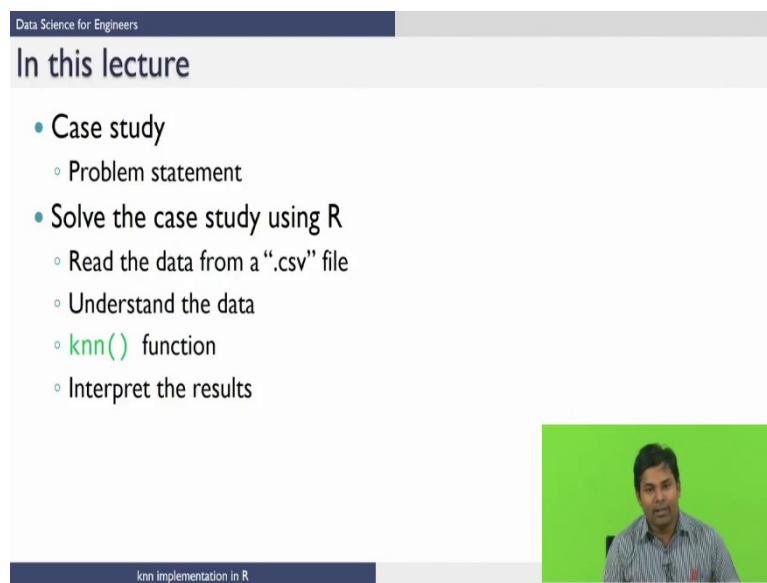


Data science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 47
K- nearest neighbours implementation in R

Hello all, welcome to this lecture on K-nearest neighbours implementation in R.

(Refer Slide Time: 00:23)



The slide is titled "In this lecture" and is part of a presentation for "Data Science for Engineers". It contains a bulleted list of topics to be covered in the lecture:

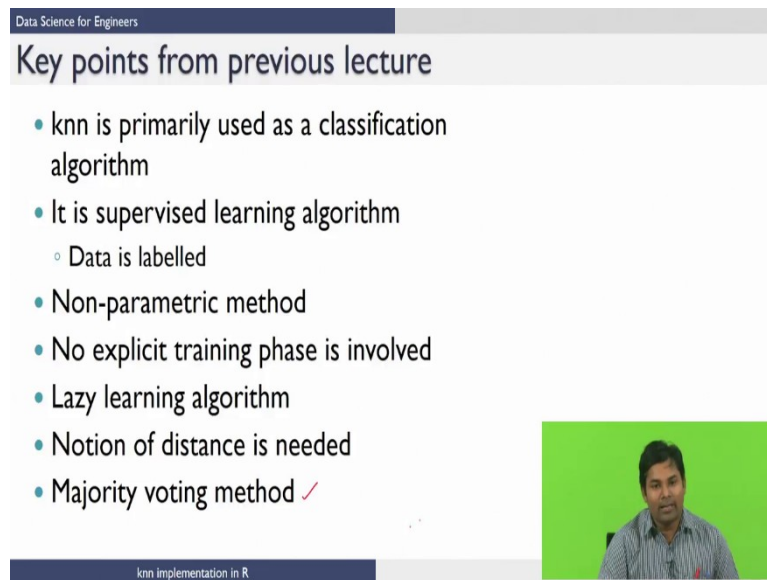
- Case study
 - Problem statement
- Solve the case study using R
 - Read the data from a ".csv" file
 - Understand the data
 - `knn()` function
 - Interpret the results

In the bottom right corner of the slide, there is a small video inset showing a man with dark hair, wearing a striped shirt, speaking against a green background. The text "knn Implementation in R" is visible in the bottom left corner of the slide.

In this lecture, what we are going to do is to introduce you to a case study which we use as a means to explain how to implement this knn algorithm in R. We will start with the problem statement of the case study and we will show how to solve this case study using R.

In the process we will show how to read the data from dot csv file, how to understand the data that is being loaded into the workspace of R and how to implement this K-nearest neighbours algorithm in R using this knn function. And we will also talk about how to interpret the results that this knn algorithm gives to us.

(Refer Slide Time: 01:12)



Data Science for Engineers

Key points from previous lecture

- knn is primarily used as a classification algorithm
- It is supervised learning algorithm
 - Data is labelled
- Non-parametric method
- No explicit training phase is involved
- Lazy learning algorithm
- Notion of distance is needed
- Majority voting method ✓

knn implementation in R

Before we jump into the case study, let us review some key points from the previous lecture of Prof. Raghu. If you remember knn is primarily used as a classification algorithm. It is a supervised learning algorithm. When I say supervised learning algorithm that means the data that is provided to you has to be labelled data and knn is a non-parametric method. So, what do you mean by this non-parametric method is that there is no extraction of the parameters of the classifiers from the data itself. And there is no explicit training phase involved in this knn algorithm.

And the knn algorithm is a lazy learning algorithm because it would not do any computations till you ask you to do classification. Because we are dealing with the K-nearest neighbours we would have seen this notion of distance is important when we are dealing with this knn algorithm. And the way the knn algorithm works is by the majority voting method. That means if you give a test point, we calculate the distance of the test point from all the data points in the given data and arrange them in the ascending order and we choose the k first nearest neighbours. And based on the voting that each of them will give for this test data, we will assign the class to the test data point. That is how essentially the knn works.

Now, let us define the case study problem statement. We have named this case study as automotive service company case study.

(Refer Slide Time: 03:07)

The screenshot shows a presentation slide with a dark blue header containing the text 'Data Science for Engineers'. The main title of the slide is 'Automotive Service Study: Problem statement'. The body of the slide contains the following text: 'An automotive service chain is launching its new grand service station this weekend. They offer to service a wide variety of cars. The current capacity of the station is to check 315 cars thoroughly per day. As an inaugural offer, they claim to freely check all cars that arrive on their launch day, and report whether they need servicing or not! Unexpectedly, they get 450 cars. The service men won't work longer than the working hours but the data analysts have to! Can you save the day for the new service station?' To the right of the text is a small video inset showing a man with dark hair and a beard, wearing a light blue shirt, speaking against a green background. At the bottom of the slide, there is a dark blue footer with navigation icons and the text 'knn implementation in R'.

Let us look at the problem statement. An automotive service chain is launching its grand new service station this weekend. They offer service to wide variety of cars. The current capacity of the station is to check the 315 cars thoroughly per day. As an inaugural offer, what they have done is, they claim to freely check all the cars that arrive on their launch day and they said they will report whether they need servicing or not.

What happened is unexpectedly, they got 450 cars. Now, since they have the testing facility for testing only 315 cars, they will not be able to check all the 450 cars very thoroughly and the servicemen will not work longer than the normal working hours.


So, what they have done is they have hired a data analyst to help them out from the situation. If you are the data analyst which is hired by this automotive service station person, how can you save the day for this new service station is the problem statement.

(Refer Slide Time: 04:34)

Data Science for Engineers

How can a data scientist save a day for them?

- He has been a data set which contains some attributes of car that can be easily measured and wont require much time and a conclusion that if service is needed for that or not. - "serviceTrainData.csv" ✓
- Now for the cars they cannot check in detail, they measure those attributes- "serviceTestData.csv" ✓
- Use knn classification technique to classify the cars they cannot test manually and say whether service is needed or not .

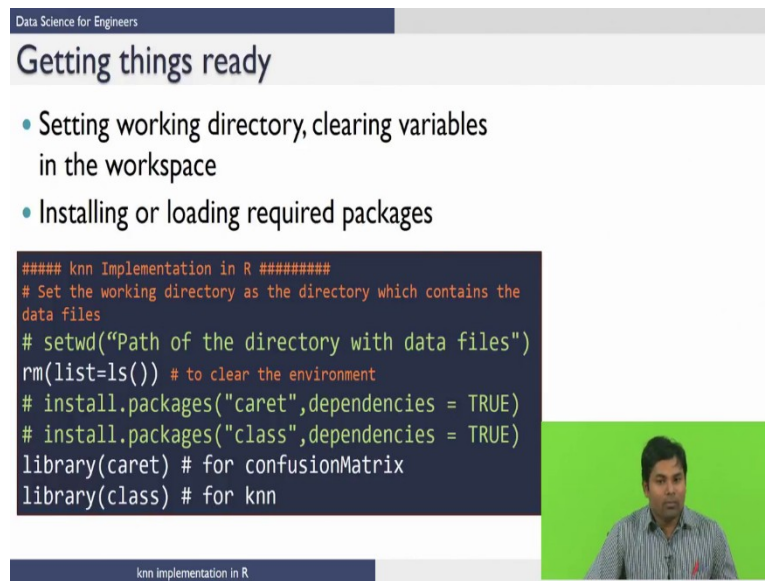


knn implementation in R

Now, let us see how a data scientist can save a day for this service station people. Since, service station has capacity to thoroughly check 315 cars, they have thoroughly checked all the 315 cars and given the data in this service traindata.csv. Now, for the rest of the cars among the 450, they cannot thoroughly check all the data and they have checked only those attributes which are easily measurable and they have given them in this service test data dot csv. So, essentially the data scientist has data which is like a training data for him which contains few attributes and with a label whether a service is needed or not.

And he also has a data for which now all the other attributes are present, he do not have this column where whether the service is needed or not. The idea here is how do one use this data, service train data, to comment upon for the readings which present which are present in the service test data to tell whether service is needed or not in this case. So, the idea is to use this knn classification technique to classify the cars in the service test data le which cannot be tested manually and say whether service is needed or not. Now, let us see how do you solve this case study in R.

(Refer Slide Time: 06:26)



The slide is titled "Getting things ready" and is part of a presentation on "Data Science for Engineers". It contains a bulleted list of two tasks: "Setting working directory, clearing variables in the workspace" and "Installing or loading required packages". Below the list is a code block with R code for knn implementation. To the right of the code block is a small video inset showing a man speaking. The slide footer says "knn implementation in R".

```
##### knn Implementation in R #####
# Set the working directory as the directory which contains the
data files
# setwd("Path of the directory with data files")
rm(list=ls()) # to clear the environment
# install.packages("caret",dependencies = TRUE)
# install.packages("class",dependencies = TRUE)
library(caret) # for confusionMatrix
library(class) # for knn
```

First you have to get things ready. When I say get things ready I mean you have to set the working directory as the directory in which the given data files are available. That you can do using set working directory command and the corresponding path you can give here. Otherwise you can use GUI option also to set the working directory. And this command here is used to clear all the variables in the environment of R. You can very well use the brush button in the environmental history pan to clear the variables in the workspace.

And another important thing one has to do is, for this knn implementation, we need two external packages which are caret and class, one has to install this caret and class packages if they have not installed it already. So, the way to install this packages we have explained in our R modules, you can install the packages through the command window using this command install dot pack-ages and the package name and say dependencies = true or you can use the GUI to install the packages. So, please install this packages caret and class. And once you install, you can load those packages using the library command as we have explained already. We will see why is this packages important as we go along this lecture.

And library caret is for generating the confusion matrix which Prof. Raghu would have talked about when he is talking about this performance matrix of a classifier. And this library class is a library which contains different classification algorithms. And here we are going to use it for implementing this knn. Now, let us see how to read the data.


(Refer Slide Time: 08:32)

Data Science for Engineers

Reading the data

- Data for this case study is provided to you in files with names “serviceTrainData.csv”, “serviceTestData.csv”
- To read the data from a “.csv” file we use `read.csv()` function

knn implementation in R



From the given les and for this case, a data is being provided in two les as we have already seen servicetraindata.csv and service test data dot csv. So, in order to read this data from the csv files, function we use is read.csv function. Let us look what this read dot csv function takes and what it returns.

(Refer Slide Time: 08:59)

Reads a file in table format and creates a data frame from it
SYNTAX

```
read.csv(file,row.names)
```

file	the name of the file which the data are to be read from. Each row of the table appears as one line of the file.
row.names	a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names.

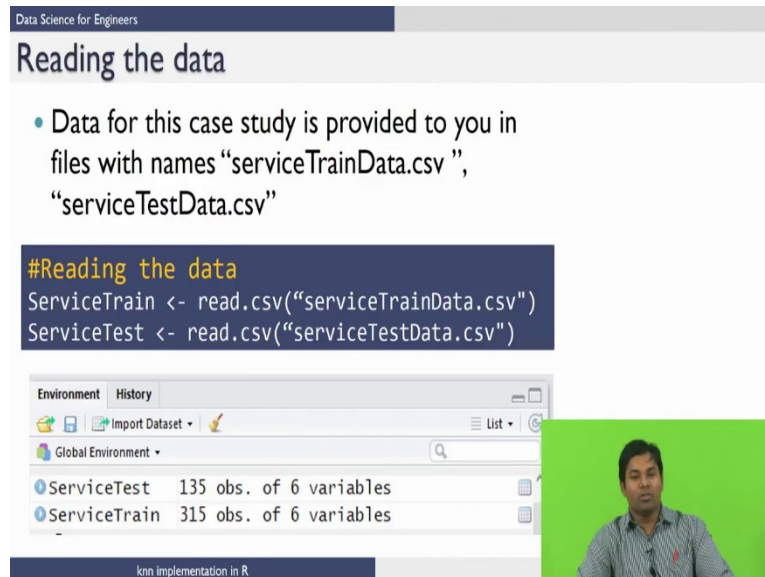
knn implementation in R



This read dot csv file reads a file in a table format and creates a data frame from it. The syntax for this read dot csv function is as follows, read dot csv the filename, and the row names. Let us look at what this input arguments le and row dot names means. File is essentially the name of the file from which you have to read the data. And row dot names is a vector of row names, it can be either a vector giving the actual row names or a single value which specifies what column of the

data set is having the row names. Let us see how to read the data in this particular case.

(Refer Slide Time: 09:48)



The slide is titled "Reading the data" and is part of a presentation for "Data Science for Engineers". It contains a bullet point stating that data for the case study is provided in files named "serviceTrainData.csv" and "serviceTestData.csv". Below this, a code block shows the R commands to read these files:

```
#Reading the data
ServiceTrain <- read.csv("serviceTrainData.csv")
ServiceTest <- read.csv("serviceTestData.csv")
```

. At the bottom, a screenshot of the R Studio environment shows two data frames loaded: "ServiceTest" with 135 observations and 6 variables, and "ServiceTrain" with 315 observations and 6 variables. A small video inset of a presenter is visible on the right side of the slide.

As we have seen the data has been given in this two dot csv files, we can use read dot csv function to read the data. As we have seen in the syntax of read.csv we have to give the filename that is the filename service train dot data from which I want to load the data. I will give this file name. And I am assigning this to a variable called service train when you execute this command what happens is, it reads a data from the service train data file and assigns it to this variable which is of the form data frame.

Similarly, you will read the data from service test data and assign it to variable service test which is again a data frame. In the R environment, once you execute these commands you will see two data frames which are service test and service train which are having this 315 observations of 6 variables and 135 observations of 6 variables.

Remember why this 315? 315 is the number of cars that they can thoroughly check, but they have given in this 315 the 6 variables are the attributes which are easily measurable and one column which says whether service is needed or not. And this 135 cars they have 6 variables, they have measured all the 5 attributes which are important and the 6 attribute is also given here we will see why the 6 attribute is given and so on as we go on in this lecture.

Now, let us see what is there in this service train and service test data. One way to see what is there in this service test and service train is to use the view command.

(Refer Slide Time: 11:42)

Data Science for Engineers

Viewing the data

`View(ServiceTest)`

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins


knn_implementation.R ServiceTrain ServiceTest

Filter

	OilQual	EnginePerf	NormMileage	TyreWear	HVACwear	Service
1	45.773338	49.936615	49.777581	48.263851	50.95267173	No
2	4.987185	7.891003	6.588986	9.493161	3.24026216	No
3	4.987185	4.891003	7.308986	8.373161	2.78026216	No
4	106.388821	104.454032	103.051485	106.282958	105.53684290	No
5	104.388821	103.744032	103.051485	106.132958	105.77684290	No
6	4.987185	4.891003	5.618986	8.373161	1.76026216	No
7	45.533338	50.696615	48.167581	50.633851	47.95267173	No
8	27.765516	29.138295	31.259536	31.226162	31.31127506	Yes
9	26.765516	28.418295	30.809536	29.266162	31.31127506	Yes
10	104.388821	103.744032	105.051485	106.212958	104.24684290	No
11	4.987185	5.891003	7.228986	8.373161	1.08026216	No
12	104.388821	103.434032	104.051485	106.062958	105.53684290	No

Showing 1 to 12 of 135 entries

knn implementation in R



This view command helps you to see the data frames. For example, if you want to see what is there in the service train data frame, what you have to do is this view service train will show a table like this in your editor environment. Now, you can see that there are how many attributes 1, 2, 3, 4, 5, 6 attributes. And if you see these are the five attributes which are measured for testing whether the service is needed or not, and this attribute is basically saying if service is needed or not.

Similarly, you can see for the service test data set which is shown here. For now, what we assume is will act such a way that we do not know this column and we will come back to this. Now, if you observe here, there are 135 entries for which they have not thoroughly checked they just measured this 6 quantities and they want to figure out whether service is needed or not using the knn algorithm that is the whole idea. Since, you have viewed what is there in this service test and service train data sets, now is there any way to know what are the data types of the these attributes that are there in this service train and service test is the next question that comes to mind. Now, let us understand the data and little more detail


(Refer Slide Time: 13:12)

Data Science for Engineers

Understanding the data

- ServiceTrain contains 315 observations of 6 variables
- ServiceTest contains 135 observations of 6 variables
- The variables are: OilQual, Engineperf, NormMileage, TyreWear, HVACwear and Service
 - First five columns are the details about the car and last column is the label which says whether a service is needed or not

knn implementation in R



What we have seen till now is the service train contains 315 observations of six variables, service test contains 135 observations in 6 variables. And variables that are present in the data sets are oil quality, engine performance, normal mileage, tyre wear, HVAC wear and service. And I as I mentioned earlier this 5 are the attributes that tells about the condition of the car. And this attribute simply says whether service is needed or not that is what here.

First five columns are the details about the car and the last column is the label which says whether a service is needed or not. Now, let us ask this question what are the data types of each of these attributes, how one get the data types of the attributes that are there in the data.

So, since we have understood the data now. Let us look at what is the structure of the data.

(Refer Slide Time: 14:11)

Data Science for Engineers

Structure of the data


- Structure of data
 - Variables and their data types
- **str()**
Compactly display the internal structure of an R object

SYNTAX

str(object)

object	any R object about which you want to have some information.
--------	---

knn implementation in R



When you say structure of data what do we mean by that is in the data set you have what are the variables that are there, and what are their data types. So, the way you get the structure of data in R is using this structure function. What does this structure function do, structure function compactly display the internal structure of an R object. The syntax for the structure function is as follows. Structure function takes one input argument which is an object. What is this object, this object is essentially any R object about which you want to have some information.

Now, let us see the structure of two data frames what we have read from the two dots csv files.


(Refer Slide Time: 14:58)

Data Science for Engineers

Structure of ServiceTrain

```
> str(ServiceTrain)
'data.frame': 315 obs. of 6 variables:
 $ OilQual : num 103.4 26.8 62.4 45.5 104.4 ...
 $ EnginePerf : num 103.5 26.2 63.7 49.9 103.3 ...
 $ NormMileage: num 103.1 31.3 59.7 48.8 103.1 ...
 $ TyreWear : num 106.2 29.2 64.7 48.1 105.8 ...
 $ HVACwear : num 105.7 31.3 58.6 48 106.5 ...
 $ Service : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 1 1
```

knn implementation in R



You can see the structure of the service train data frame. Here, if you execute this command structure of service train, what it gives is the following information which says service train is a data frame which contains 315 observations of six variables. And the variables are oil quality, engine performance and so on. And they will say the data type of all this five attributes is numeric, and the last attribute service is a factor with two levels that means we have yes or no in this attribute. And this one two represents each entry for example one corresponds to no and two corresponds to yes and so on. Let us use the structure command on the service test data and see what it has.


(Refer Slide Time: 15:54)

Data Science for Engineers

Structure of ServiceTest

```
> str(ServiceTest)
'data.frame': 135 obs. of 6 variables:
 $ OilQual : num 45.77 4.99 4.99 106.39 104.39 ...
 $ EnginePerf : num 49.94 7.89 4.89 104.45 103.74 ...
 $ NormMileage: num 49.78 6.59 7.31 103.05 103.05 ...
 $ TyreWear : num 48.26 9.49 8.37 106.28 106.13 ...
 $ HVACwear : num 50.95 3.24 2.78 105.54 105.78 ...
 $ Service : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2
```

knn implementation in R



This is the output you see when you execute this command here. It says the service test is also data frame which contains 135 observations in 6 variables. These are the variables that are available. The first 5 variables are numeric type variables, and the service variable is a factor with two levels which contains yes or no.

Since we have seen the structure of the data, let us ask this question is there any way that I will get a summary of the data which I have read.

(Refer Slide Time: 16:27)

Data Science for Engineers


Summary of the data

- Summary of data
 - The function invokes particular methods which depend on the class of the first argument.
- **summary()**
Summary gives a 5 point summary for numeric attributes in the data
SYNTAX

summary(object)

object	any R object about which you want to have some information.
--------	---

knn implementation in R



The answer is yes, you can get. The summary of data is obtained by the summary function. Essentially what it does is it invokes particular methods depending upon the class of the argument that goes along with this summary function. For example, summary function gives a 5 point summary for numeric attributes in the data. Syntax for the summary function is as follows. The summary function takes one argument

which is an object. This object is any R object about which you want to get some information.

Let us use the summary function on our data frames which we have loaded and see what the results are.

(Refer Slide Time: 17:15)

Data Science for Engineers


Summary of ServiceTrain

```
> summary(ServiceTrain)
```

OilQual	EnginePerf
Min. : 0.9872	Min. : 1.891
1st Qu.: 26.7655	1st Qu.: 27.418
Median : 59.6633	Median : 59.741
Mean : 59.6493	Mean : 60.306
3rd Qu.:104.3888	3rd Qu.:103.744
Max. :106.4288	Max. :105.744

NormMileage	TyreWear
Min. : 3.359	Min. : 6.213
1st Qu.: 31.260	1st Qu.: 29.036
Median : 57.221	Median : 60.304
Mean : 60.297	Mean : 61.759
3rd Qu.:103.051	3rd Qu.:106.173
Max. :105.051	Max. :108.173

HVACwear	Service
Min. : -1.72	No :232
1st Qu.: 31.34	Yes: 83
Median : 60.62	
Mean : 60.39	
3rd Qu.:105.54	
Max. :107.54	



knn implementation in R

So, when we execute this command summary of service train, you will get the details about all the numeric variables which are 5 point summaries including mean and for the service variable which is the categorical variable it gives how many no's are there in that particular attribute and how many yes values are there in that particular attribute.

(Refer Slide Time: 17:45)

Data Science for Engineers


Summary of ServiceTest

```
> summary(ServiceTest)
```

OilQual	EnginePerf
Min. : 2.597	Min. : 1.891
1st Qu.: 26.696	1st Qu.: 27.418
Median : 61.023	Median : 61.501
Mean : 58.629	Mean : 59.077
3rd Qu.:104.229	3rd Qu.:103.744
Max. :106.389	Max. :105.744

NormMileage	TyreWear
Min. : 3.589	Min. : 6.143
1st Qu.: 31.260	1st Qu.: 28.901
Median : 59.351	Median : 61.304
Mean : 59.118	Mean : 60.864
3rd Qu.:103.051	3rd Qu.:106.173
Max. :105.051	Max. :108.173

HVACwear	Service
Min. : -1.72	No :99
1st Qu.: 31.31	Yes: 36
Median : 62.62	
Mean : 58.99	
3rd Qu.:105.33	
Max. :105.83	



knn implementation in R

You can use the same summary on service test and you can see that it will return you the 5 point summary for all the numeric variables and

it will return you the number of no values and yes values in the service test.

Let us keep this number in mind we have 99 no values and 36 yes values in the service test. As I said earlier, we are going to act in such a way that we do not know the true yes and no values and we use knn to predict which of them are yes and which of them are no.

(Refer Slide Time: 18:20)

Data Science for Engineers


Implementation of k-nearest neighbours: `knn()`

knn(train, test, cl, k = 1)

Arguments

train	matrix or data frame of training set cases.
test	matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.
cl	factor of true classifications of training set
k	number of neighbours considered.

knn Implementation in R



Now, let us do the important task as far as this lecture is concerned which is implementation of K-nearest neighbours in R. As I said earlier, the function which we use to implement this K-nearest neighbours is knn function. This knn function takes several arguments but I have listed few which are very important as far as this course is concerned. The arguments it takes are train, test, cl and k.

Let us see what each of this mean. Train is essentially a matrix or a data frame of the training set cases. That means you need to give all the data. In this case this is our service train data frame. And this test is a matrix or data frame for the test set cases. In this case, what will be our test matrix or a data frame this will be our service test data frame. This cl is a factor of true classifications of a training set and this k is the important parameter which is the number of neighbours that are needed to be considered while you do this algorithm which works on this majority voting criteria.

Now, let us implement this knn on our data. How do you do that? So, the way you do it is as follows.

(Refer Slide Time: 19:51)

Data Science for Engineers


Applying knn algorithm on data

```
# Applying k-NN algorithm
# K Nearest neighbour is a lazy algorithm and can do prediction directly with the testing
dataset, command "knn", accepts training and testing datasets the class variable of interest
i.e outcome categorical variable is provided for the parameter "cl". parameter "k" is to
specify the number of nearest neighbours required.

predictedknn <- knn(train = ServiceTrain[,-6],
                    test = ServiceTest[,-6],
                    cl = ServiceTrain$Service,
                    k = 3)
```

- ServiceTrain[,-6] gives information in ServiceTrain except the last column
- ServiceTest[,-6] gives information in ServiceTest except the last column
- ServiceTrain\$Service gives the last column of training data as a classification factor to the algorithm

knn implementation in R



There are certain comments here, let us study what those comments are. So, as we have seen in the previous lecture, K-nearest neighbour is a lazy algorithm and can do prediction directly with the testing data set. It accepts training and testing data sets and the class variable of interest that is outcome categorical variable and the parameter k as I have mentioned is to specify the number of nearest neighbours that are to be considered for the classification.

So, the way I implement this knn algorithm is through this knn command. As a training data set I will give all my service train dataset. Remember I have a negative 6 here, I will talk about it while later. And the test data set what I have given is the attributes in the service test except the 6th column. And in the class variables, I have given this 6th column as my classification parameter.

And let us say I want to build a knn which takes the number of nearest neighbours as 3. So, these are the input arguments for this knn function. When I execute this whole command here, it will calculate the labels for the test data set and store them in this predicted knn. I will show you the results in the coming slide.

Meanwhile let us interpret the service train a square bracket and - 6 means this if you remember since service train is a data frame from a data frames lectures. The statement here means that in the service train data frame take all the rows and exclude column 6 that is what it says. This command here gives information in service train except the last column. Similarly, this command here gives the information in the service test except the last column and service train dollar symbol service gives the last column of the training data as a classification factor for the algorithm.

Once you give all these parameters, execute this. The knn will classify the test data points and then store the labels in this predicted knn. Let us look at the results and what this predicted knn contains. (Refer Slide Time: 22:32)

Data Science for Engineers


Results: predicted classes

- "predictedknn" is the output from the algorithm, which has a categorical variable "Yes" or "No", indicating whether service is needed or not for each case in Test data

```

> # printing the information in predictedknn
> predictedknn
 [1] No No No No No No No Yes Yes No No
 [12] No No No No No Yes No No Yes Yes No
 [23] Yes No No No No No No No No No No
 [34] No No Yes No No No No No Yes No Yes
 [45] No No No Yes Yes No Yes No Yes No No
 [56] No No No No No No No No No Yes Yes
 [67] Yes No Yes No No Yes No No No No No
 [78] No Yes Yes Yes Yes No Yes No No Yes Yes
 [89] Yes No No No Yes No Yes No No No No
 [100] No No No No No No Yes No No No No
 [111] No Yes No Yes No Yes Yes No Yes No No
 [122] No No No Yes No No No No No No No
 [133] Yes No No
 Levels: No Yes

```



knn implementation in R

So, as we have seen in the earlier slide, predicted knn is the output from the algorithm, which has categorical variable yes or no indicating whether service is needed or not for each case in the test data. When you print this predicted knn, this is the output you see. It essentially says in this 135 values you have, first car no service is needed, and second car no service is needed, and for the 23rd car service is needed and so on.

So, that is what this knn algorithm does and you have actually nished your job of classifying the test cars as whether the service is needed or not. When you do not have this luxury of knowing the true value this is where you stop. But in R case what happened is, we already have the true values whether service is needed or not for this data set what we have. Now, when you have this luxury of knowing the true classes, you can generate what is called confusion matrix and see how well you have classified this performing.

(Refer Slide Time: 23:51)

Data Science for Engineers

Results: generating confusion matrix manually


```
# Command to develop and print a confusion matrix
conf_matrix = table(predictedknn,ServiceTest[,6])

predictedknn No Yes
No 99 0
Yes 0 36

# A measure of accuracy is calculated by summing the true
positives and true negatives and dividing them by total
number of samples
knn_accuracy = sum(diag(conf_matrix))/nrow(ServiceTest)

> knn_accuracy
[1] 1
```

knn implementation in R



So, there are two ways of generating this confusion matrix. One you can generate the confusion matrix manually, the other way is to use this caret package which can generate confusion matrix and along with it lot of other parameters what Prof. Raghu has talked about in his performance matrix lecture. Let us see how to generate this confusion matrix manually. So, this predicted knn is the labels that is being predicted using the knn algorithm. And when you observe this command here, this is the last column of the service test data frame which says the true labels of whether the service is needed or not.

When I do the table, it generates contingency table and it stores the result in this confusion matrix. When I print this confusion matrix, the result what I see is as follows. This is the predicted no and yes and these are the true no and yes. Recall that we have seen in your test data service is not needed for 99 cars and service is needed for 36 cars. This knn has exactly predicted all of them correctly, this is what is confusion matrix.

(Refer Slide Time: 26:28)

```
# confusionMatrix command shown below used from caret package
ConF_Matrix <- confusionMatrix(data = predictedknn, ServiceTest$Service)

> ConF_Matrix

Confusion Matrix and Statistics

      Reference
Prediction No Yes
No      99    0
Yes     0    36

      Accuracy : 1
      95% CI : (0.973, 1)
No Information Rate : 0.7333
P-Value [Acc > NIR] : < 2.2e-16
```

What we have seen this is the way you generate the confusion matrix manually. Once you have this confusion matrix, you can calculate the accuracy.

So, how do you calculate the accuracy the formula of accuracy is given in Prof. Raghu's performance matrix lecture. Essentially, I am taking the diagonal elements that is the correctly predicted values divided by the total number of entries in the service test. When you divide that you will get the accuracy as $99 + 36$ is 135, and the n row of service is also 135. This command here diag of confusion matrix take this element 99 and 36 and the some command will sum them up. And when you divide that with the number of rows in the service test that is 135 by 135, you will get the value of knn accuracy as 1.

Since, knn is managed to predict all the no cases has correctly has no and all the yes cases correctly as yes, your accuracy is 1. This is how you generate the confusion matrix manually. Now, let us see how to generate this confusion matrix using the caret package and the command confusion matrix.

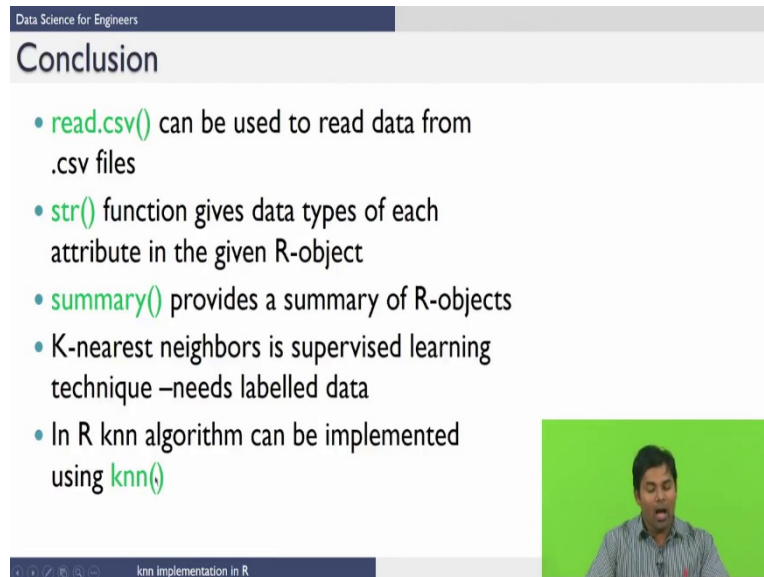
So, the command to generate confusion matrix, which is there in this caret package, is confusion matrix. And the input arguments that you need to give are the predicted labels and the true labels. When you pass these two arguments, this is the confusion matrix that is generated along with confusion matrix it will generate whole lot of other parameters. We have already calculated accuracy manually. We have seen that that is 1. You can also compare now the confusion matrix functions also giving this accuracy as 1.

(Refer Slide Time: 27:12)

Along with this confusion matrix, we will also get a lot of parameters such as sensitivity, specificity, etcetera. So, the reason why you have sensitivity = 1. And specificity = 1 in this case is because all the positive classes are correctly classified all the negative classes are also correctly classified that is the reason why you have the ideal values of one and one for sensitivity and specificity.

So, the balance accuracy is again sensitivity + specificity by 2 which is 2 by 2, it is 1. So, this is how one can implement this knn algorithm in R.

(Refer Slide Time: 27:46)



Data Science for Engineers

Conclusion

- `read.csv()` can be used to read data from .csv files
- `str()` function gives data types of each attribute in the given R-object
- `summary()` provides a summary of R-objects
- K-nearest neighbors is supervised learning technique –needs labelled data
- In R knn algorithm can be implemented using `knn()`

knn implementation in R

In summary what we have seen in this lecture is how to read the dot csv files, how to use the structure and summary functions to know the data types and the summary of R objects and how to implement this K-nearest neighbours algorithm, which is a supervised learning algorithm which needs labelled data. And we have also seen how to implement this K-nearest neighbours algorithm in R using this knn function.

So, with this we end this tutorial session on how to implement knn algorithm in R. In the next lecture, Prof. Raghu will talk about this k means clustering algorithm after which I will come back with a case study on how to implement k means clustering.

Thank you.