will be dominant effects will take dominance and the local patterns might be ignored.

So, therefore, we have to trade of between benefits from local pattern verses global effects. So, we go we target lot local pattern and we select a small value of K we might and fitting the noise therefore, we do not want that and we go for the global effects and go for the smoothening smoothing effects. So, that we are able to control over fitting issues, we might end up ignoring the local patterns and we might end up ignoring the predictors information as well.

So, for example, if we take the next point being, if we take the value of K as n that is; that means, all the observation that are there then that would essentially lead to naive rule; that means, majority hold decision. So, the if you take all the observation; that means, the class which is the most prevalent among all those observation that would be assign to the new observation. So, irrespective of the predictors information. So, we would end up ignoring the predictors information and more often the not all most in all the cases for all new observation the majority class would be assigned. So, we do not want this situation as well, we would like to balance between you know ignoring the predictors information and also fitting to the noise.

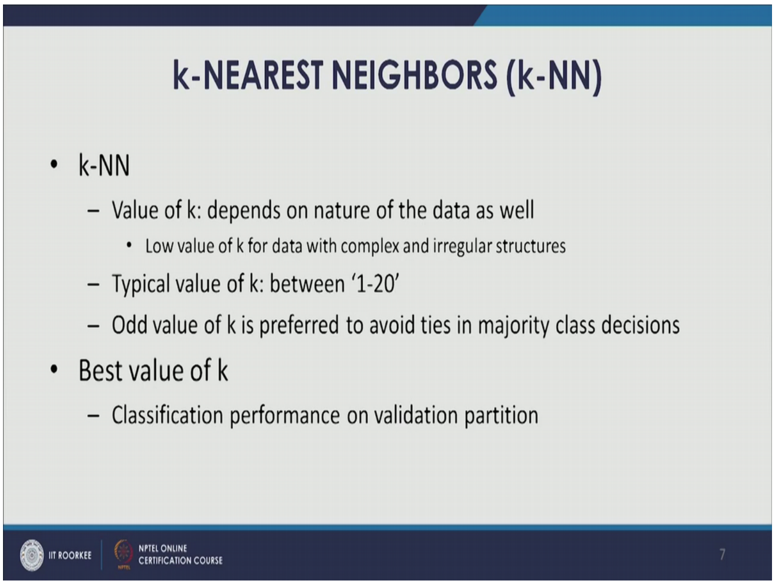
When we have low value of K we want to fee we want to fit to the local pattern right, the genuine patterns that could be there and we will not like to fit to the voice. When we have higher value of K we will not like to ignore the predictors information or you know too much of smoothing effects. So, you need to balance out, the local patterns and global effects incorporation of predictors information and avoiding the fitting to the noise.

So, let us a move further, now value of appropriate value of K will also defined on nature of that, nature of data for example, if the data that we have is slightly complex and has carries the irregular structures then we would probably better of having low value of K right. Because the global effects might not be suitable higher value of K and therefore, global effects there is smoothing might not be suitable for this type of data and we would be better of capturing the local patterns because of the complexity that is involved in the data.

However if we have you know, if we have regular dataset regular a structures which we are familiar with in those cases it is you know higher value of K might be preferred right. We understand those structures and there for we would we better of focusing on the smoothing effects. So, that we are able to control the over fitting issues or due to which mainly arise due to fitting to the noise.

Now, we are look at the typical value of K that could be you know generally that is suitable. So, between 1 to 20 so typically.

(Refer Slide Time: 08:35)



## k-NEAREST NEIGHBORS (k-NN)

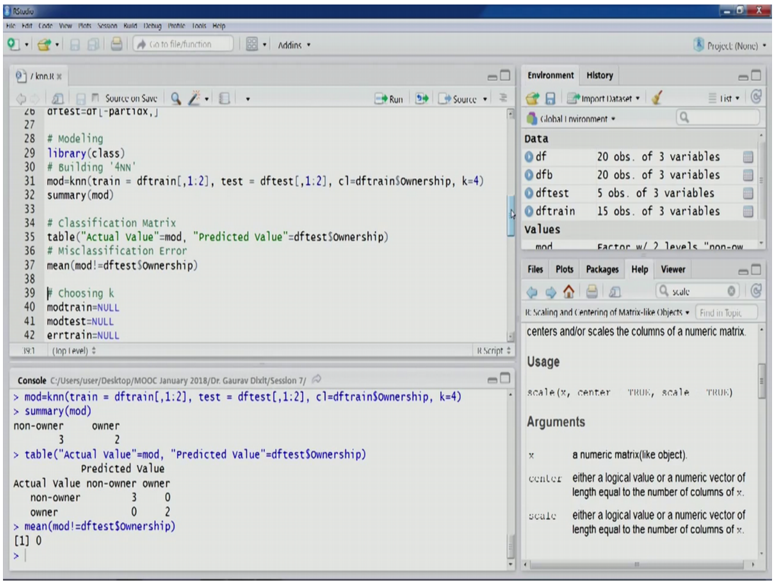
- k-NN
  - Value of k: depends on nature of the data as well
    - Low value of k for data with complex and irregular structures
  - Typical value of k: between '1-20'
  - Odd value of k is preferred to avoid ties in majority class decisions
- Best value of k
  - Classification performance on validation partition

This is, so we take a value of K between 1 to 20, now even within this range the odd value of K is preferred to avoid price in majority class decision. So, because we are in a classification task discussing classification task, there when we have to make our prediction decision using to majority class rule. Once the K neighbours have been identified then there the if there are the value of K is even number 4, 6, 8 or 10 then sometime there could be ties 2 class having suppose if the value of K is 8 you know there are 2 class, one having 4 neighbours you know representing that class the other class is being representing by remaining 4 neighbours.

So, in that case there would be ties and it would be difficult to assign the new observation you know to assign the new observation there for to avoid those ties the ideal value of K would be the odd value. Now, having discussed all these points the low value of K the high value of K the nature of the data the structure that is complex and irregular or regular difficult value of K and the hard value. So, having discussed all these points how do we find out the best value of K should not would be based on the performance. So, classification performance on validation partition it could actually be used to find out the best value of k.

So, what we are going to do is will try to understand through an excise in R how do we select the best value of k.

(Refer Slide Time: 10:31)



```
dftest=df[-partidx,]

# Modeling
library(class)
# Building '4NN'
mod=knn(train = dftrain[,1:2], test = dftest[,1:2], cl=dftrain$Ownership, k=4)
summary(mod)

# Classification Matrix
table("Actual Value"=mod, "Predicted Value"=dftest$Ownership)
# Misclassification Error
mean(mod!=dftest$Ownership)

# Choosing k
modtrain=NULL
modtest=NULL
errtrain=NULL
```

```
> mod=knn(train = dftrain[,1:2], test = dftest[,1:2], cl=dftrain$Ownership, k=4)
> summary(mod)
non-owner     owner
        3         2
> table("Actual Value"=mod, "Predicted Value"=dftest$Ownership)
            Predicted Value
Actual Value non-owner owner
   non-owner         3     0
   owner             0     2
> mean(mod!=dftest$Ownership)
[1] 0
>
```

```
Data
df        20 obs. of 3 variables
dfb       20 obs. of 3 variables
dftest    5 obs. of 3 variables
dftrain   15 obs. of 3 variables
Values
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
knn.R
Source on Save
Run
Source
Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/
Environment
History
Import Dataset
List
Global Environment
Files
Plots
Packages
Help
Viewer
R: Scaling and Centering of Matrix-like Objects
Find in Topic

centers and/or scales the columns of a numeric matrix.

Usage

scale(x, center = TRUE, scale = TRUE)

Arguments

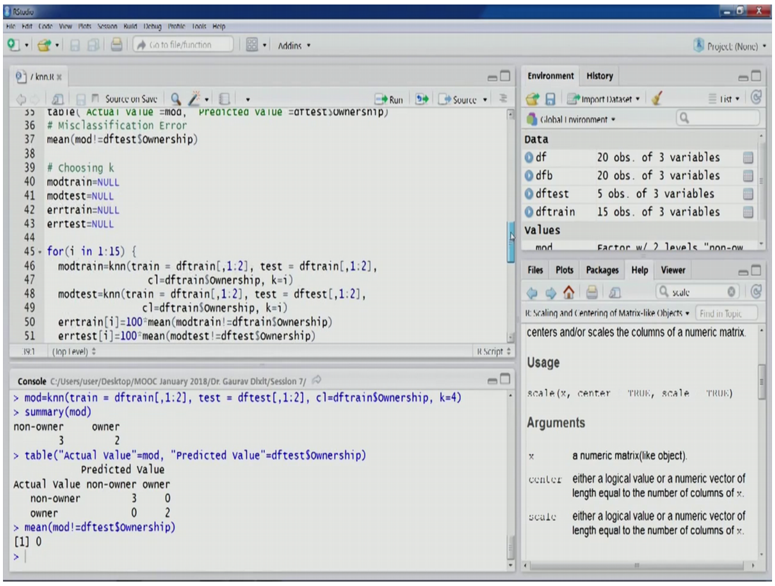
x a numeric matrix(like object).

center either a logical value or a numeric vector of length equal to the number of columns of x.

scale either a logical value or a numeric vector of length equal to the number of columns of x.

So, let us open R studio and the data set for the excise that we are going to use is the same data set on that using in previous lecture. So, that is the sedan car data set. So, we already know 3 variables that are there the house hold income, the income annual income, the house hold area and the ownership. So, what we are going to do is.

(Refer Slide Time: 10:59)

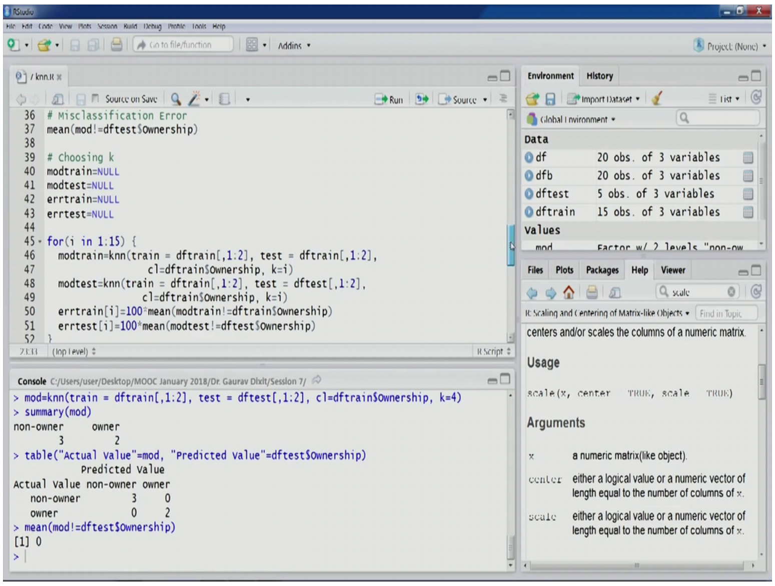


```
table("Actual Value"=mod, "Predicted Value"=dftest$Ownership)
# Misclassification Error
mean(mod!=dftest$Ownership)

# Choosing k
modtrain=NULL
modtest=NULL
errtrain=NULL
errtest=NULL

for(i in 1:15) {
  modtrain=knn(train = dftrain[,1:2], test = dftrain[,1:2],
                cl=dftrain$Ownership, k=i)
  modtest=knn(train = dftrain[,1:2], test = dftest[,1:2],
               cl=dftrain$Ownership, k=i)
  errtrain[i]=100*mean(modtrain!=dftrain$Ownership)
  errtest[i]=100*mean(modtest!=dftest$Ownership)
```

```
> mod=knn(train = dftrain[,1:2], test = dftest[,1:2], cl=dftrain$Ownership, k=4)
> summary(mod)
non-owner     owner
        3         2
> table("Actual Value"=mod, "Predicted Value"=dftest$Ownership)
            Predicted Value
Actual Value non-owner owner
   non-owner         3     0
   owner             0     2
> mean(mod!=dftest$Ownership)
[1] 0
>
```

There are in this particular in this particular data set that we have total number of observations are 20 and out of these 20 observation we had we had partitioned 15 observations into the training partition and remaining 5 in to test partition we were also able to we also build a model in the previous lecture using 4 K value as 4 right. So, right, so not.

(Refer Slide Time: 11:30)



```
# Misclassification Error
mean(mod!=dftest$Ownership)

# Choosing k
modtrain=NULL
modtest=NULL
errtrain=NULL
errtest=NULL

for(i in 1:15) {
  modtrain=knn(train = dftrain[,1:2], test = dftrain[,1:2],
                cl=dftrain$Ownership, k=i)
  modtest=knn(train = dftrain[,1:2], test = dftest[,1:2],
               cl=dftrain$Ownership, k=i)
  errtrain[i]=100*mean(modtrain!=dftrain$Ownership)
  errtest[i]=100*mean(modtest!=dftest$Ownership)
}
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
> mod=knn(train = dftrain[,1:2], test = dftest[,1:2], cl=dftrain$Ownership, k=4)
> summary(mod)
non-owner     owner
        3         2
> table("Actual Value"=mod, "Predicted Value"=dftest$Ownership)
            Predicted Value
Actual Value non-owner owner
   non-owner         3     0
   owner             0     2
> mean(mod!=dftest$Ownership)
[1] 0
> 
```

```
Data
df        20 obs. of 3 variables
dfb       20 obs. of 3 variables
dftest    5 obs. of 3 variables
dftrain   15 obs. of 3 variables
Values
mod       Factor w/ 2 levels "non-ow
```

R: Scaling and Centering of Matrix-like Objects

centers and/or scales the columns of a numeric matrix.

Usage

```
scale(x, center   TRUE, scale   TRUE)
```

Arguments

x	a numeric matrix(like object).

center	either a logical value or a numeric vector of length equal to the number of columns of x.

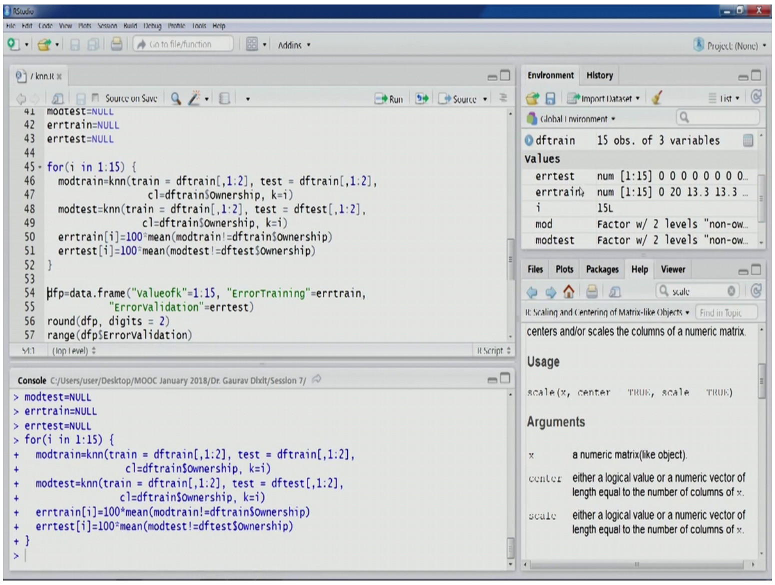
scale	either a logical value or a numeric vector of length equal to the number of columns of x.

Now, what we are going to do is different possibility this the possible value of K could be from 1 to 50 depending. So, the these are the observation that are there 15 observation are there in the with any partition and these are less than the typical value that we talk about 1 to 20. So, therefore, will len loop for all these you know 15 times. So, K value of K could range from 1 to 15.

So, in the. So, what we are going to do is for different, for different value of K we would be building K NN model and for each of those models we would be predicting the class of observations that are there in the text partition as well as we would also be rescoring the training partition, we would be predicting the training partition observation of training partition itself, right. So, and then we would be recording the errors this classification errors in training partition and testing partition. So, as you can see we have initialized these 4 variables, 4 vectors mod train. So, that is for modelling, that is for the modelling using train training partition and then scoring the training partition itself.

Then the next one is mod test. So, this is for scoring the test partition and. So, let us set this, initialize this then we have error training. So, this is the misclassification this variable is to record the misclassification error and different in the training partition for different values of K.

(Refer Slide Time: 13:27)



```
modtest=NULL
errtrain=NULL
errtest=NULL

for(i in 1:15) {
  modtrain=knn(train = dftrain[,1:2], test = dftrain[,1:2],
                cl=dftrain$Ownership, k=i)
  modtest=knn(train = dftrain[,1:2], test = dftest[,1:2],
               cl=dftrain$Ownership, k=i)
  errtrain[i]=100*mean(modtrain!=dftrain$Ownership)
  errtest[i]=100*mean(modtest!=dftest$Ownership)
}

dfp=data.frame("Valueofk"=1:15, "ErrorTraining"=errtrain,
           "ErrorValidation"=errtest)
round(dfp, digits = 2)
range(dfp$ErrorValidation)
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
> modtest=NULL
> errtrain=NULL
> errtest=NULL
> for(i in 1:15) {
+   modtrain=knn(train = dftrain[,1:2], test = dftrain[,1:2],
+                cl=dftrain$Ownership, k=i)
+   modtest=knn(train = dftrain[,1:2], test = dftest[,1:2],
+               cl=dftrain$Ownership, k=i)
+   errtrain[i]=100*mean(modtrain!=dftrain$Ownership)
+   errtest[i]=100*mean(modtest!=dftest$Ownership)
+ }
> 
```

```
dftrain     15 obs. of 3 variables
Values
  errtest   num [1:15] 0 0 0 0 0 0 0 0...
  errtrain  num [1:15] 0 20 13.3 13.3 ...
  i         15L
  mod       Factor w/ 2 levels "non-ow...
  modtest   Factor w/ 2 levels "non-ow...
```

R: Scaling and Centering of Matrix-like Objects

centers and/or scales the columns of a numeric matrix.

Usage

```
scale(x, center = TRUE, scale = TRUE)
```

Arguments

| | |
|---|---|
| x | a numeric matrix(like object). |
| center | either a logical value or a numeric vector of length equal to the number of columns of x. |
| scale | either a logical value or a numeric vector of length equal to the number of columns of x. |

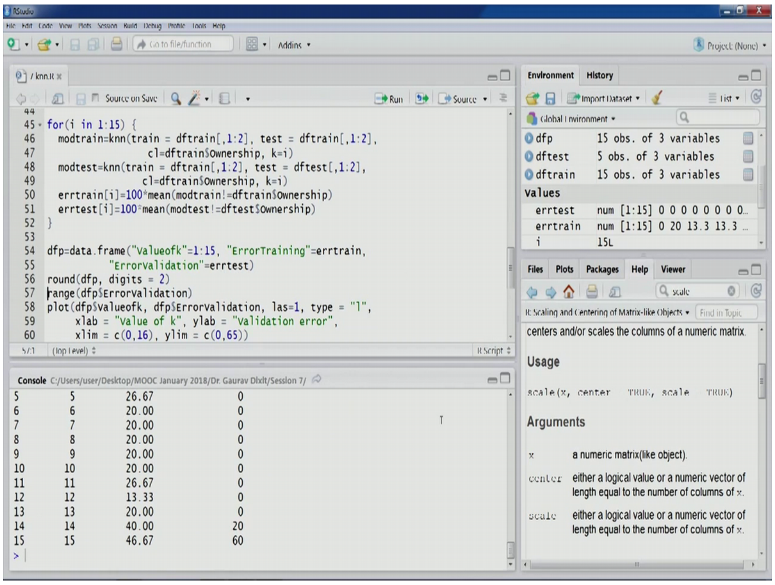
Then we have another vector another variable error test to record the misclassification error misclassification error associated with the you know different value of K for the observation and testing partition.

So, once this is done we can run this loop, you can see that for both the training and test test partition as usual we have selected the first 2 variables numerical variables the income and household area. It also see that the one different within these 2 lines of code is that this third argument that is c l function this is this remain same. So, the training of finding the K neighbours would be based on the training partition and it is the test partition that is different in these 2 lines off code we can see here it is d f train. So, we are building the K NN modelling on the training partition and then the scoring the observation of training partition itself and then the second pa code second line of code we are building the K NN model, training observation that essentially means identifying K neighbours and then scoring the observation belonging to test partition.

So, an every traction for every value of K we would be recording different errors we would be you know running the these doing this scoring welding these models and doing these scoring part training and testing partition. So, let us execute this particular code, we would see that different variable have been created the, right now we would like to focus on error test and error train. So, you can see two numerical variables in the environment section having 15 values right. So, once this loop has been executed we can move to the next line of code. So, we are constructing a data frame where the first column is representing the a value of K that is between one to 50 and then the second column is error training. So, that is has been captured in this particular variable error train and the third column is for misclassification error that is error validation that has been captured in error test variable. So, let us construct this data frame.

Now, we are interested in just the 2 values after decimal points. So, let us round these values.

(Refer Slide Time: 16:03)



```
for(i in 1:15) {
  modtrain=knn(train = dftrain[,1:2], test = dftrain[,1:2],
                cl=dftrain$Ownership, k=i)
  modtest=knn(train = dftrain[,1:2], test = dftest[,1:2],
               cl=dftrain$Ownership, k=i)
  errtrain[i]=100*mean(modtrain!=dftrain$Ownership)
  errtest[i]=100*mean(modtest!=dftest$Ownership)
}

dfp=data.frame("Valueofk"=1:15, "ErrorTraining"=errtrain,
            "ErrorValidation"=errtest)
round(dfp, digits = 2)
range(dfp$ErrorValidation)
plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = "l",
     xlab = "Value of k", ylab = "Validation error",
     xlim = c(0,16), ylim = c(0,65))
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
5        5       26.67          0
6        6       20.00          0
7        7       20.00          0
8        8       20.00          0
9        9       20.00          0
10      10       20.00          0
11      11       26.67          0
12      12       13.33          0
13      13       20.00          0
14      14       40.00         20
15      15       46.67         60
>
```

```
dfp        15 obs. of 3 variables
dftest     5 obs. of 3 variables
dftrain    15 obs. of 3 variables
Values
  errtest    num [1:15] 0 0 0 0 0 0 0 0...
  errtrain   num [1:15] 0 20 13.3 13.3 ...
  i          15L
```

centers and/or scales the columns of a numeric matrix.

Usage

```
scale(x, center  TRUE, scale  TRUE)
```

Arguments

x  a numeric matrix(like object).

center  either a logical value or a numeric vector of length equal to the number of columns of x.

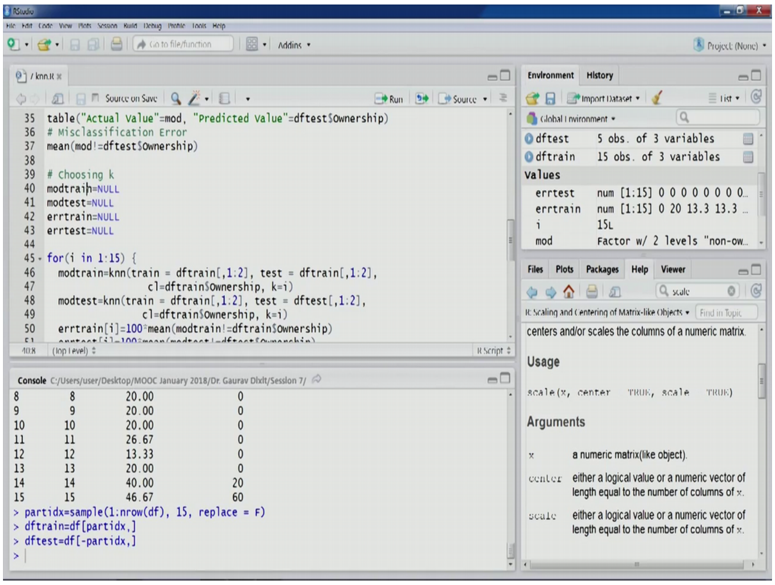
scale  either a logical value or a numeric vector of length equal to the number of columns of x.

So, these are the values. So, this is the output we have for value of K when the value of K is 1 you can see the training error is 0 and the validation error is 0. So, when value of K is 1. So, essentially we are going to use just 1 observation, if you look at the training. So, for any observation in the training set we are going to use just one close by observation right. So, this is yielding this result that error is 0. So, almost all the observation using the value of K they are mean correctly classified, the same case say same is applicable in error validation. So, there also as we can see that that 0 value is there typically this is not the case, this is mainly because of the way partitioning has been done.

So, if we look at the different values of k. So, you can see for value of K as two the error training is 20 then for 3 then 4 and then 5 the error keeps on increasing at has been increase the value of k. So, as we move from 1 to 15 the error in training partition keeps on increasing and if we look at the error validation it remains at 0. So, this is peculiarity about this particular data set this particular partitioning is actually been captured not the typical case, we would see only at the K value of K 14 and 15 we see the different errors. So, properly we need to run these partition do this partition partitioning and do this modelling again because the some peculiarity about the you know would be (Refer Time: 18:00) small data set just 5 observation in the validation. So, that is being captured and we are not getting a typical kind of result or closer to typical result. So, what will do will run the will execute the partitioning again and then again will execute the following code.

So, let us do this. So, again the 15 random we want to, we will like to have different 15 observation at any training partition in 5 observations. So, that we are able to get a typical result suppose, if we are using a small data set. So, that is why some peculiarity in the results can be same. So, let us execute this, you can see we are recreating these partitions and hopefully we get different partition this time and therefore, different result.

(Refer Slide Time: 19:02)



```
table("Actual Value"=mod, "Predicted Value"=dftest$Ownership)
# Misclassification Error
mean(mod!=dftest$Ownership)

# Choosing k
modtrain=NULL
modtest=NULL
errtrain=NULL
errtest=NULL

for(i in 1:15) {
  modtrain=knn(train = dftrain[,1:2], test = dftrain[,1:2],
                cl=dftrain$Ownership, k=i)
  modtest=knn(train = dftrain[,1:2], test = dftest[,1:2],
               cl=dftrain$Ownership, k=i)
  errtrain[i]=100*mean(modtrain!=dftrain$Ownership)
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
8       8      20.00        0
9       9      20.00        0
10     10      20.00        0
11     11      26.67        0
12     12      13.33        0
13     13      20.00        0
14     14      40.00       20
15     15      46.67       60
> partidx=sample(1:nrow(df), 15, replace = F)
> dftrain=df[partidx,]
> dftest=df[-partidx,]
> 
```

```
dftest      5 obs. of 3 variables
dftrain     15 obs. of 3 variables
Values
  errtest   num [1:15] 0 0 0 0 0 0 0 0...
  errtrain  num [1:15] 0 20 13.3 13.3 ...
  i         15L
  mod       Factor w/ 2 levels "non-ow...
```

centers and/or scales the columns of a numeric matrix.

Usage

```
scale(x, center = TRUE, scale = TRUE)
```

Arguments

x a numeric matrix(like object).

center either a logical value or a numeric vector of length equal to the number of columns of x.

scale either a logical value or a numeric vector of length equal to the number of columns of x.

So, we will have two null form here once again I am execute this code and lets construct this particular data frame again, now there is some change some change there, you can see some change you can notice there. So, still even if we look at these results the error training as you can see for different value of K this keeps on increasing. So, increases then side drop there and then again keeps on increasing with few drops and we are mainly interest in the validation error, we would see the ah that again in validation itself again we see some particularity in the in the results.

So, what we actually expect is that the error in validation would actually increase initially and then it would decrease and as been increase as be go long for more values of K it will increase further had. So, there is going to be an a minima right the. So, we are interest in the very first and minima that could be there.

So, because of the smaller data set we are no able to see those kind of results. So, what will do will try once more. So, as you can see I am z initializing these variables and we executing the loop and then again, now I thing yes. Now, this time we are getting more typical results yes this is more, more of a typical scenario. So, we have a good enough sample size probably this particular a result is replicating the regular scenario we have a larger sample size.

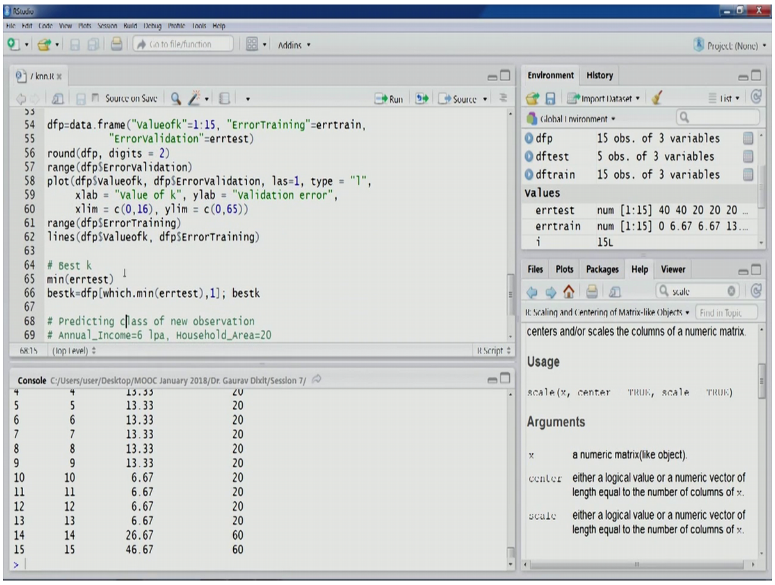
So, here you can see that error training it is starts from 0 right and then in the validation it start from a higher value and then as we increase the value of K it key, it goes down and it remains a again some peculiarity are still there are still it is it is better than the previous cases. So, its keeps on it decreases or remain same and then further it increases had the extreme points.

So, optimal value of K would actually be have would actually have to be identified as using the error in validation. So, the error in the validation would typically be more than the error in the training partition because we are able building the, we have build the model on the training partition and in this case K NN. So, essentially we are identifying the using the same observations to identify K neighbours and using those K neighbours we are trying to classifying the observation of the same partition training partition, you would see in the training partition start from 0 then error keeps on increasing then again (Refer Time: 22:35) and then further its increases.

So, typically validation will error will be start with the higher side and then will decrease and it will increase further. So, if we look at the data and then probably value of K equal to 3 is the optimal value where we see a direct dip in the error value from 40 for K of value of 1 and K value of 2, again the error is 40 then there is a dip and becomes 20.

So, further we keep on increasing these value of K the results the error result do not change, error rate does not change and later on as be reach to the value of K as 2 14 and 15 then again there is increase. So, let us use this particular result for our further discussion.

(Refer Slide Time: 23:32)



```
dfp=data.frame("Valueofk"=1:15, "ErrorTraining"=errtrain,
          "ErrorValidation"=errtest)
round(dfp, digits = 2)
range(dfp$ErrorValidation)
plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = "l",
     xlab = "Value of k", ylab = "Validation error",
     xlim = c(0,16), ylim = c(0,65))
range(dfp$ErrorTraining)
lines(dfp$Valueofk, dfp$ErrorTraining)

# Best k
min(errtest)
bestk=dfp[which.min(errtest),1]; bestk

# Predicting class of new observation
# Annual_Income=6 lpa, Household_Area=20
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
4         4       13.33          20
5         5       13.33          20
6         6       13.33          20
7         7       13.33          20
8         8       13.33          20
9         9       13.33          20
10       10        6.67          20
11       11        6.67          20
12       12        6.67          20
13       13        6.67          20
14       14       26.67          60
15       15       46.67          60
> 
```

```
dfp        15 obs. of 3 variables
dftest     5 obs. of 3 variables
dftrain    15 obs. of 3 variables
Values
  errtest    num [1:15] 40 40 20 20 20 ...
  errtrain   num [1:15] 0 6.67 6.67 13....
  i          15L
```

centers and/or scales the columns of a numeric matrix.

Usage

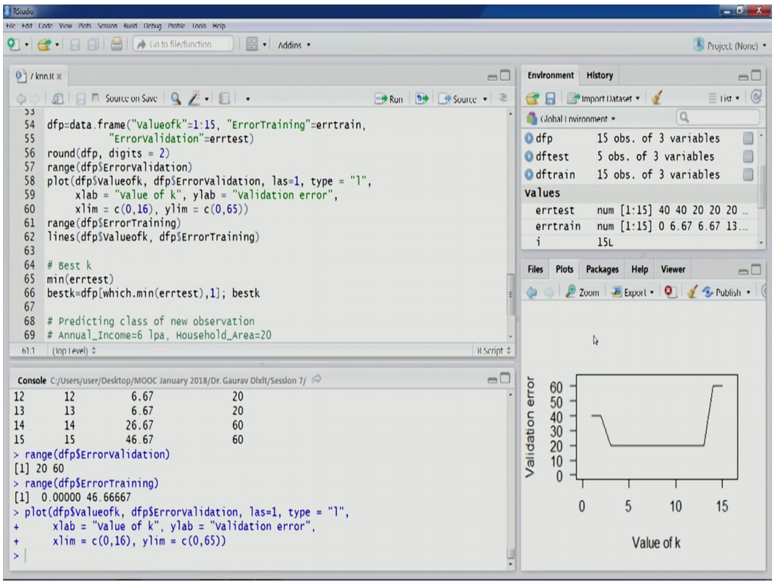
`scale(x, center = TRUE, scale = TRUE)`

Arguments

| | |
|---|---|
| x | a numeric matrix(like object). |
| center | either a logical value or a numeric vector of length equal to the number of columns of x. |
| scale | either a logical value or a numeric vector of length equal to the number of columns of x. |

So, we can draw a lot between the error rate and the value of k. So, we are mainly interested in the validation error. So, let us look at the range 20 to 60 and will have to specify this range here. So, appropriately covered 0 to 65 it would be covered there and 0 to 16 the limit on x axis it would be curved, if we look at the range of training error training and that is between 0 and 40 (Refer Time: 24:11) this will also lie in this range. So, this particular range for x and x axis and y axis they are appropriately specified. So, let us generate this scatter plot.

(Refer Slide Time: 24:23)



```
dfp=data.frame("Valueofk"=1:15, "ErrorTraining"=errtrain,
               "ErrorValidation"=errtest)
round(dfp, digits = 2)
range(dfp$ErrorValidation)
plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = "l",
     xlab = "Value of k", ylab = "Validation error",
     xlim = c(0,16), ylim = c(0,65))
range(dfp$ErrorTraining)
lines(dfp$Valueofk, dfp$ErrorTraining)

# Best k
min(errtest)
bestk=dfp[which.min(errtest),1]; bestk

# Predicting class of new observation
# Annual_Income=6 lpa, Household_Area=20
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
12        12          6.67              20
13        13          6.67              20
14        14         26.67              60
15        15         46.67              60
> range(dfp$ErrorValidation)
[1] 20 60
> range(dfp$ErrorTraining)
[1]  0.00000 46.66667
> plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = "l",
+      xlab = "Value of k", ylab = "Validation error",
+      xlim = c(0,16), ylim = c(0,65))
> |
```

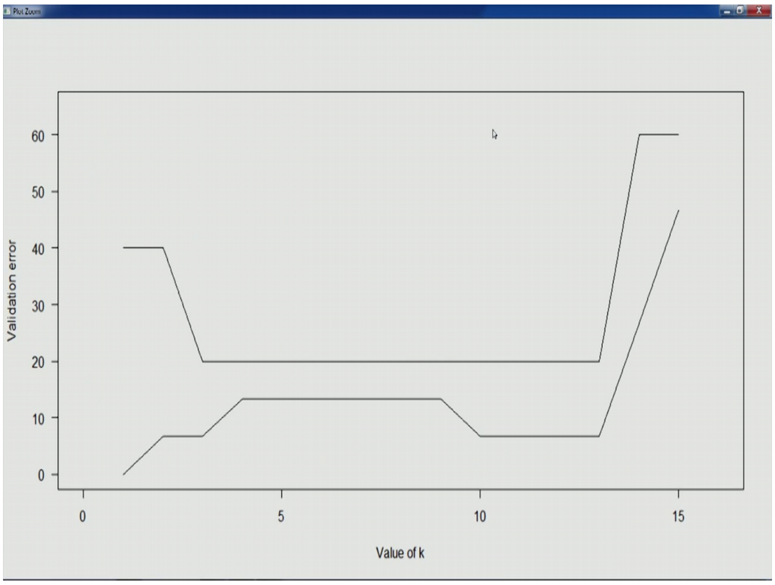
Environment

```
dfp        15 obs. of 3 variables
dftest     5 obs. of 3 variables
dftrain    15 obs. of 3 variables
Values
  errtest    num [1:15] 40 40 20 20 20 ...
  errtrain   num [1:15] 0 6.67 6.67 13....
  i          15L
```

Now, this is the scatter plot or validation error. So, this typically would be more is smooth if there are more num, you know more numb more values of K to choose from. So, in that cases we if we plot smaller number of smaller number of values of K then again the plot would look like this, but if we have large number of values for K to pick from then and then larger sample size then the error values will also smooth out. So, this curve would be you know much more smooth that is also a plot, that also at the curve for training partition.

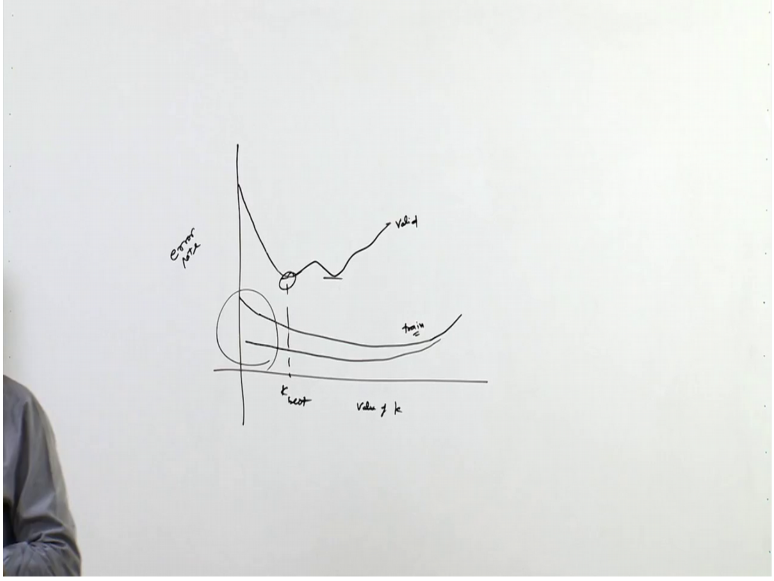
Now, this is the curve (Refer Time: 25:12) look at this lets zoom in.

(Refer Slide Time: 25:15)



If we look at the plot we would see that in the training case as we start from 0 the error is on the lower side and as be keep on increasing the value of K, the error keeps on increasing right few dips here in there and then its keeps on increasing. So, if we look at the validation, the validation error is always about the training partition error right. So, we start from higher value then it decreases right. So, this is this kind of curve that we are getting this is mainly because of this smaller data set that we are using, if we want to know typical curve that would be actually expect is would be something like this.

(Refer Slide Time: 26:02)



So, probably if you are having a larger data set. So, what is expected is that these you know value of k. So, different values of K on x axis and here you have error rate. So, this could be on the validation partition or on the training partition. So, typically the training error rate on training partition would be on the would be smaller in comparison to error rate on the validation partition.

So, validation partition it will be start from higher side and has been increase the value of K it will probably decrease and it will reach to the minima optimal minima and then further it might increase and in that fashion it might go. So, this typically this might for applicable for validation, if we look at the, if we want to understand the curve for training partition. So, again be we believe that for low value of K our model right our K NN model if it is starts fitting the noise then even for low values of K as well the error should be on the higher side.

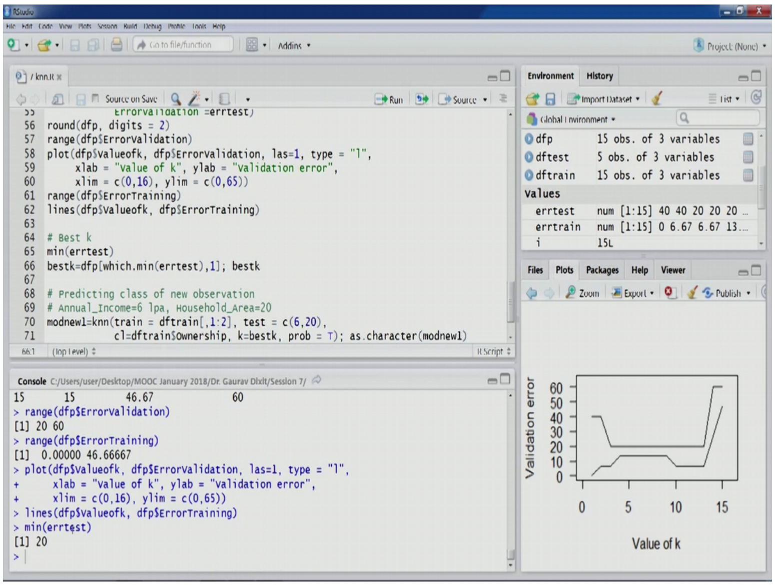
So, but essentially when we are talking about the training partition. So, especially in K NN case there are is going to much lower right. So, it might start from where between this range right and the as we move further, as we move further. So, the error it will it will come down, it will come down right and as we further for more values of K again it will come up, so this would be mainly applicable for training.

Now, in this particular range if, in this particular range for value of K starting from K 0 to starting from K 1 to few smaller values of K if we are not fitting the noise then probably this particular curve actually draw further at. So, this might be like this. So, different variation for this particular depending on the, depending on the data set and when you on the modeling excise that we have depending on the predictors information that we have and how much the model is able to how much better the model is able to classify.

So, this is typical scenario that we expect which is in a sense in a rough sense is reflected here in this particular curve, but we have to you know do many iterations to reach to this particular result, design being we are dealing with smaller data set and therefore, for this is small data set sometimes the results might not be as expected in a regular scenario. So, it is therefore, recommended that a larger data set should actually be used in a data mining excises.

So, that we are able to get the usual result and then the analyse the same. Now, once this is done, once you have a table of this kind where you have the error values. So, you would interested. So, in this particular case we are just dealing with 15 values, 15 different values of k. So, it is easier to spot the best value of K out of all these values. So, it is probably the K is the value is 3. So, it easier to spot in this case; so, probably we the best value of K is 3, but if you are having a you know a large number of you know we are trying of this particular excises for large number of values you can use this particular function minimum and find out the minimum of this values we can see this is 20.

(Refer Slide Time: 30:12)



```
              ErrorValidation =errtest)
round(dfp, digits = 2)
range(dfp$ErrorValidation)
plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = "l",
     xlab = "Value of k", ylab = "Validation error",
     xlim = c(0,16), ylim = c(0,65))
range(dfp$ErrorTraining)
lines(dfp$Valueofk, dfp$ErrorTraining)

# Best k
min(errtest)
bestk=dfp[which.min(errtest),1]; bestk

# Predicting class of new observation
# Annual_Income=6 lpa, Household_Area=20
modnew1=knn(train = dftrain[,1:2], test = c(6,20),
            cl=dftrain$Ownership, k=bestk, prob = T); as.character(modnew1)
```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
15       15         46.67              60
> range(dfp$ErrorValidation)
[1] 20 60
> range(dfp$ErrorTraining)
[1]  0.00000 46.66667
> plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = "l",
+      xlab = "Value of k", ylab = "Validation error",
+      xlim = c(0,16), ylim = c(0,65))
> lines(dfp$Valueofk, dfp$ErrorTraining)
> min(errtest)
[1] 20
> 
```

Global Environment

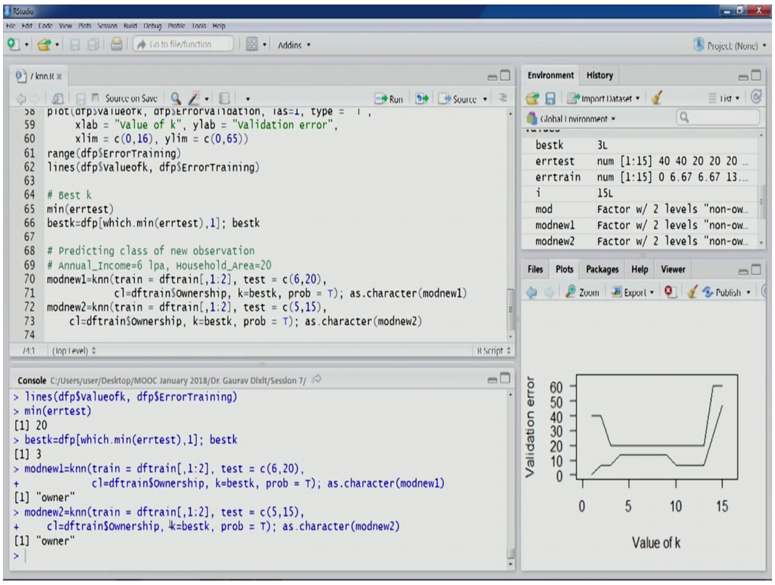
```
dfp        15 obs. of 3 variables
dftest     5 obs. of 3 variables
dftrain    15 obs. of 3 variables
Values
  errtest    num [1:15] 40 40 20 20 20 ...
  errtrain   num [1:15] 0 6.67 6.67 13....
  i          15L
```

Now, for this particular value we would like to find out the low lowest value of K for which this particular error was achieved. So, for different value of K this particular error could be the lowest error there could be many minima there could be may many minima points. So, this particular graph could also be different depending on this different data set scenario to grow come up and down and this is in this fashion.

So, there could be 2 such points. So, how to we find out the best value of K? So, it is the generally the first 1 it is generally the first one that be would like to pick. So, this is the value. So, this is the value of K that would we would like to pick because here it might we fitting to noise or some peculiarity in the data set. So, it is generally the first one that you fit.

So, we can do this using this particular code. So, in the brackets you can see for in the row side we are trying to identify the index where this particular value would be this (Refer Time: 31:23) value would be there right which dot mean and the very first. So, for different values indices of early rows where this value of 20 is there they would be there in the result, but we want to pick the first 1. So, that would be selected here. So, would see that 3 as been selected which be also had identified by looking at the table itself. So, we are dealing with smaller data set. So, this it was easier for us to do that.

(Refer Slide Time: 31:58)



```
plot(dfp$Valueofk, dfp$ErrorValidation, las=1, type = 'l',
     xlab = "Value of k", ylab = "Validation error",
     xlim = c(0,16), ylim = c(0,65))
range(dfp$ErrorTraining)
lines(dfp$Valueofk, dfp$ErrorTraining)

# Best k
min(errtest)
bestk=dfp[which.min(errtest),1]; bestk

# Predicting class of new observation
# Annual_Income=6 lpa, Household_Area=20
modnew1=knn(train = dftrain[,1:2], test = c(6,20),
            cl=dftrain$Ownership, k=bestk, prob = T); as.character(modnew1)
modnew2=knn(train = dftrain[,1:2], test = c(5,15),
   cl=dftrain$Ownership, k=bestk, prob = T); as.character(modnew2)

```

Console C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 7/

```
> lines(dfp$Valueofk, dfp$ErrorTraining)
> min(errtest)
[1] 20
> bestk=dfp[which.min(errtest),1]; bestk
[1] 3
> modnew1=knn(train = dftrain[,1:2], test = c(6,20),
+             cl=dftrain$Ownership, k=bestk, prob = T); as.character(modnew1)
[1] "owner"
> modnew2=knn(train = dftrain[,1:2], test = c(5,15),
+    cl=dftrain$Ownership, k=bestk, prob = T); as.character(modnew2)
[1] "owner"
> 
```

```
bestk      3L
errtest    num [1:15] 40 40 20 20 20 ...
errtrain   num [1:15] 0 6.67 6.67 13....
i          15L
mod        Factor w/ 2 levels "non-ow...
modnew1    Factor w/ 2 levels "non-ow...
modnew2    Factor w/ 2 levels "non-ow...
```

Now, once this excises is done and we have identified the optimum value of K, now we can also see how we can actually use this optimum value of K and predict the class of a new observation. So, the in the is scatter plot that we had generated we had use this particular new observation anural income 6 and house hold area 20. So, how do we how we can score this particular observation how we can classify this observation.

So, again you can we are using K NN function and training train first argument is the training you know partition that we have specified, then in the test we have a specified just to one point it is not the test partition. It is just the new observation 6 and 20 using in combined function and then the class is for the third argument, it is the classes ownership variable from the training partition then K value as you can see we have picked up the best K value and let us look at the classification.

So, this particular observation would be classified as owner as for the our model, if for a different point let us say 5 or 15 this observation having income as 5 and area is 15. So, let us run for this as well let us score this one also this one also would be specified as owner.

So, in this fashion for its specific points we can classified m using K NN and we also had a look on how for a test partition we can a score it if you looked at how we can identify the optimum value of k. So, we will start will stop here and will start our discussion then ne next lecture of the remaining points of K NN and also how K NN could be used for the prediction task.

Thank you.