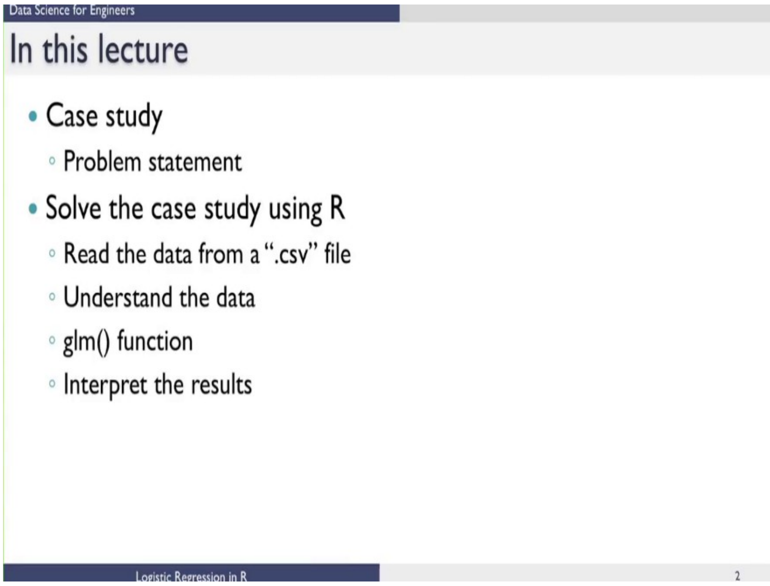
Data science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

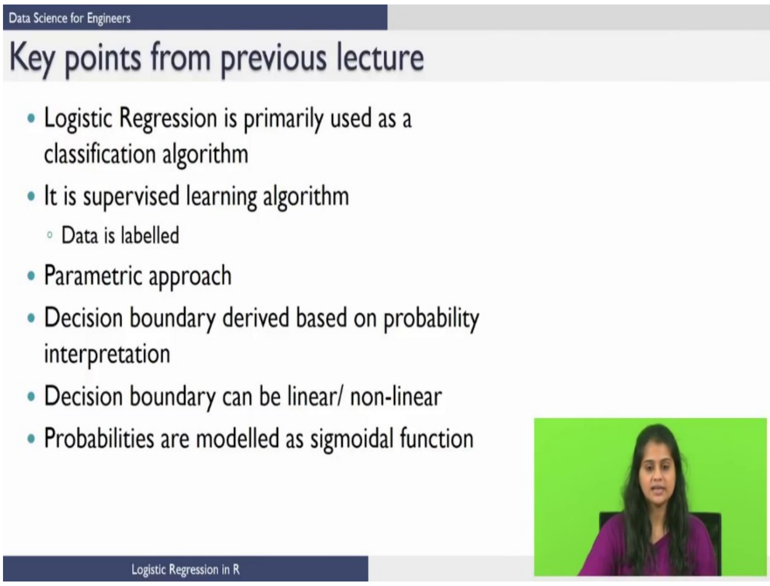
Lecture - 45
Logisitic Regression implementation in R

(Refer Slide Time: 00:20)



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 le, how to understand the data and how to interpret the results.

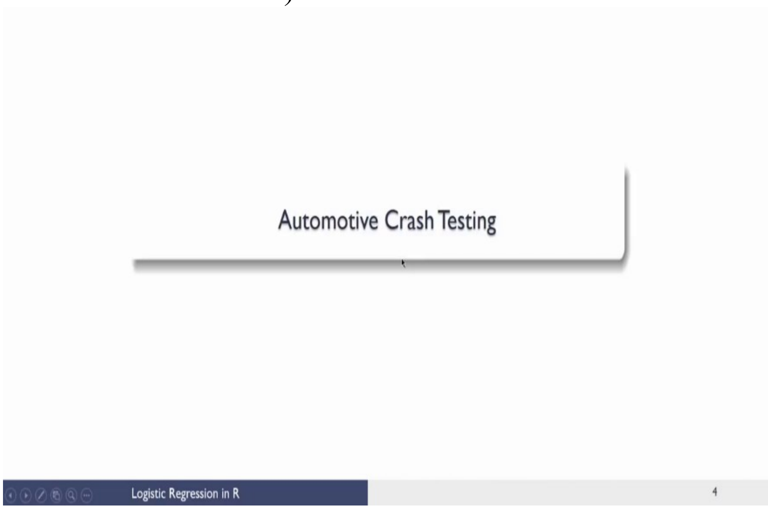
(Refer Slide Time: 00:38)



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.

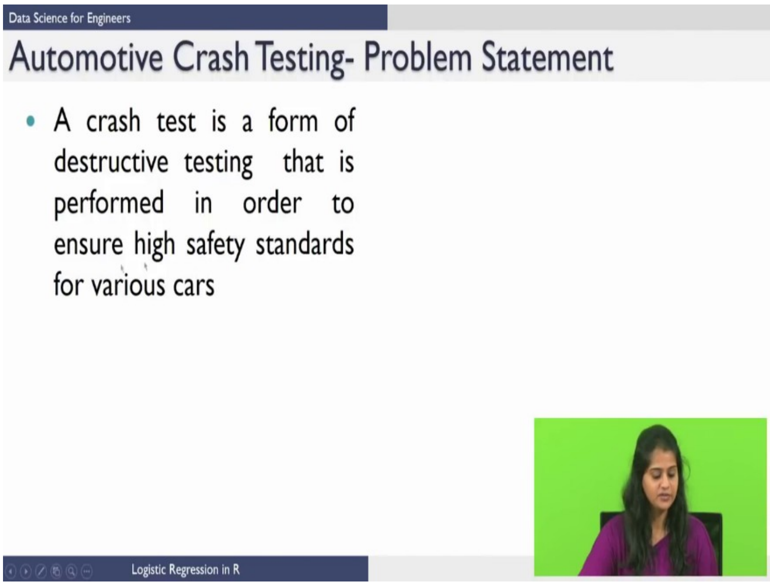
The decision boundary is derived based on the probability interpretation and it can be linear or non-linear. The probabilities are also modeled as a sigmoidal function.

(Refer Slide Time: 01:14)



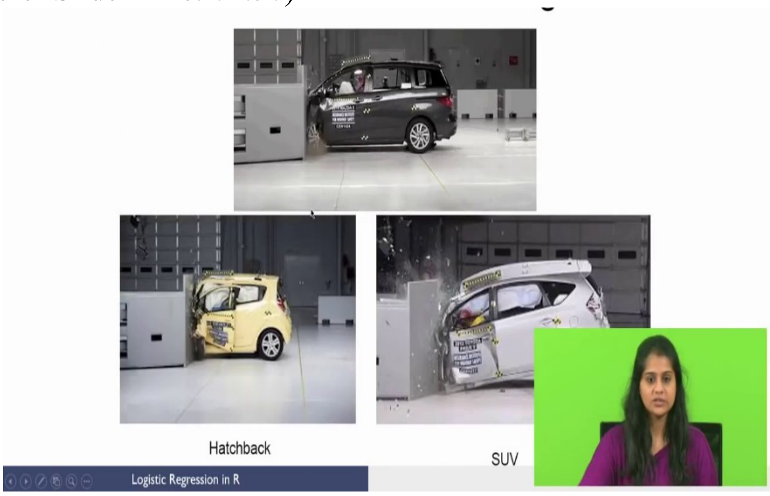
So, let us look at the problem statement. We are going to use automotive crash testing case to illustrate this concept.

(Refer Slide Time: 01:22)



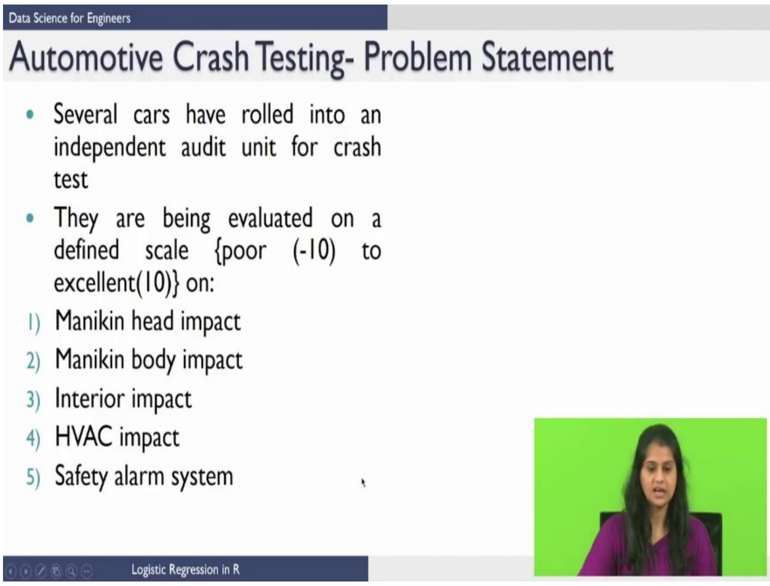
So, a crash test is a form of destructive testing that is performed in order to ensure high safety standard for various cars.

(Refer Slide Time: 01:30)



Now, this is how a crash test is performed.

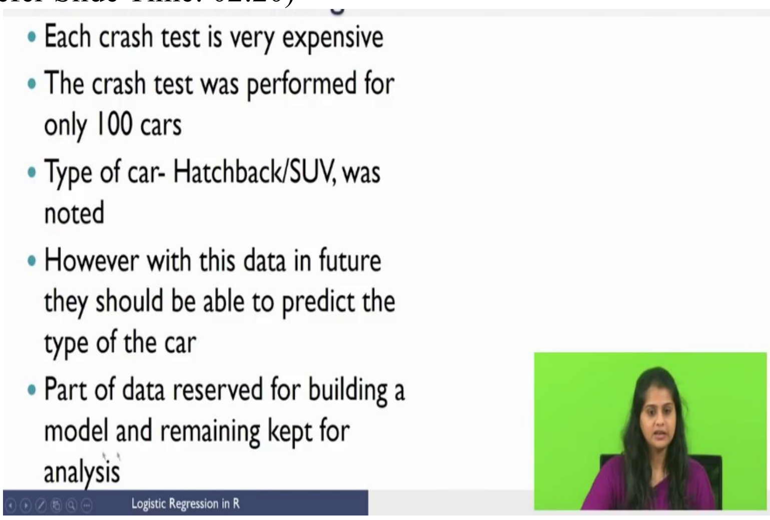
(Refer Slide Time: 01:34)



So, several cars have rolled into an independent audit unit for crash test and they have been evaluated on a defined scale from poor to excellent with poor being - 10 and excellent being + 10. So, from - 10 to + 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.

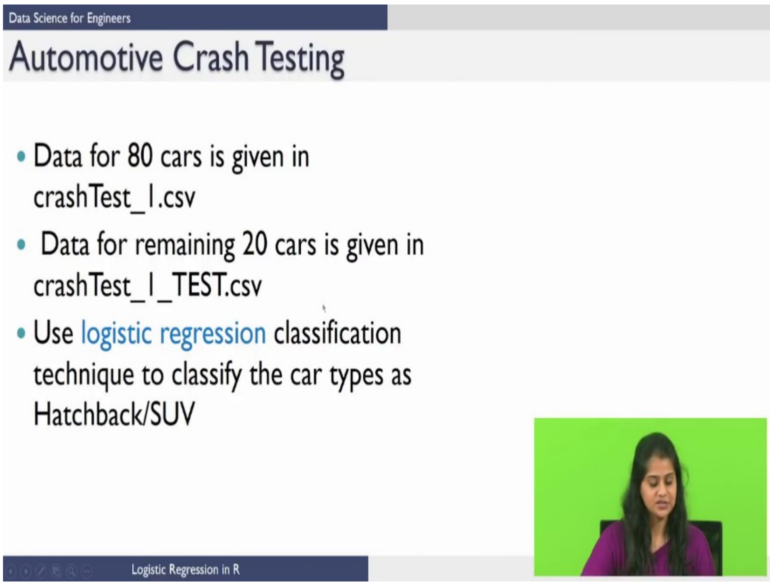
So, I have the manikin head impact which is at what impact the head of the car crashes, the manikin body impact, the impact on the body of the car, the interior impact, the heat ventilation air conditioning impact and the safety alarm system.

(Refer Slide Time: 02:20)



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.

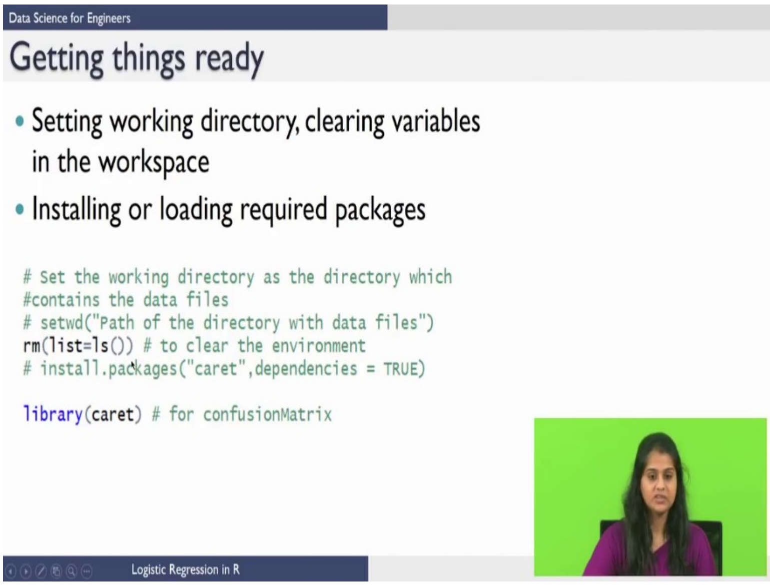
(Refer Slide Time: 03:04)



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.

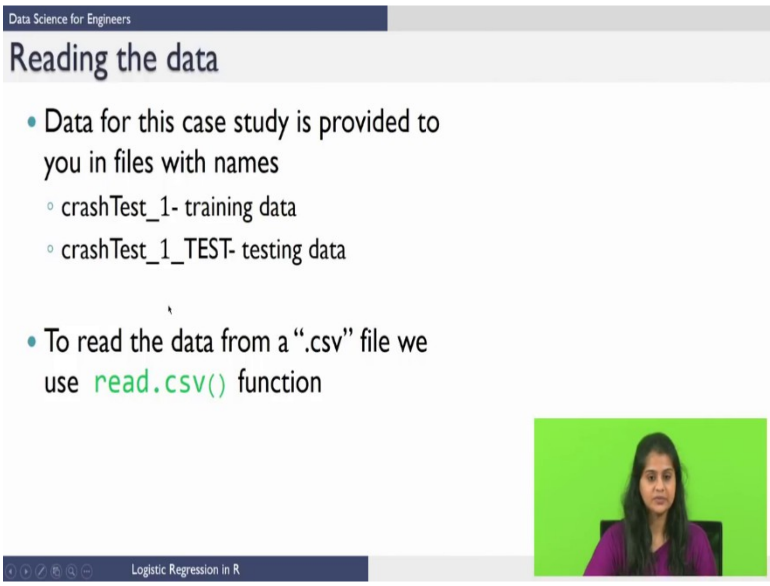
Now, we need to use logistic regression technique to classify the car type as hatchback or SUV.

(Refer Slide Time: 03:37)



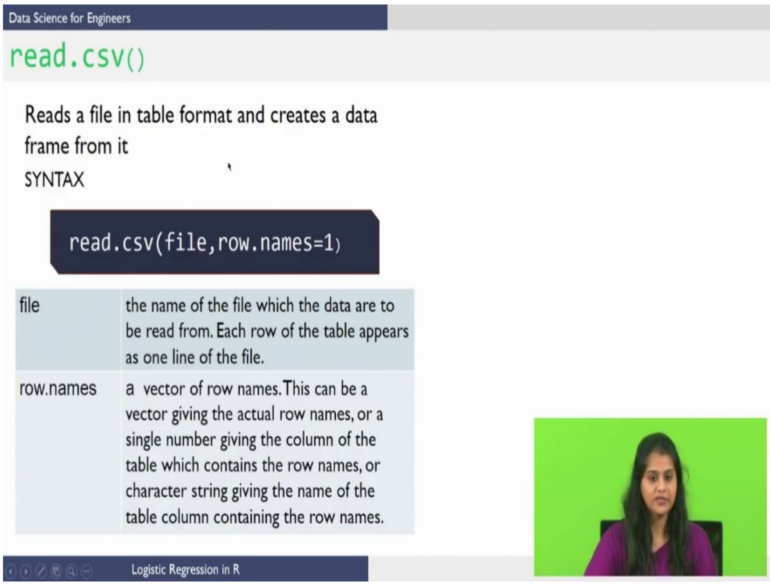
Now, let us look into the solution approach. So, before we jump into modelling, let us get the things ready. We need to set the working directory, clear the variables in the workspace, we also need to load the required packages. So, in this case glm is an inbuilt function so we do not need any specific package to loaded whereas 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)



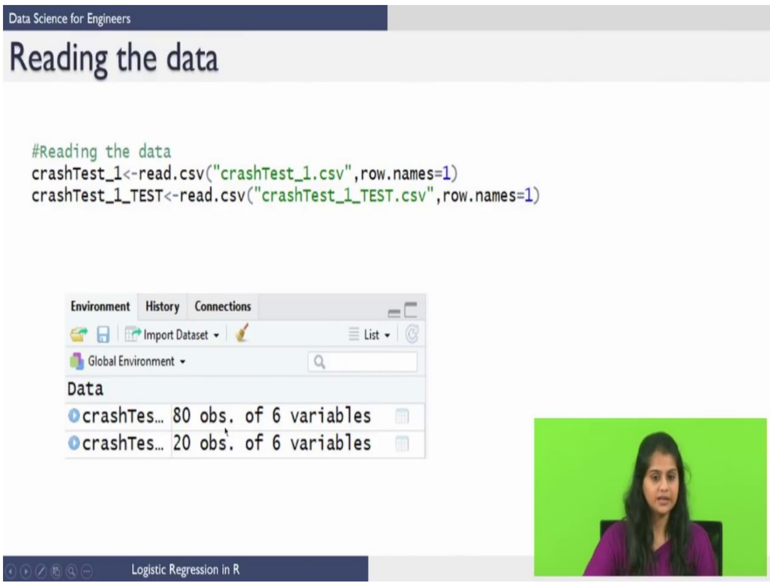
So, now let us read the data. So, the data for this case study like I said is provided to you with these to file name. So, crash test underscore 1 is the train data and crash test underscore 1 underscore TEST is the test data.

(Refer Slide Time: 04:38)



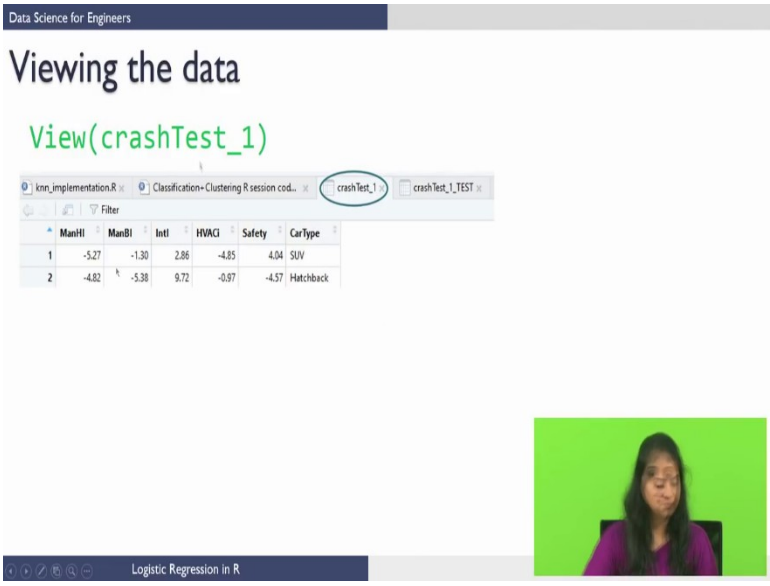
Now, to read the data from a csv le we are going to use the csv function. Like I said from my earlier lecture it reads a file in table format and creates a data frame from it. So, its syntax is given below the inputs are le and column name.

(Refer Slide Time: 04:51)



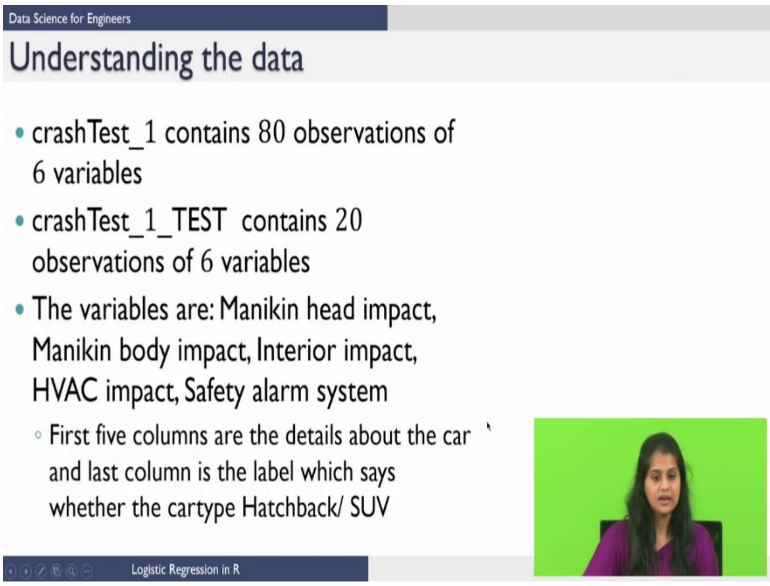
So, now let us read the data. So, I am going to use a function read dot csv followed by the name of my le. Now once this command is run, its going to save it in an object called crash test underscore 1 which is a data frame. Similarly I do it for the other data set as well and now I have another object crash test underscore one underscore test. Now both these data frames will be reflected in the environment.

(Refer Slide Time: 05:21)



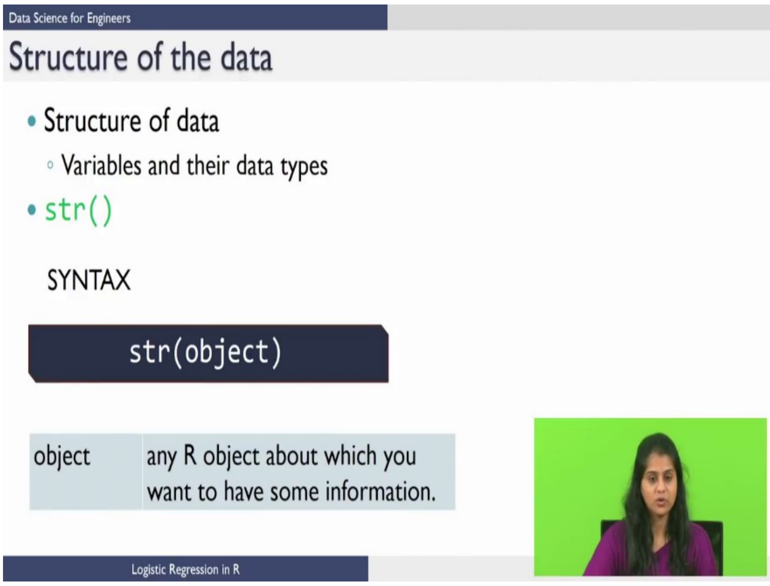
Now, let us view the data. So, I am going to use the view command followed by the name of my data frame. So, this is how it appears. Once you run the command a separate tab will appear with all the variables and the values.

(Refer Slide Time: 05:40)



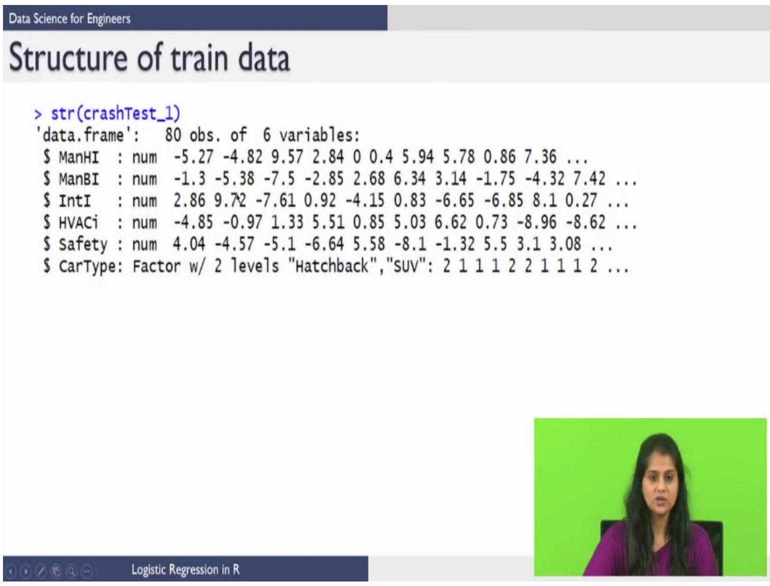
Now, let us try to understand the data. The data set crash test underscore one contains 80 observations of 6 variables and similarly crash test underscore 1 underscore TEST contains 20 observations of 6 variables. Now, like I said earlier we have 5 variables here which have been measured at the end of 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 card type is hatchback or SUV.

(Refer Slide Time: 06:15)



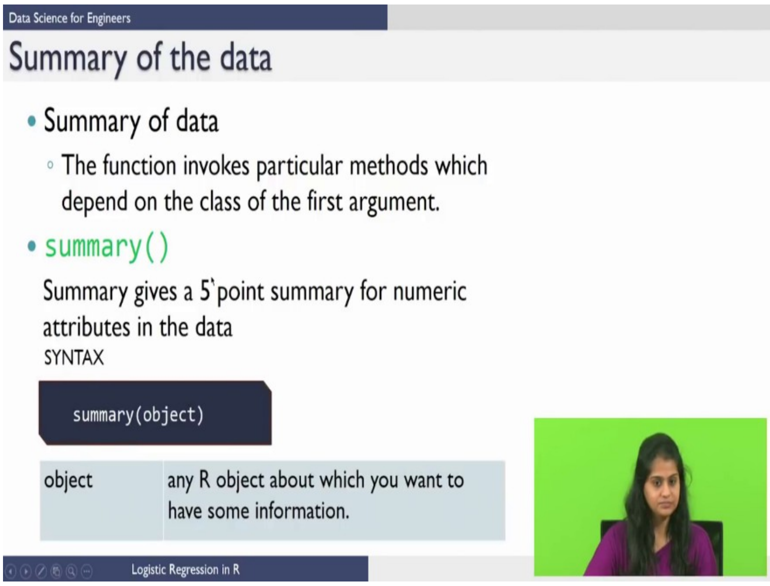
So, let us look at the structure of the data. By structure I mean the variables and their corresponding data types. So, structure is the command which is represented as str. I need to give an object to it as input the object here is the desired object for which we want to look at the structure.

(Refer Slide Time: 06:36)



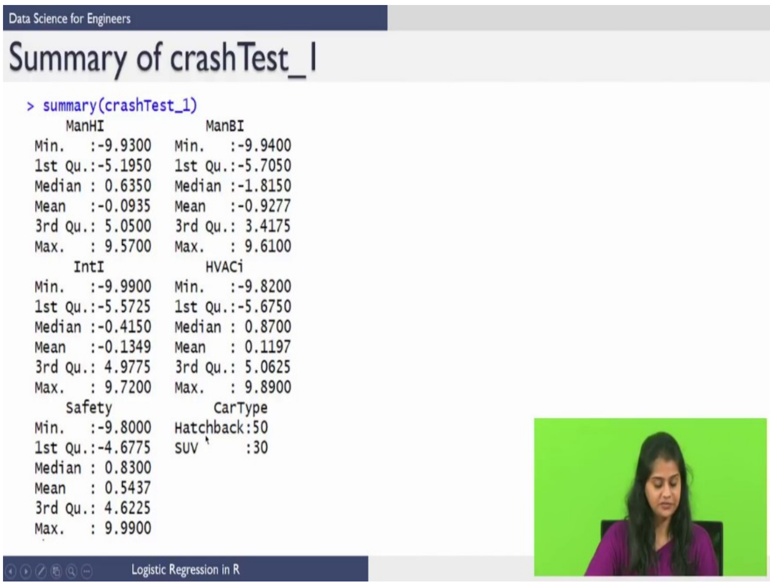
So, if you look at the structure of the train data it tells you that crashTest_1 is of the type data frame with 80 observations and 6 variables and all the five variables are numeric and the class variable which is card type is a factor with levels hatchback and SUV.

(Refer Slide Time: 06:59)



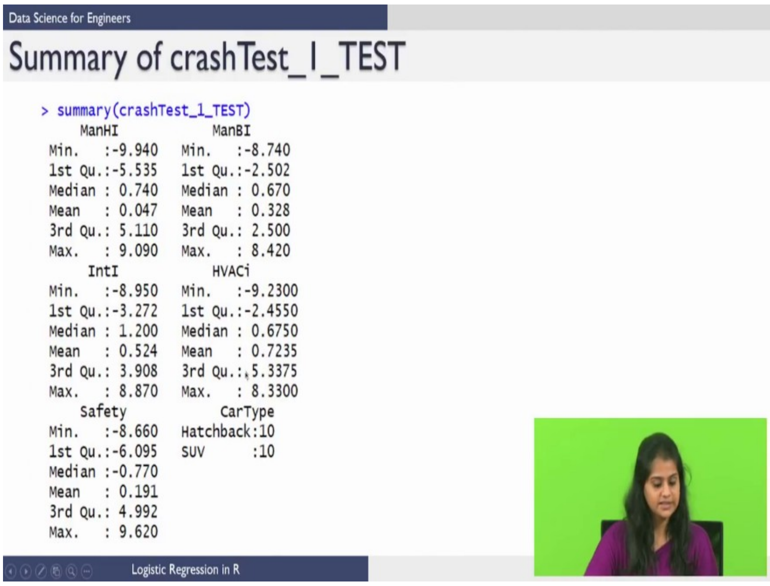
Similarly you can also look at the structure of the test set. So now, let us look at the summary of the data and let us see what it has to tell about the data. So, summary is the five point summary if the input is a data frame and if the input is an object than the corresponding summary for the object is returned. So, this is the syntax.

(Refer Slide Time: 07:20)



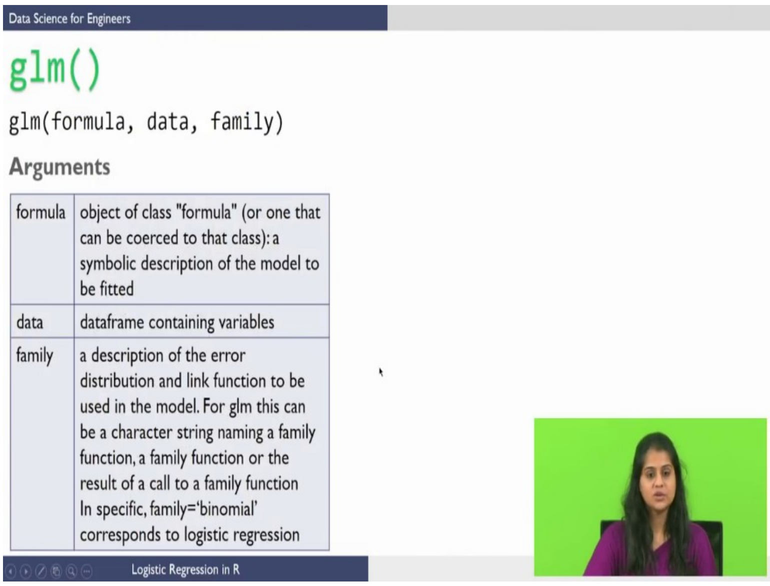
So, the summary for the train data which is crashTest_1 is given below in the snippet. So, for the numerical variables it is a five point summary with minimum first quartile and median, mean, third quartile and maximum. For the categorical variable which is the factor car type here it returns the frequency count.

(Refer Slide Time: 07:43)



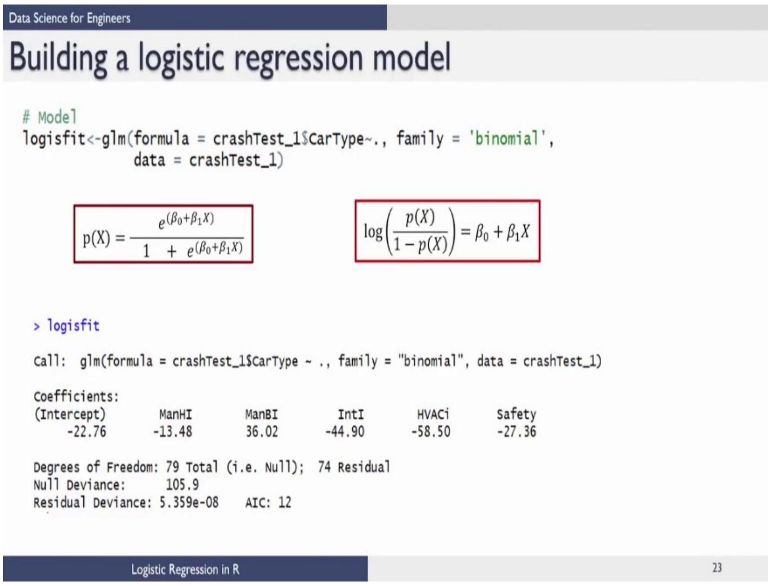
So, for the test it again returns a five point summary for the numerical variables and for the car type it tells me that there are 10 cars of type hatchback and 10 cars of the type SUV.

(Refer Slide Time: 07:58)



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 t. So, in our case it become the car 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 other families which are listed inside the function but if you write family = binomial then it specifically corresponds to logistic regression.

(Refer Slide Time: 08:45)

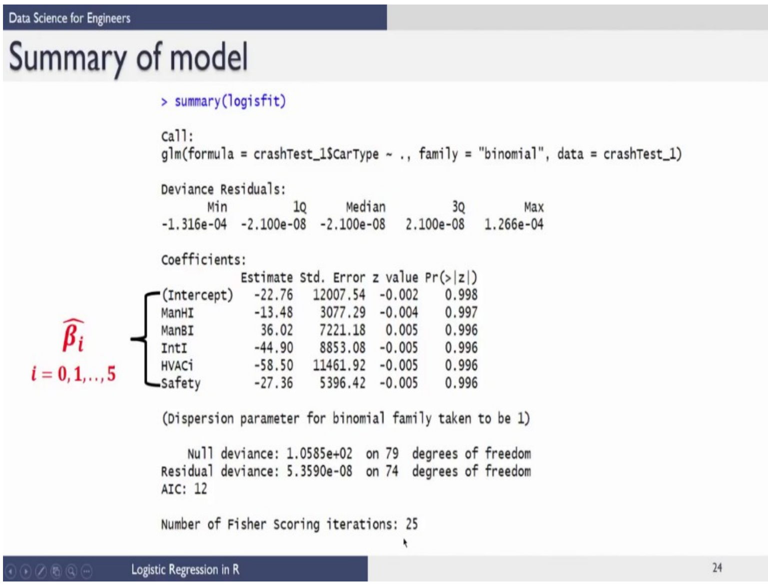


Now, let us build a logistic regression model. Now, I am going to use the glm function the formula here says that get the variable card type which is our class here from the data crashTest_1 and to access it I use a dollar symbol. Now, crashTest_1 is my train data. Now, like I said earlier family = binomial corresponds to logistic regression and now the variable car type is to be obtained from crashTest_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, we model the probabilities as a sigmoidal function and on the right hand side I have the log odds ratio and this = the hyperplane equation. This is also the decision boundary. Here p (x)/(1 - p (x)) is the odds, where p (x) is the probability of success in (1 - p (x)) is the probability of failure. Now, p (x)in our case is the probability that the car type is hatchback and (1 - p (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 = 74 which means that I have included all the variables into my modeling.

(Refer Slide Time: 10:42)

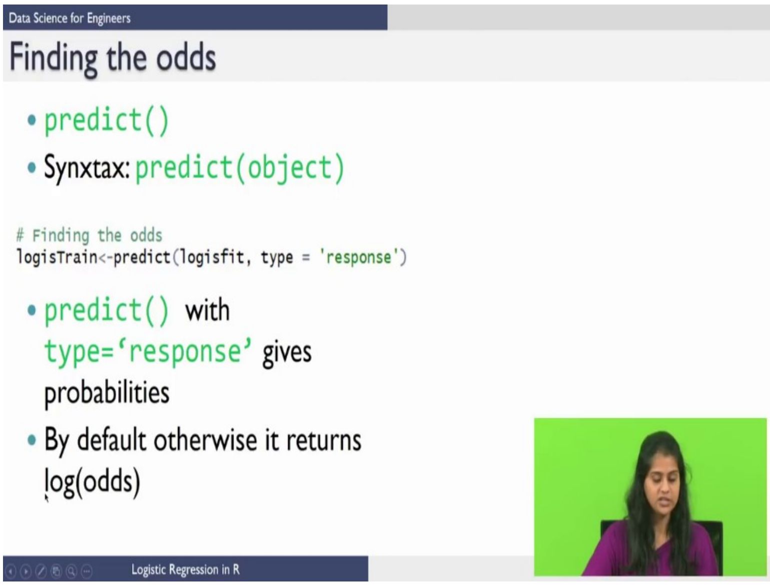


So, now let us look at each of the coefficients in detail. So, I am going to again use summary of logisfit. So now, this is similar to what we have done in linear regression. The first section tells you the formula that we have used, second section tells you the measure of a t and the 5 point summary for it. I have the next section as the coefficients. So, these are the $\hat{\beta}_i$'s, where i = 0 to 5. Now, for the intercept it is $\hat{\beta}_0$ 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 interceptor mistaken 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.

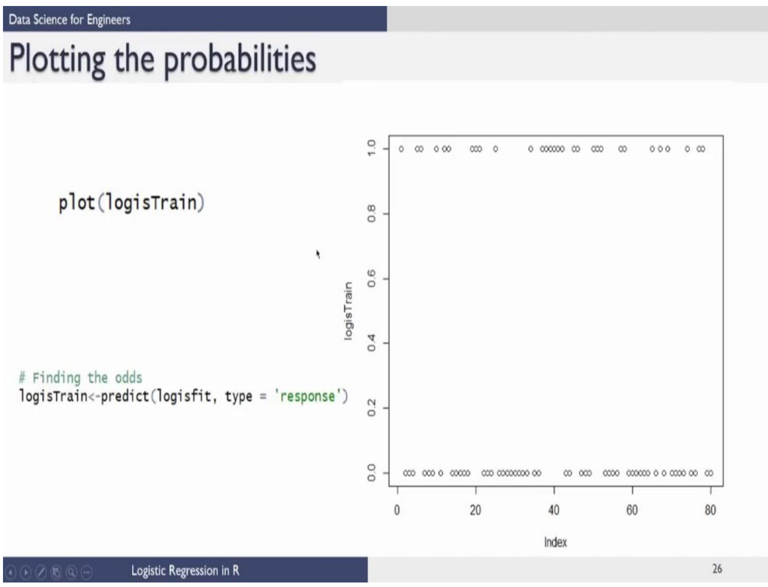
So, for the reduced model I take only the interceptor. So, my degrees of freedom reduced by 1, so 80 - 1, 79. Whereas, for the full model I take all the variables into account. So, I have 80 - 6 degrees of freedom which is 74. So, I have something called the Fisher scoring iteration. So, the Fisher's 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)



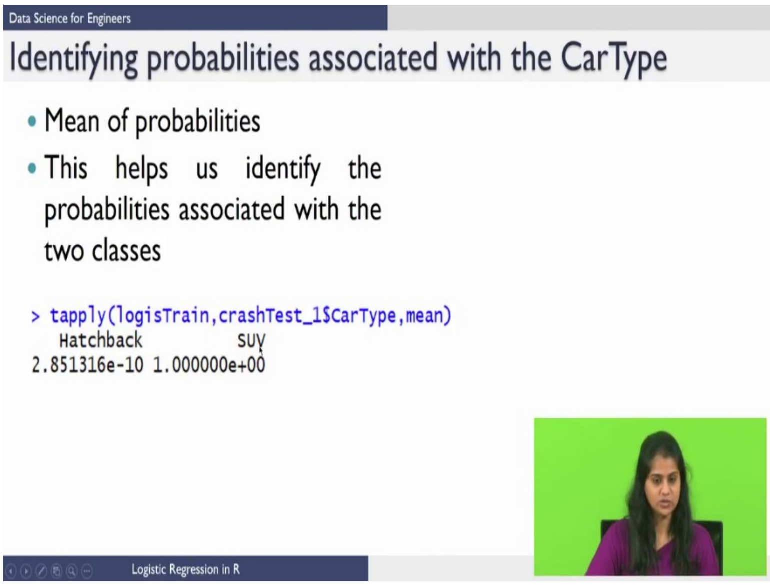
Now, let us find the odds. To find the odds we are going to use the predict function and the syntax is predict and my input is an object for which I want to predict it. Now, for our data I am going to use predict. My input here is the logistic regression model, now if I do not give any data then the function assumes that I want to predict it on the train set which is crashTest_1 in our case. Now, type = 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.

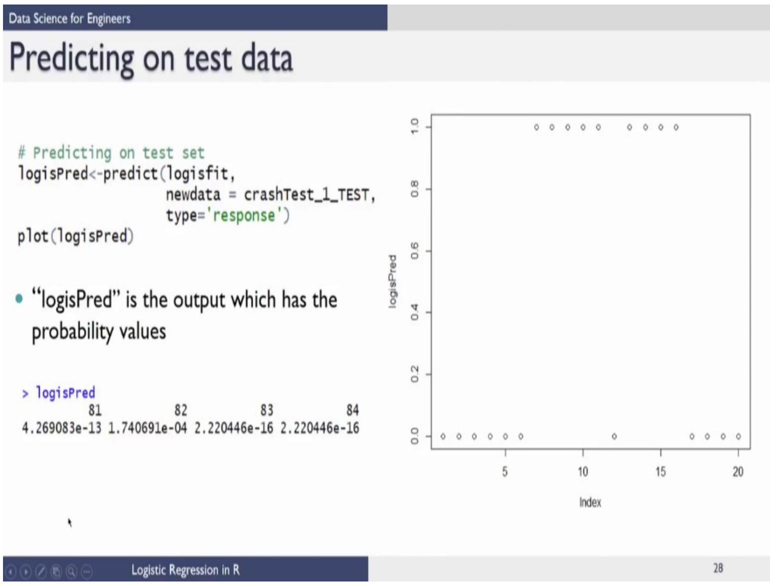
(Refer Slide Time: 14:09)



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 this 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_1$car 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 into 10 power - 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.

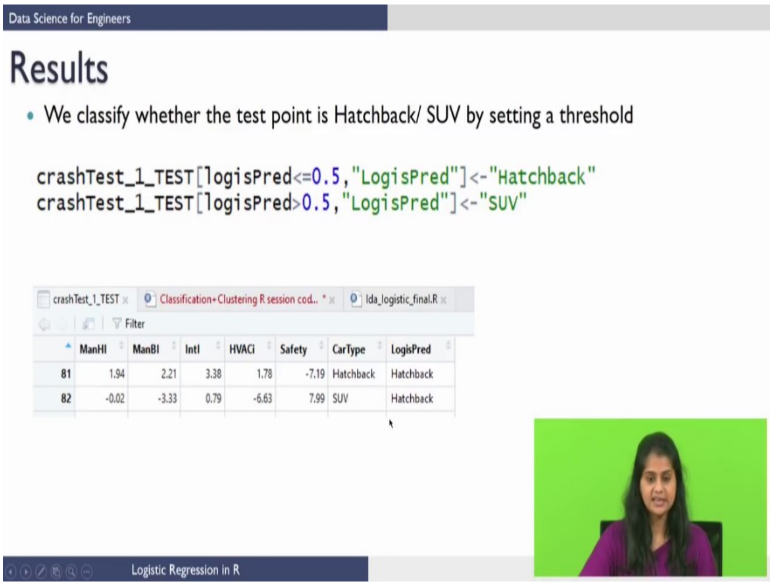
(Refer Slide Time: 15:13)



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_1_TEST is the test set. Now, since I want to again model the probabilities, I am going to give type = response, now once this command is executed it gets stored as an object logispred and now I will plot the logispred. So, if I plot it, 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 prob-ability 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)

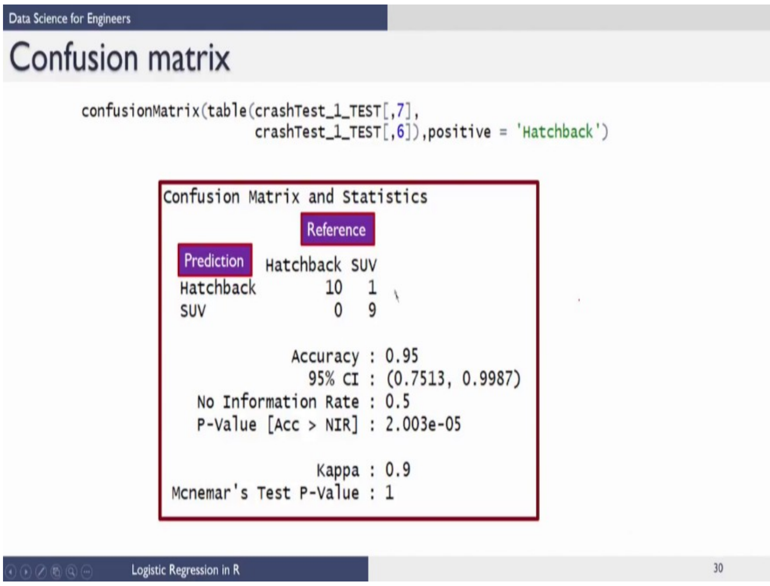


Now, let us look at the result. Now we want to classify whether the test point is hatchback or SUV by setting a threshold value. So, in this case I am going to set a threshold value of 0.5.

So, now, I am going to say that from the data crashTest_1_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 = 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 an unseen data.

(Refer Slide Time: 17:34)



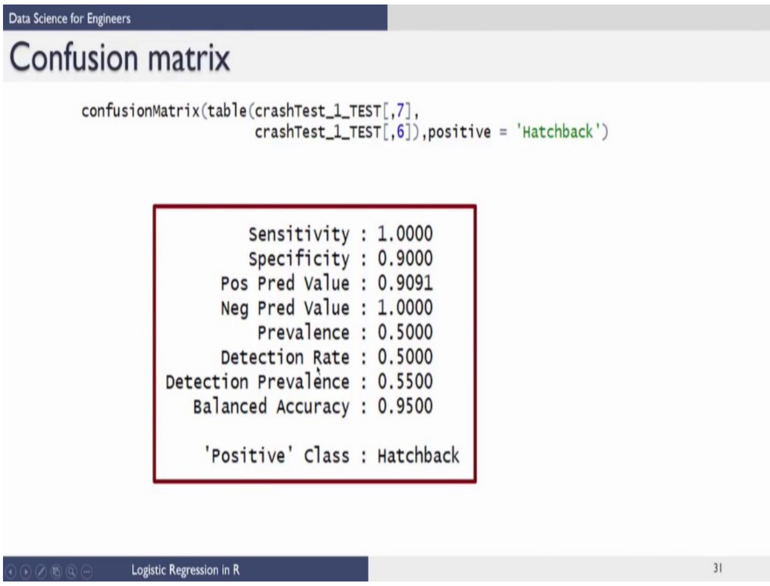
So, now let us look at the confusion matrix. So, the function is confusion matrix with a capital M, now again to use this function you should have already loaded the library caret.

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_1_test, and I am giving the actual values which are the actual labels. There is also another parameter called positive. Now 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. But, 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 mis-classification.

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 balanced accuracy which = 0.95. Now, balanced 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.