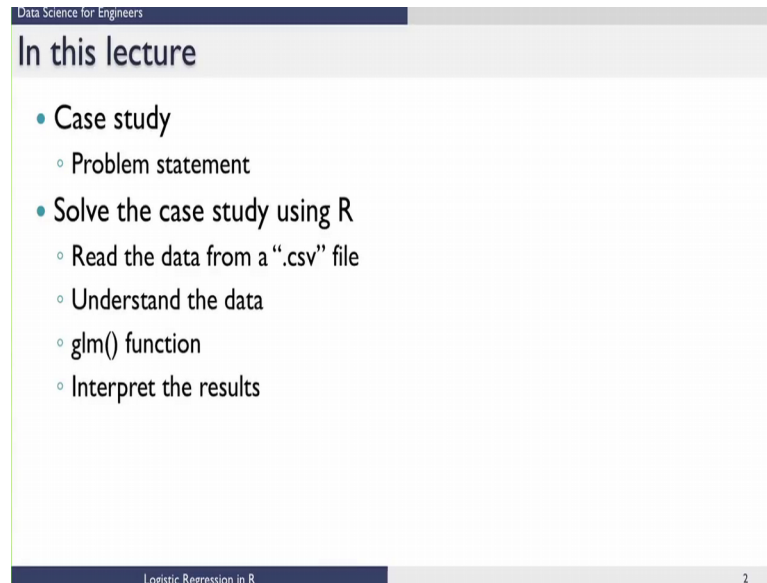


Data science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 45
Logistic Regression implementation in R

(Refer Slide Time: 00:20)

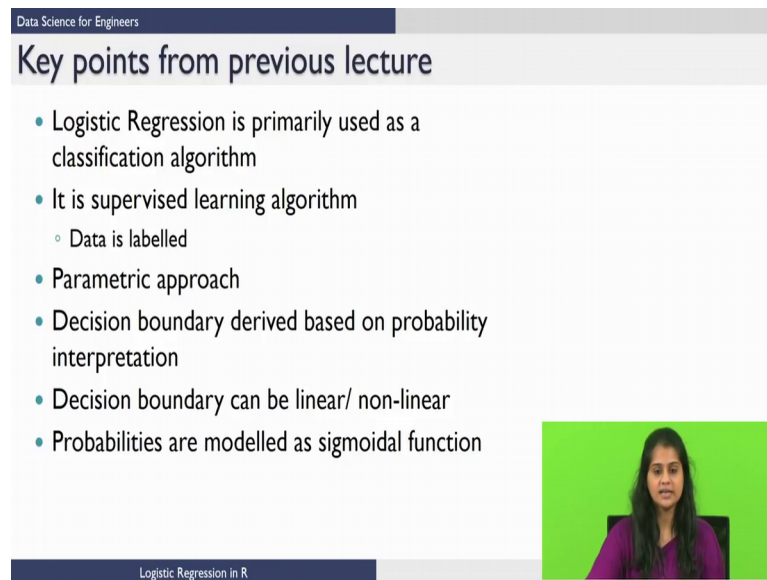


The slide is titled "In this lecture" and is part of a presentation for "Data Science for Engineers". It contains a bulleted list of topics to be covered in the lecture. The slide has a dark blue header bar with the text "Data Science for Engineers" and a dark blue footer bar with the text "Logistic Regression in R" and the number "2".

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

Welcome to the lecture of Implementation of Logistic Regression in R. In this lecture we are going to look at a case study and a problem statement associated with it. We are also going to solve the case study using R and as a part of this we are going to look at, how to read a data from a csv file, how to understand the data and how to interpret the results.

(Refer Slide Time: 00:38)



Data Science for Engineers

Key points from previous lecture

- Logistic Regression is primarily used as a classification algorithm
- It is supervised learning algorithm
 - Data is labelled
- Parametric approach
- Decision boundary derived based on probability interpretation
- Decision boundary can be linear/ non-linear
- Probabilities are modelled as sigmoidal function

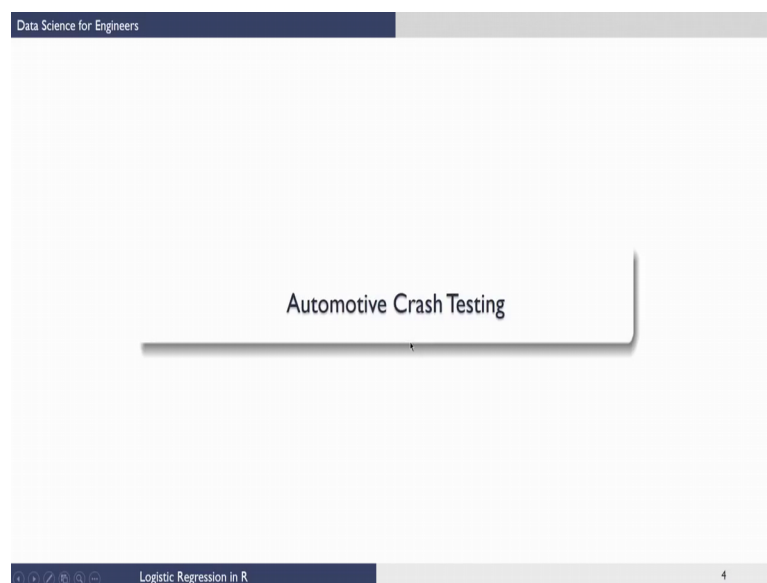
Logistic Regression in R

A video inset in the bottom right corner shows a woman with long dark hair wearing a purple top, speaking against a green background.

Some key points from previous lecture of Professor Raghu's. We know that logistic regression is a classification technique and it is a supervised learning algorithm. By supervised I mean that the data is labelled. It is also a parametric approach since at the end of the application of the algorithm we are going to get parameters out of it.

The decision boundary is derived based on the probability interpretation and it can be linear or non-linear. The probabilities are also modeled as a sigmoidal function.

(Refer Slide Time: 01:14)



Data Science for Engineers

Automotive Crash Testing

A large, empty rectangular box with a thin border and a drop shadow is centered on the slide.

Logistic Regression in R

A video inset in the bottom right corner shows a woman with long dark hair wearing a purple top, speaking against a green background.

So, let us look at the problem statement. We are going to use automotive crash testing case to illustrate this concept.

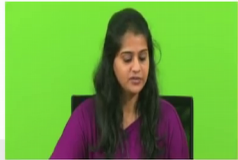
(Refer Slide Time: 01:22)

Data Science for Engineers

Automotive Crash Testing- Problem Statement

- A crash test is a form of destructive testing that is performed in order to ensure high safety standards for various cars

Logistic Regression in R

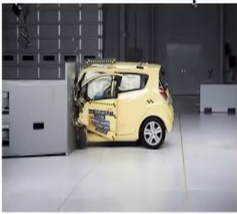


So, a crash test is a form of destructive testing that is performed in order to ensure high safety standard for various cars.

(Refer Slide Time: 01:30)

Data Science for Engineers

Automotive Crash Testing



Hatchback

SUV

Logistic Regression in R



Now, this is our crash test is performed.

(Refer Slide Time: 01:34)

Data Science for Engineers

Automotive Crash Testing- Problem Statement

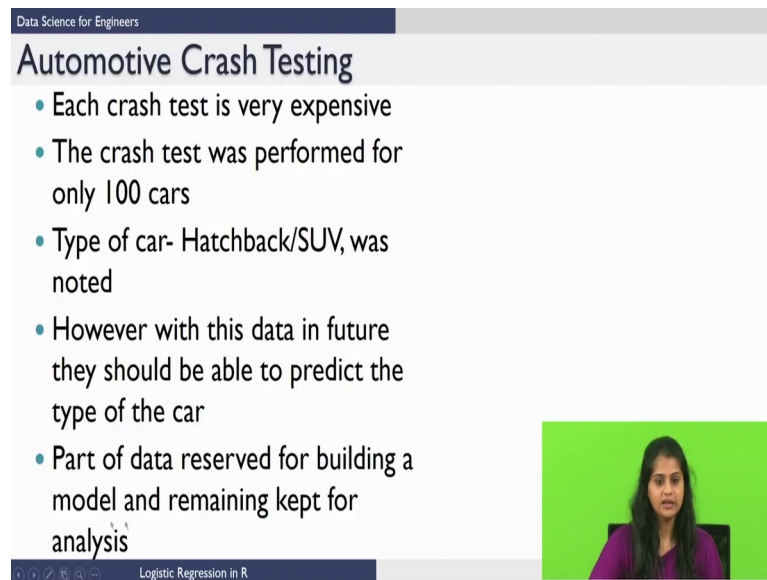
- Several cars have rolled into an independent audit unit for crash test
- They are being evaluated on a defined scale {poor (-10) to excellent(10)} on:
 - 1) Manikin head impact
 - 2) Manikin body impact
 - 3) Interior impact
 - 4) HVAC impact
 - 5) Safety alarm system

Logistic Regression in R

So, several cars have rolled into an independent audit unit for crash test and they have been evaluated on a define scale from poor to excellent with poor being minus 10 and excellent being plus 10. So, from minus 10 to plus 10 is the scale and they are being evaluated on a few parameters. So, let us look at the parameters they have been evaluated on.

So, I have the manikin head impact which is at what impact the head of the car crashes. The manikin body impact the impact on the body of the car the interior impact, the heat ventilation air conditioning impact and the safety alarm system.

(Refer Slide Time: 02:20)



The slide is titled "Automotive Crash Testing" and is part of a presentation on "Data Science for Engineers". It contains a list of five bullet points:

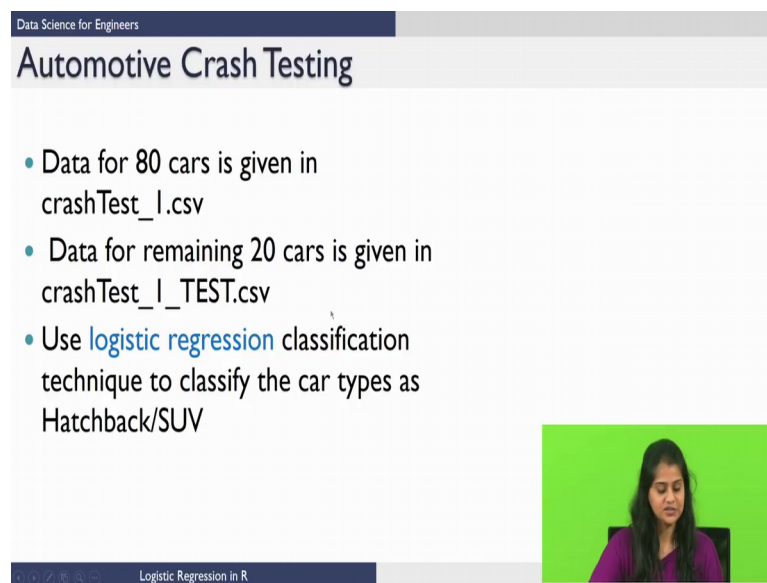
- Each crash test is very expensive
- The crash test was performed for only 100 cars
- Type of car- Hatchback/SUV, was noted
- However with this data in future they should be able to predict the type of the car
- Part of data reserved for building a model and remaining kept for analysis

In the bottom right corner, there is a small video inset showing a woman with long dark hair, wearing a purple top, speaking. The slide also has a footer that reads "Logistic Regression in R".

Now, each crash test is very expensive to perform and hence the company does a crash test for only 100 cars. At the end of the crash test the type of the car is noted. So, that type here is either hatchback or SUV. However, since the crash test is very expensive to perform every time.

So, the company is going to take this data build a model and with this model it should be able to predict the type of the car in future. So, for this we are going to reserve a part of the data for building a model and for training and the rest of the data will be kept for analysis.

(Refer Slide Time: 03:04)



Data Science for Engineers

Automotive Crash Testing

- Data for 80 cars is given in crashTest_1.csv
- Data for remaining 20 cars is given in crashTest_1_TEST.csv
- Use **logistic regression** classification technique to classify the car types as Hatchback/SUV

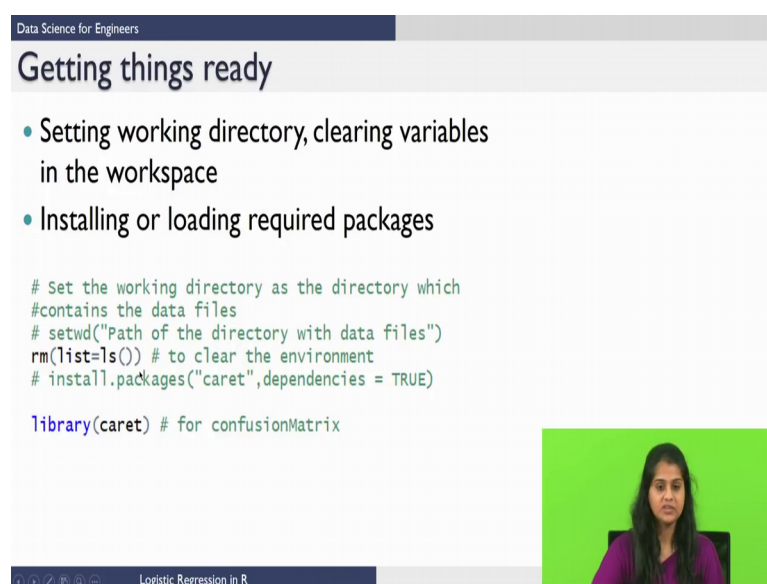
Logistic Regression in R

A video inset in the bottom right corner shows a woman with long dark hair, wearing a purple top, speaking against a green background.

So, for this we have 100 cars in total out of which 80 cars is going to be taken as train and the remaining 20 cars is going to be taken as test. So, the 80 cars is given in crash test underscore 1 dot csv file and the remaining 20 cars is given in crash test underscore 1 underscore TEST dot csv.

Now, we need to use logistic regression technique to classify the car type as hatchback or SUV.

(Refer Slide Time: 03:37)



Data Science for Engineers

Getting things ready

- Setting working directory, clearing variables in the workspace
- Installing or loading required packages

```
# 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)

library(caret) # for confusionMatrix
```

Logistic Regression in R

A video inset in the bottom right corner shows a woman with long dark hair, wearing a purple top, speaking against a green background.

Now, let us look into the solution approach. So, before we jump into modelling let us get the things ready. We need to set the working directory clear the variables in the workspace we also need to load the required packages. So, in this case glm is an inbuilt function.

So, we do not need any specific package to loaded where asked to use the confusion matrix I need a package called carrot which I am going to load before I begin my modeling. I am also going to clear all the variables in the environment using this function which we have already learnt.

(Refer Slide Time: 04:18)

Data Science for Engineers

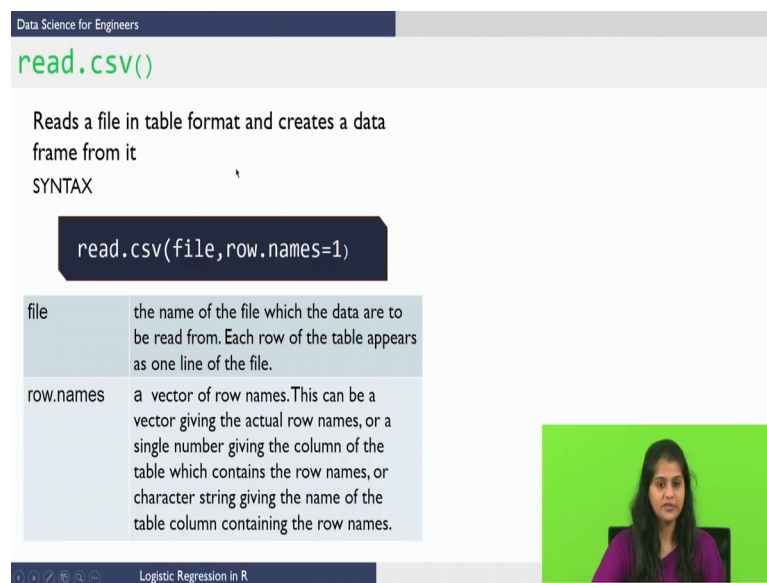
Reading the data

- Data for this case study is provided to you in files with names
 - crashTest_1- training data
 - crashTest_1_TEST- testing data
- To read the data from a “.csv” file we use `read.csv()` function

Logistic Regression in R

So, now let us read the data. So, the data for this case study like I said is provided to you with these to find name. So, crash test underscore 1 is the train data and crash test underscore 1 underscore TEST is the test data.

(Refer Slide Time: 04:38)



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

SYNTAX

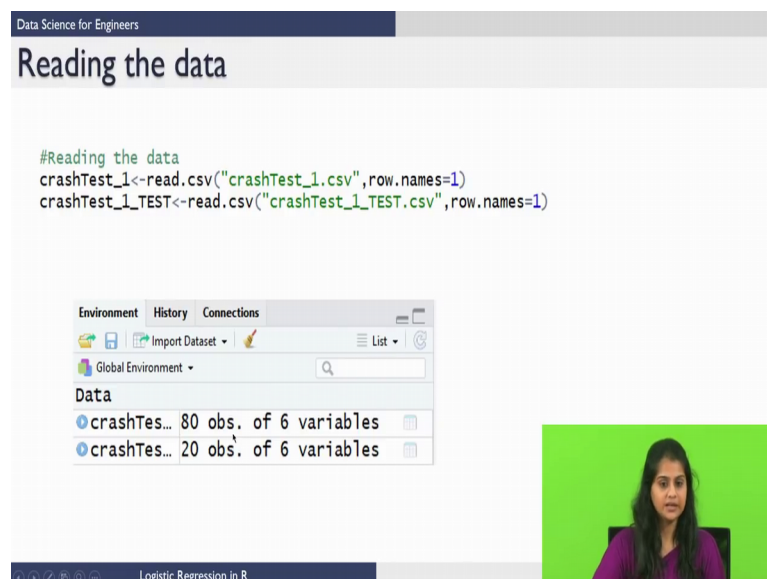
```
read.csv(file, row.names=1)
```

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.

Logistic Regression in R

Now, to read the data from a csv file we are going to use the csv function. Like I said from my earlier lecture it reads a file in table format and creates a data frame format. So, its syntax is given below the inputs are file and column name.

(Refer Slide Time: 04:51)



```
#Reading the data
crashTest_1<-read.csv("crashTest_1.csv", row.names=1)
crashTest_1_TEST<-read.csv("crashTest_1_TEST.csv", row.names=1)
```

Environment History Connections

Import Dataset

Global Environment

Data

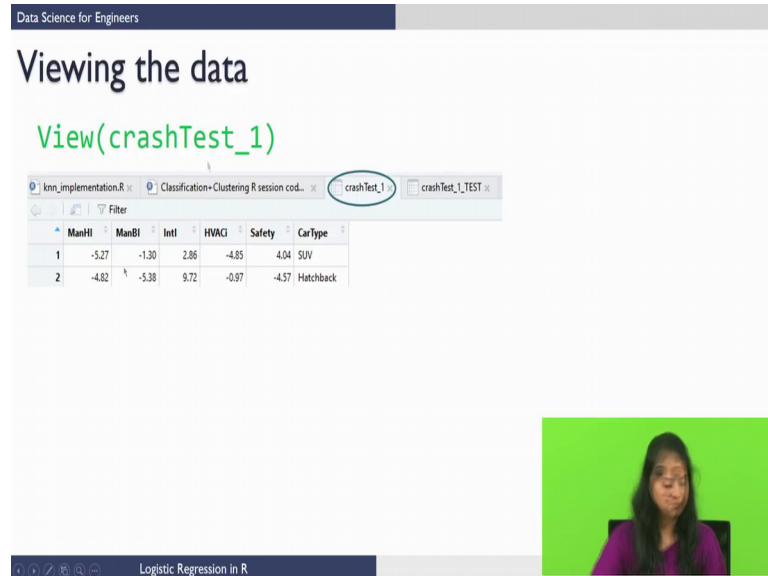
- crashTes... 80 obs. of 6 variables
- crashTes... 20 obs. of 6 variables

Logistic Regression in R

So, now let us read the data. So, I am going to use a function read dot csv followed by the name of my file. Now once this command is run its going to save it in an object called crash test underscore 1 which is a data frame. Similarly I do it for the other data

set as well and now I have another object crash test underscore one underscore test. Now both these data frames will be reflected in the environment.

(Refer Slide Time: 05:21)

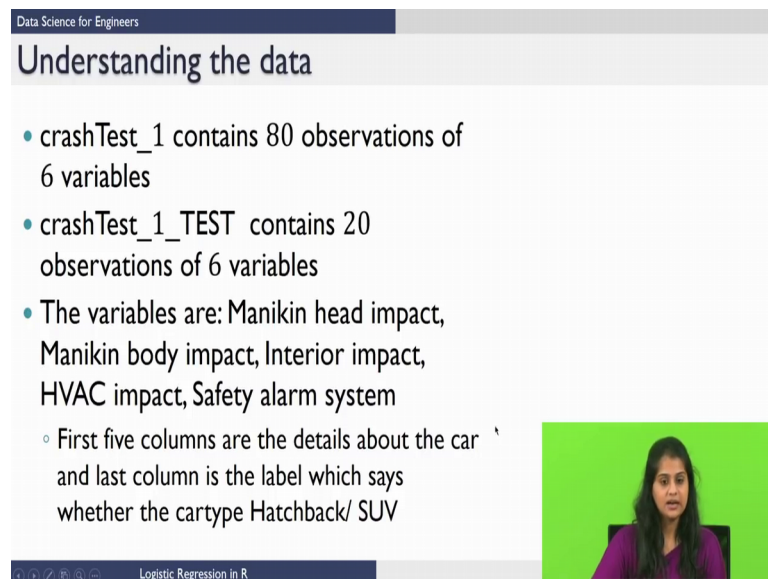


The screenshot shows the RStudio interface with the title 'Data Science for Engineers'. The main window displays the command `View(crashTest_1)` in green text. Below the command, a preview of the `crashTest_1` data frame is shown. The preview includes a table with 2 rows and 6 columns: `ManHl`, `ManBl`, `Intl`, `HVAC`, `Safety`, and `CarType`. The first row shows values: 1, -5.27, -1.30, 2.86, -4.85, 4.04, SUV. The second row shows values: 2, -4.82, -5.38, 9.72, -0.97, -4.57, Hatchback. A small video inset in the bottom right corner shows a woman with long dark hair wearing a purple top.

	ManHl	ManBl	Intl	HVAC	Safety	CarType
1	-5.27	-1.30	2.86	-4.85	4.04	SUV
2	-4.82	-5.38	9.72	-0.97	-4.57	Hatchback

Now, let us view the data. So, I am going to use the view command followed by the name of my data frame. So, this is how it appears. Once you run the command a separate tab will appear with all the variables and the values.

(Refer Slide Time: 05:40)



The screenshot shows the RStudio interface with the title 'Data Science for Engineers'. The main window displays a slide titled 'Understanding the data'. The slide contains the following bullet points:

- `crashTest_1` contains 80 observations of 6 variables
- `crashTest_1_TEST` contains 20 observations of 6 variables
- The variables are: Manikin head impact, Manikin body impact, Interior impact, HVAC impact, Safety alarm system
 - First five columns are the details about the car and last column is the label which says whether the cartype Hatchback/ SUV

A small video inset in the bottom right corner shows a woman with long dark hair wearing a purple top.

Now, let us try to understand the data the data set crash test underscore one contains 80 observations of 6 variables and similarly crash test underscore 1 underscore TEST

contains 20 observations of 6 variables. Now, like I said earlier we have 5 variables here which have been measured at the end of a crash test, and if you can see from the data the first five columns are the details about the car and the last column is the label which says whether the car type is hatchback or SUV.

(Refer Slide Time: 06:15)

Data Science for Engineers

Structure of the data

- Structure of data
 - Variables and their data types
- `str()`

SYNTAX

```
str(object)
```

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

Logistic Regression in R

So, let us look at the structure of the data by structure I mean the variables and their corresponding data types. So, structure is the command which is represented as `str` I need to give an object to it as input the object here is the desired object for which we want to look at the structure.

(Refer Slide Time: 06:36)

Data Science for Engineers

Structure of train data

```
> str(crashTest_1)
'data.frame': 80 obs. of 6 variables:
 $ ManHI : num -5.27 -4.82 9.57 2.84 0 0.4 5.94 5.78 0.86 7.36 ...
 $ ManBI : num -1.3 -5.38 -7.5 -2.85 2.68 6.34 3.14 -1.75 -4.32 7.42 ...
 $ IntI : num 2.86 9.72 -7.61 0.92 -4.15 0.83 -6.65 -6.85 8.1 0.27 ...
 $ HVACi : num -4.85 -0.97 1.33 5.51 0.85 5.03 6.62 0.73 -8.96 -8.62 ...
 $ Safety : num 4.04 -4.57 -5.1 -6.64 5.58 -8.1 -1.32 5.5 3.1 3.08 ...
 $ CarType: Factor w/ 2 levels "Hatchback", "SUV": 2 1 1 1 2 2 1 1 1 2 ...
```



Logistic Regression in R

So, if you look at the structure of the same data it tells you that crashTest underscore 1 is of the type data frame with 80 observations and 6 variables and all the five variables are numeric and the class variable which is card type is a factor with levels hatchback and SUV.

(Refer Slide Time: 06:59)

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.
--------	---



Logistic Regression in R

Similarly you can also look at the structure of the test set so now, let us look at the summary of the data and let us see what it has to tell about the data. So, summary is the

five point summary if the input is a data frame and if the input is an object than the corresponding summary for the object is returned. So, this is the syntax.

(Refer Slide Time: 07:20)

Data Science for Engineers

Summary of crashTest_1

```
> summary(crashTest_1)
      ManHI      ManBI
Min.   :-9.9300 Min.   :-9.9400
1st Qu.: -5.1950 1st Qu.: -5.7050
Median :  0.6350 Median : -1.8150
Mean    :-0.0935 Mean    :-0.9277
3rd Qu.:  5.0500 3rd Qu.:  3.4175
Max.     :  9.5700 Max.     :  9.6100

      IntI      HVACi
Min.   :-9.9900 Min.   :-9.8200
1st Qu.: -5.5725 1st Qu.: -5.6750
Median : -0.4150 Median :  0.8700
Mean    :-0.1349 Mean     :  0.1197
3rd Qu.:  4.9775 3rd Qu.:  5.0625
Max.     :  9.7200 Max.     :  9.8900

      Safety      CarType
Min.   :-9.8000 Hatchback:50
1st Qu.: -4.6775 SUV      :30
Median :  0.8300
Mean    :  0.5437
3rd Qu.:  4.6225
Max.     :  9.9900
```



Logistic Regression in R

So, these summary for the train data which is crashTest underscore 1 is given below in the snippet. So, for the numerical variables it is a five point summary with minimum first quartile and median, mean, third quartile and maximum. For the categorical variable which is the factor car type here it returns the frequency count.

(Refer Slide Time: 07:43)

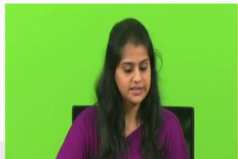
Data Science for Engineers

Summary of crashTest_1_TEST

```
> summary(crashTest_1_TEST)
      ManHI      ManBI
Min.   :-9.940  Min.   :-8.740
1st Qu.: -5.535 1st Qu.: -2.502
Median :  0.740 Median :  0.670
Mean    :  0.047 Mean    :  0.328
3rd Qu.:  5.110 3rd Qu.:  2.500
Max.     :  9.090 Max.     :  8.420

      IntI      HVACi
Min.   :-8.950  Min.   :-9.2300
1st Qu.: -3.272 1st Qu.: -2.4550
Median :  1.200 Median :  0.6750
Mean    :  0.524 Mean    :  0.7235
3rd Qu.:  3.908 3rd Qu.:  5.3375
Max.     :  8.870 Max.     :  8.3300

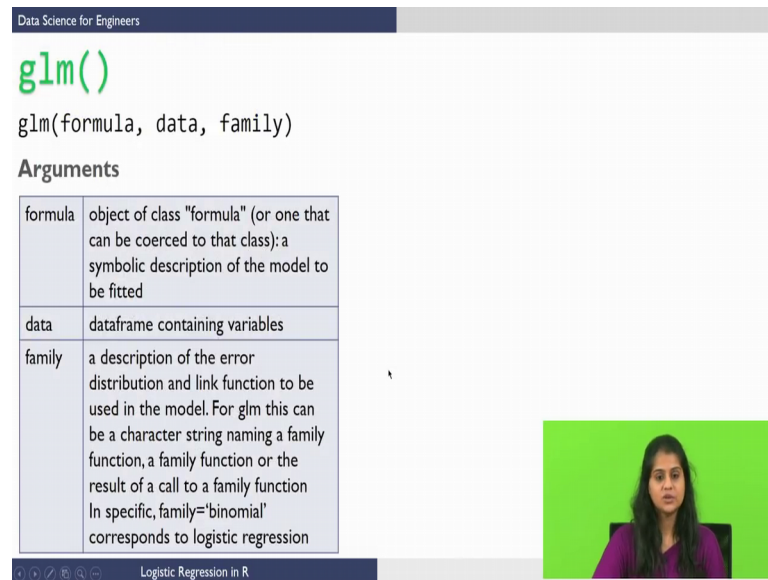
      Safety      CarType
Min.   :-8.660  Hatchback:10
1st Qu.: -6.095 SUV      :10
Median : -0.770
Mean    :  0.191
3rd Qu.:  4.992
Max.     :  9.620
```



Logistic Regression in R

So, for the test it again returns a five point summary for the numerical variables and for the card type it tells me that there are 10 cars of type hatchback and 10 cars of the type SUV.

(Refer Slide Time: 07:58)



Data Science for Engineers

glm()

glm(formula, data, family)

Arguments

formula	object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted
data	dataframe containing variables
family	a description of the error distribution and link function to be used in the model. For glm this can be a character string naming a family function, a family function or the result of a call to a family function. In specific, family='binomial' corresponds to logistic regression

Logistic Regression in R

So, now let us look at the function glm which we are going to use for logistic regression. So, glm stands for generalized linear model and the input to it is a formula a data and family.

So, formula is basically a symbolic representation of the model you want to fit. So, in our case it become the card type. So, it is basically a class. Data is a data frame from which you want to obtain your variables and family is binomial if you use logistic regression. There are also the family which I listed inside the function that if you write family equal to binomial then it specifically corresponds to logistic regression.

(Refer Slide Time: 08:45)

Data Science for Engineers

Building a logistic regression model

```
# Model  
logisfit<-glm(formula = crashTest_1$CarType~., family = 'binomial',  
              data = crashTest_1)
```

$$p(X) = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

```
> logisfit  
  
Call: glm(formula = crashTest_1$CarType ~ ., family = "binomial", data = crashTest_1)  
  
Coefficients:  
(Intercept)      ManH1      ManB1      Int1      HVAC1      Safety  
      -22.76       -13.48        36.02       -44.90       -58.50       -27.36  
  
Degrees of Freedom: 79 Total (i.e. Null); 74 Residual  
Null Deviance:      105.9  
Residual Deviance: 5.359e-08    AIC: 12
```

Logistic Regression in R

23

Now, let us build a logistic regression model. Now, I am going to use the glm function the formula here says that get the variable card type which is our class here from the data crashTest underscore 1 and to access it I use a dollar symbol. Now, crashTest underscore 1 is my train data. Now, like I said earlier family equal to binomial correspond to logistic regression and now the variable car type is to be obtained from crashTest underscore 1.

Now, once I run this command an object of the type glm is created and I call it logisfit. So, if you could recall from Professor Raghu's lecture E-model the probabilities as a sigmoidal function and on the right hand side I have the log odds ratio and this is equal to the hyperplane equation. This is also the decision boundary. Here p of x by 1 minus p of x is the odds, where p of x is the probability of success in 1 minus p of x is the probability of failure. Now, p of x in our case is the probability that the car type is hatchback and 1 minus p of x is the probability that the car type is SUV.

Now, let us look at the model now if you run the model logisfit in your console. This is what is displayed in the first line it displays the formula, in the next line it displays the coefficient then I have degrees of freedom. So, it displays two degrees of freedom. So, the first degrees of freedom is when you have a null model that is only with the intercept and in the second case you have a degrees of freedom equal to 74 which means that I have included all the variables into my modeling.

(Refer Slide Time: 10:42)

Data Science for Engineers

Summary of model

```
> summary(logisfit)
```

Call:
glm(formula = crashTest_1\$CarType ~ ., family = "binomial", data = crashTest_1)

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-1.316e-04	-2.100e-08	-2.100e-08	2.100e-08	1.266e-04

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-22.76	12007.54	-0.002	0.998
ManHI	-13.48	3077.29	-0.004	0.997
ManBI	36.02	7221.18	0.005	0.996
IntI	-44.90	8853.08	-0.005	0.996
HVACi	-58.50	11461.92	-0.005	0.996
Safety	-27.36	5396.42	-0.005	0.996

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1.0585e+02 on 79 degrees of freedom
Residual deviance: 5.3590e-08 on 74 degrees of freedom
AIC: 12

Number of Fisher Scoring iterations: 25

$\hat{\beta}_i$
 $i = 0, 1, \dots, 5$

Logistic Regression in R

24

So, now let us look at each of the coefficients in detail. So, I am going to again use summary of logisfit. So, now, this is similar to what we have done in linear regression the first action tells you the formula that we have used, second section tells you the measure of a fit and the 5 point summary for it.

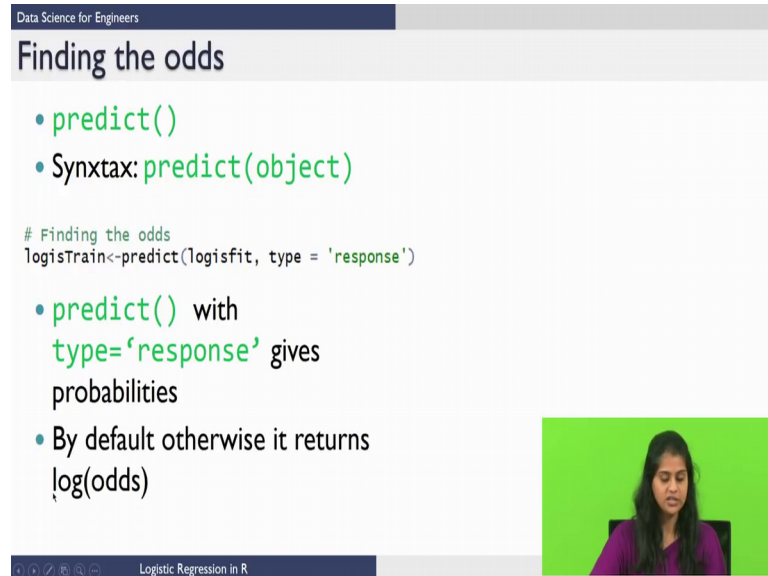
I have the next section as the coefficients. So, these are the beta i's hat, where i is equal to 0 to 5. Now, for the intercept it is beta not hat so on and so forth, I have 4 columns here I have the estimate standard error the z value and the associated probability. Now in logistic regression the coefficients gives you the change in log odds of the output for a unit increase in the predictor value which is the input value. So, now if you can see the probability is really really high and none of the variables are statistically significant.

If you go to next section I have something called null deviance and residual deviance. So, null deviance is the deviance of your model when only the intercept is taken into account and residual deviance is a deviance of your model when all the terms are taken into account. So, you can look at the degrees of freedom and tell whether it is a null model that is a reduced model or the full model.

So, for the reduced model I take only the intercept. So, my degrees of freedom reduced by 1, so 80 minus 1, 79 whereas, for the full model I take all the variables into account. So, I have 80 minus 6 degrees of freedom which is 74. So, I have something called the Fisher scoring iteration. So, the Fisher's scoring is used for maximum likelihood

estimation and it is a derivative of Newton Raphson method. So, it tells you that the number of iterations that it has taken is 25.

(Refer Slide Time: 12:46)



Data Science for Engineers

Finding the odds

- `predict()`
- Syntax: `predict(object)`

```
# Finding the odds
logisTrain<-predict(logisfit, type = 'response')
```

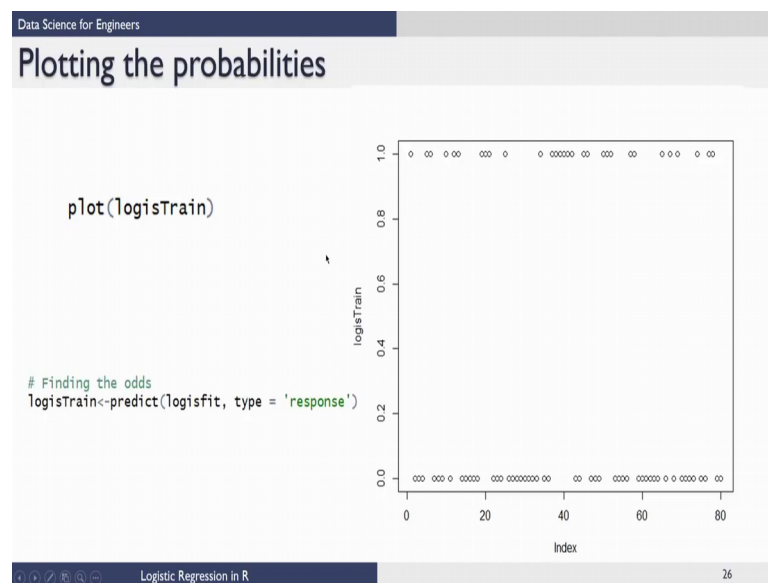
- `predict()` with `type='response'` gives probabilities
- By default otherwise it returns `log(odds)`

Logistic Regression in R

Now, let us find the odds. To find the odds we are going to use the predict function and the syntax is credit and my input is an object for which I want to predict it. Now, for our data I am going to use predict.

My input here is the logistic regression model, now if I do not give any data then the function assumes that I want to predict it on the train set which is crashTest underscore 1 in our case. Now, type equal to response gives you the output as probabilities, but if you do not mention this it by default gives you the log odds.

(Refer Slide Time: 13:33)



Now, let us plot the probabilities, so after you run the command I have saved it as logisTrain and I am going to use the plot function to plot the probabilities. On to my right I have the plot on the y axis I have the probabilities and on the x axis I have the index. So, from this plot it is clear that the classes are well separated, but we still do not know which site belongs to which car type. So, let us see how to find out which side belongs to which car type.

(Refer Slide Time: 14:09)

Data Science for Engineers

Identifying probabilities associated with the CarType

- Mean of probabilities
- This helps us identify the probabilities associated with the two classes

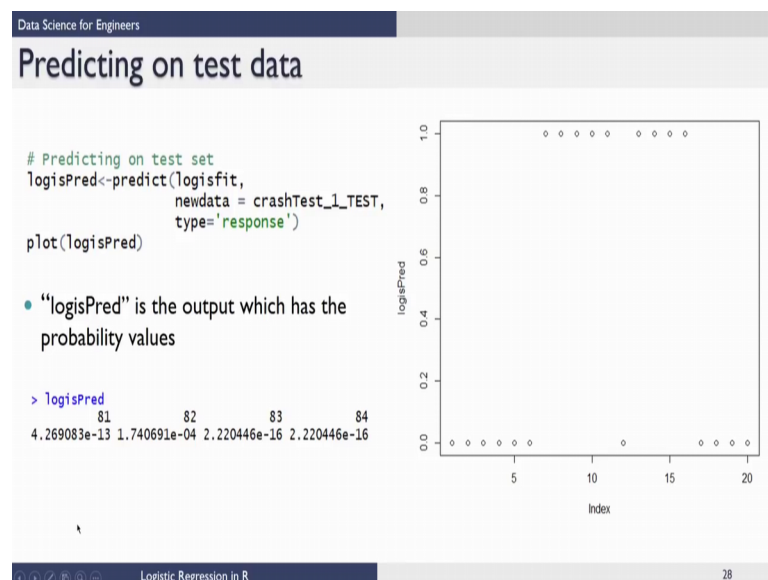
```
> tapply(logisTrain, crashTest_1$CarType, mean)  
Hatchback      SUV  
2.851316e-10  1.000000e+00
```

Logistic Regression in R

So, to do that I am going to find the mean of the probabilities and this will help us to identify the probabilities which are associated with the two classes. I am going to use the `t` apply function now they should ring a bell we will on this in the introduction to basics of R programming lecture. Then put here I give as `logistrain`. Now I am want to classified based on the car type. So, I am going to give `crashTest` underscore 1 dollar card type and the function I want to find is the mean. So, I am giving mean as the function.

Now, if you run the command I have the probabilities associated with each car type for hatchback it is really really low its 2.85×10^{-10} where as for SUV it is 1. So, this tells us that the lower probabilities are associated with the car type hatchback where as the higher probabilities are associated with the car type SUV.

(Refer Slide Time: 15:13)

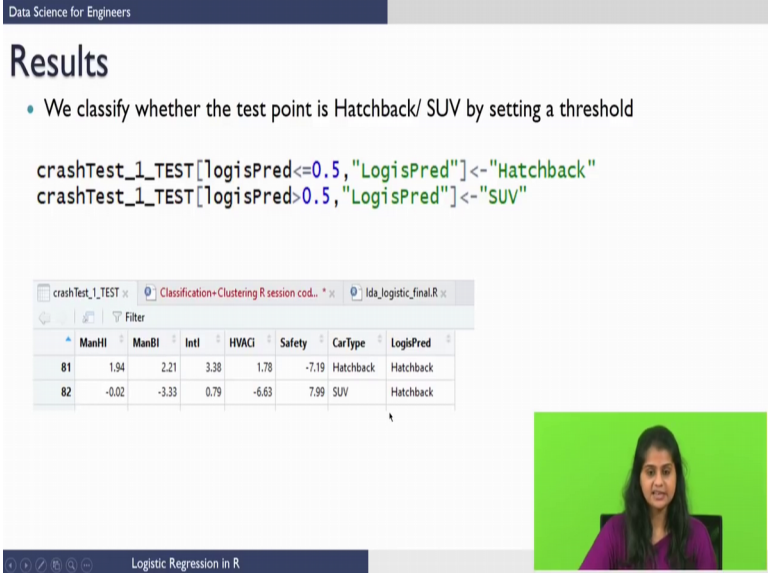


So, now let us predict this on the test data. I am again going to use the `predict` function my input again is the logistic regression model. Now my new data is the test data. So, `crashTest` underscore 1 underscore TEST is the test set. Now, since I want to again model the probabilities I am going to give type equal to `response`, now once this command is executed it gets stored as an object `logisPred` and now I will plot the `logisPred`. So, if I plotted I have the plot on my right, again I have the predicted values of probability on my y axis and I can see that on the x I have the index.

Now, even for the test set the classes are well separated, now I know that the points which fall here belong to the class of hatchback and the points which are above belong to

the class SUV. Now, logispred is the output it has the probability values and I have a small snippet below that shows me the probability values. Now, I have shown you only for the first four points you will also get similar values for the remaining points.

(Refer Slide Time: 16:28)



The screenshot shows a presentation slide titled "Results" from a course "Data Science for Engineers". The slide contains a bullet point: "We classify whether the test point is Hatchback/ SUV by setting a threshold". Below this, two lines of R code are displayed: `crashTest_1_TEST$logisPred<=0.5,"LogisPred"]<-"Hatchback"` and `crashTest_1_TEST$logisPred>0.5,"LogisPred"]<-"SUV"`. A data table is shown with columns: ManHl, ManBl, Intl, HWAG, Safety, CarType, and LogisPred. The table has two rows of data. A small video inset in the bottom right corner shows a woman with dark hair wearing a purple top.

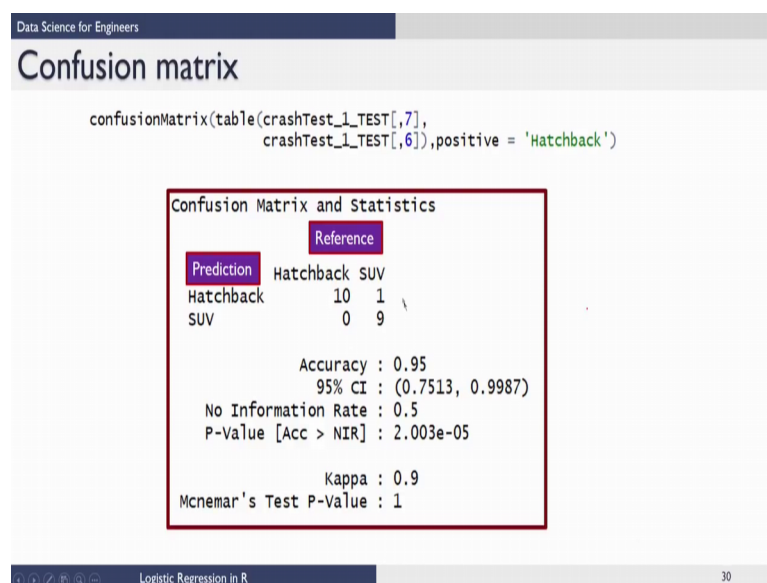
	ManHl	ManBl	Intl	HWAG	Safety	CarType	LogisPred
81	1.94	2.21	3.38	1.78	-7.19	Hatchback	Hatchback
82	-0.02	-3.33	0.79	-6.63	7.99	SUV	Hatchback

Now, let us look at the result now we want to classify whether the test point is hatchback or SUV by setting a threshold value. So, in this case I am going to set a threshold value of 0.5.

So, now, I am going to say that from the data crashTest underscore 1 underscore TEST create a column call logispred and if the value that we have calculated for that point which is logispred if that is less than or equal to 0.5 then assigned hatchback under this column and again from the same data if the logispred is greater than 0.5 assigned SUV under this column.

If you do that and if you run the commands this is how it creates a column. So, if you can see the last column which is the 7th column contains the predicted values and under each of these I have whether it is hatchback or SUV. Now, the reason to do this is to check how accurately our classifier is able to classify and unseen data.

(Refer Slide Time: 17:34)



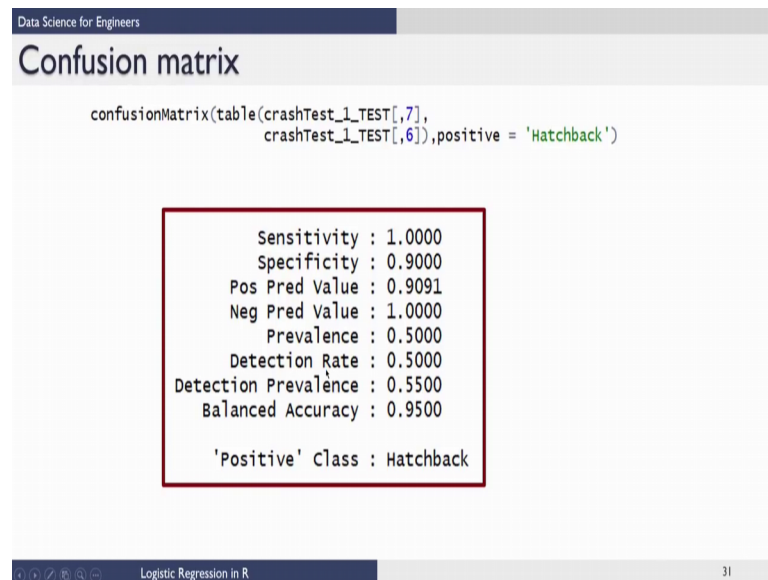
So, now let us look at the confusion matrix. So, the function is confusion matrix with a capital M, now again to use this function you should have already loaded the library (Refer Time: 17:43).

Now, my input to this function is a table. So, I have the table command, now inside the table command I am giving my predicted values which is in the column 7 from the data crashTest underscore 1 underscore TEST, and I am giving the actual values which are the actual labels. There is also another parameter positive by default if you do not give any class as a positive class the command chooses the first class that it encounters as the positive class. So, if you do not want that you can always go back and change it under the parameter positive.

Now, I have the confusion matrix below. So, if you look at it I have the reference labels here and I have the predicted labels here. So, this says that predicted as hatchback truly hatchback there are 10 cases, it has identified all the 10 hatchbacks correctly that predicted as hatchback, but truly SUV is one and predicted as SUV and truly SUV are 9. So, out of the 10 SUV cases it has identified 9 correctly and there has been 1 misclassification.

Now, if you look at the accuracy value it is 0.95 if you can recall from Professor Raghu's performance measure lecture accuracy is nothing but the sum of the true positive and true negative divided by the total number of observations which is 20 in this case.

(Refer Slide Time: 19:21)



So, I again have the command on the top. I have the sensitivity value which is equal to one the positive labels here are hatchback and all of them have been identified correctly, whereas if you can see there has been one misclassification and hence this specificity drops to 0.9. There is also something called balance accuracy which is equal to 0.95. Now, balance accuracy is the average of sensitivity and specificity.

All the other performance measures have been explained by Professor Raghu in his lecture of performance measure and you can always go back and refer to it to know more about the other performance measures. In this lecture we looked at the case study of automotive crash testing we also saw how to read the data, we saw how to understand it we used glm function to model logistic regression and we looked at using confusion matrix to interpret the results.

Thank you.