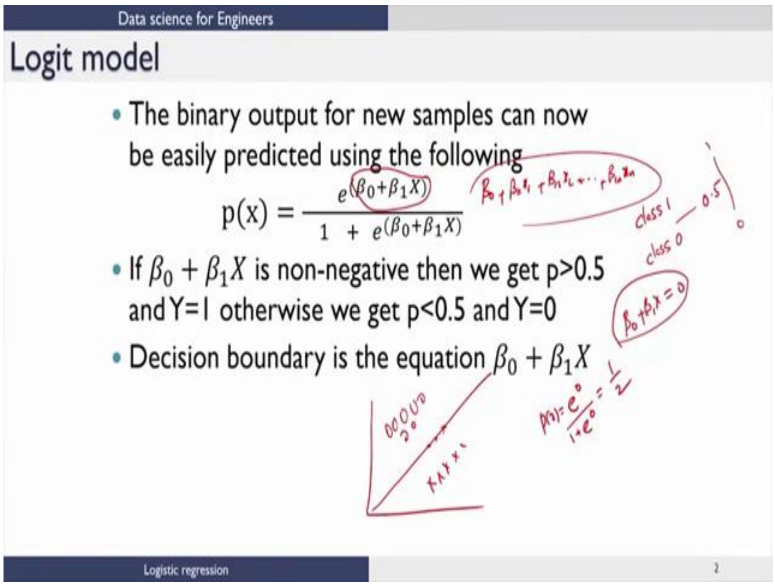
Python for Data Science
Prof. Ragunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 38
Logistic Regression (Continued)

We will continue our lecture on Logistic Regression that we introduced in the last lecture. And, if you recall from the last lecture, we modeled the probability as a sigmoidal function and the sigmoidal function that we used is given here. And notice that this is your hyperplane equation.

(Refer Slide Time: 00:27)



Data science for Engineers

Logit model

- The binary output for new samples can now be easily predicted using the following

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

- If $\beta_0 + \beta_1 X$ is non-negative then we get p>0.5 and Y=1 otherwise we get p<0.5 and Y=0
- Decision boundary is the equation $\beta_0 + \beta_1 X$

Logistic regression 2

$\beta_0+\beta_1x_1+\beta_2x_2+\dots+\beta_nx_n$
class 1
0.5
class 0
$\beta_0+\beta_1x=0$
$p(x)=\frac{e^0}{1+e^0}=\frac{1}{2}$

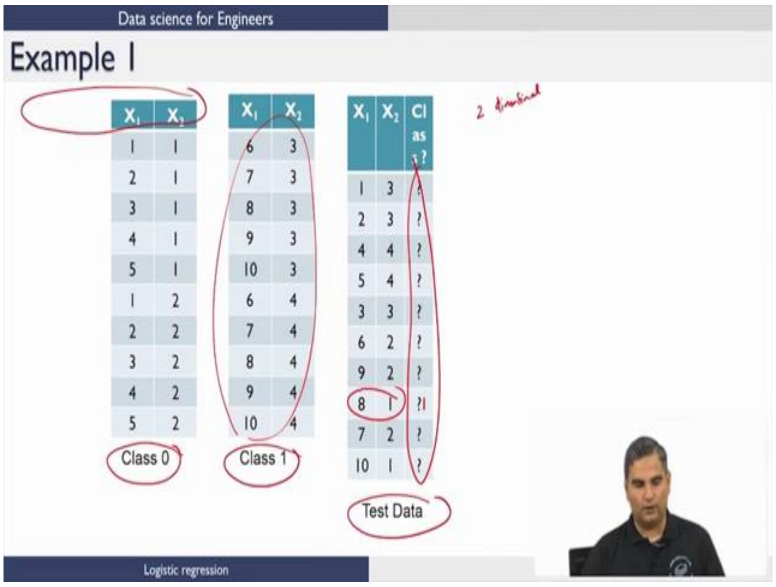
And in n dimensions this quantity is a scalar because you have x element, n elements in x and n elements in β_1 and this becomes something like $\beta_0 + \beta_1x_1 + \beta_2x_2 + \ldots + \beta_nx_n$. So, this is a scalar. And, then we saw that if this quantity is a very large negative number then the probability is 0 and if this quantity is a very large positive number the probability is 1 and, the transition of the probability at 0.5.

So, remember I said you have to always look at it from one class's viewpoint. So, let us say if you want class 1 to have high probability and class 0 is a row problem low probability case, then you need to have a threshold that we described before that you could convert this into a binary output by using a threshold. So, if you were to use a threshold of 0.5

because probabilities go from 0 and 1, then you notice that this p(x) becomes 0.5 exactly when $\beta_0 + \beta_1 X$ x equals 0. This is because p(x) then equals $\frac{e^0}{1+e^0}$, which is equal to 1 by 2.

Also notice another interesting thing that this equation is then the equation of the hyper plane. So, if I had data like this, and data like this and if I draw this line any point on this line is the probability equal to 0.5, a point. That basically says that any point on this line in this 2D case or hyperplane in the n dimensional case will have an equal probability of belonging to either class 0 or class 1 which makes sense from what we are trying to do. So, this model is what is called a Logit Model.

(Refer Slide Time: 02:51)


Data science for Engineers

Example I

X_1	X_2
1	1
2	1
3	1
4	1
5	1
1	2
2	2
3	2
4	2
5	2

Class 0

X_1	X_2
6	3
7	3
8	3
9	3
10	3
6	4
7	4
8	4
9	4
10	4

Class 1

X_1	X_2	Class?
1	3	?
2	3	?
4	4	?
5	4	?
3	3	?
6	2	?
9	2	?
8	1	?!
7	2	?
10	1	?

Test Data

Logistic regression

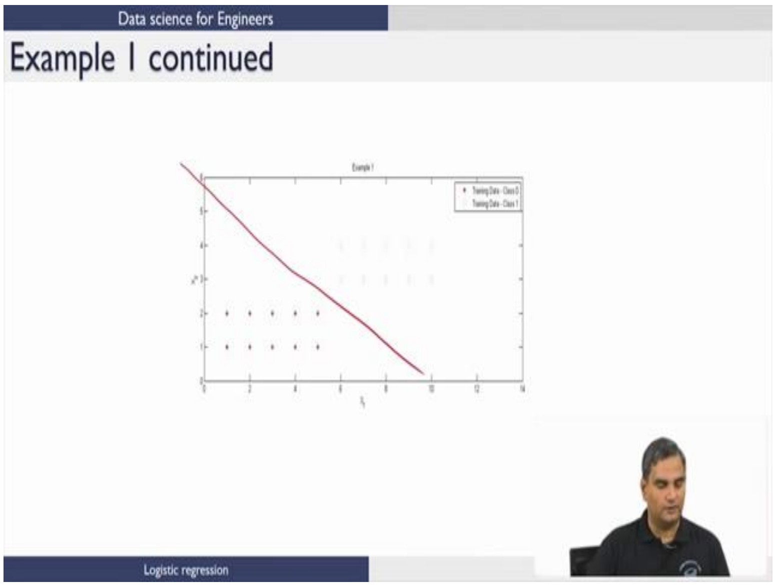
Let us take a very simple example to understand this. So, let us assume that we are given data. So, here we have data for class 0 and data for class 1 and then clearly this is a two-dimensional problem. So, the hyperplane is going to be a line. So, a line will separate this. And, in a typical case in these kinds of classification problems this is actually called as supervised classification problem. We call this a supervised classification problem because all of this data is labeled. So, I already know that all of this data is coming from class 0 and all of this data is coming from class 1. So, in other words I am being supervised in terms of what I should call as class 0 and what I should call as class 1.

So, in these kinds of problems typically you have this and then you are given a new data which is called the test data. And then the question is what class does this test data belong to. So, it is either class 0 or class 1 as far as we are concerned in this example. Just to keep in mind that there would be used problems like this. Remember, at the beginning of this course I talked about fraud detection and so on, where you could have lots of records of fraudulent let us say credit card use and all of those instances of fraudulent credit card use you could describe by certain attribute.

So, for example, the time of the day, whether the credit card was done at the place where the person lives, credit card transfer or a credit card use was done at the place the person lives and many other attributes. So, if those are the attributes let us say many attribute are there, and you have lots of records for normal use of credit card and some records for fraudulent use of credit card, then you could build a classifier which given a new set of attributes that is a new transaction that is being initiated could identify what likelihood it is of this transaction being fraudulent. So, that is one other way of thinking about the same problem.

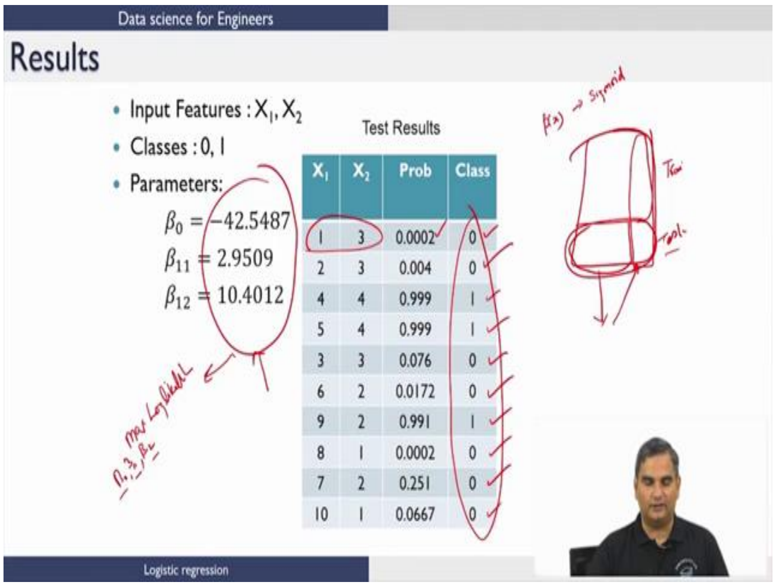
So, nonetheless as far as this example is concerned what we need to do is we have to fill this column with 0s and 1s. If I fill a column with a row with 0 then that means, this data belongs to class 0 and if I fill it one then let us say this belongs to class 1 and so on. So, this is what we are trying to do. We do not know what the classes are.

(Refer Slide Time: 05:31)



So, just so, let me see this it is a very simple problem. We have plotted the same data that was shown in the last table and you would notice that if you wanted a classifier something like this would do. So, this problem is linearly separable, so it could you could come up with a line that does it. So, let us see what happens if we use logistic regression to solve this problem.

(Refer Slide Time: 06:02)



Data science for Engineers

Results

- Input Features : X_1, X_2
- Classes : 0, 1
- Parameters:

$\beta_0 = -42.5487$

$\beta_{11} = 2.9509$

$\beta_{12} = 10.4012$

Test Results

X_1	X_2	Prob	Class
1	3	0.0002	0
2	3	0.004	0
4	4	0.999	1
5	4	0.999	1
3	3	0.076	0
6	2	0.0172	0
9	2	0.991	1
8	1	0.0002	0
7	2	0.251	0
10	1	0.0667	0

Logistic regression

So, if you did a logistic regression solution then in this case it turns out that the parameter values are these. And how did we get these parameter values? These parameters values are guard through the optimization formulation, where one is maximizing log likelihood with β_0, β_{11} and β_{12} as decision variables. And as we see here there are 3 decision variables because this was a two-dimensional problem; so, one coefficient for each dimension and then one constant.

Now, once you have this then what you do is you have your expression for p(x) which is as written before the sigmoid. So, this is a sigmoidal function that we have been talking about then whenever you get a test data let us say 1, 3, you plug this into this sigmoidal function and you get a probability. Let us say the first data point when you plug in you get a probability this. So, if we use a threshold of 0.5, then what we are going to say is anything less than 0.5 is going to belong to class 0, and anything greater than 0.5 is going to belong to class 1. So, you will notice that this is 0, class 0, class 1, class 1, class 0, class 0, class 1, class 0, class 0, class 0, right.

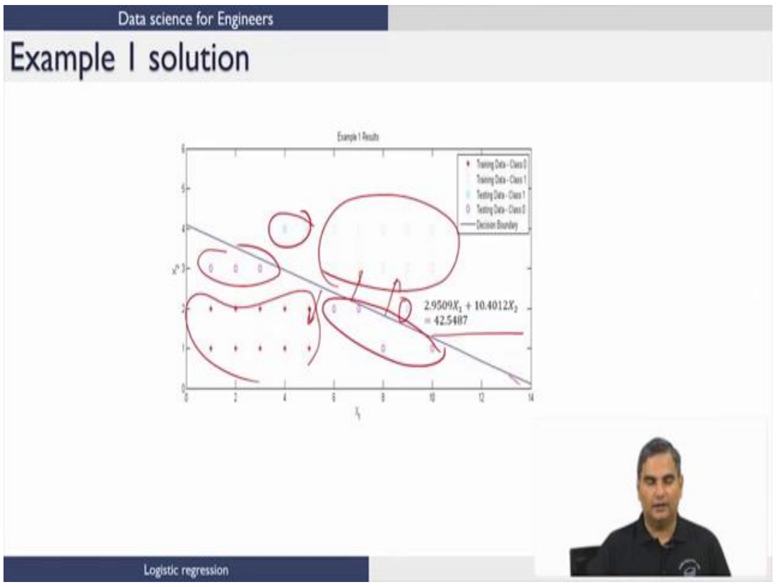
So, as I mentioned in the previous slide what we wanted was to fill this column and if you go across row then it says that particular sample belongs to which class. So, now, what we have done is we have classified these test cases which the classifier did not see while you were identifying these parameters. So, the process of identifying these parameters is what is usually called in machine learning algorithms as training. So, you are training the classifier to be able to solve test cases later. On the data that you use while these parameters are being identified are called the training data and this is called the test data, that you are testing a classifier with.

So, typically, what you do is if you have lots of data with class labels already given. One of the good things you know that one should do is to split this into training data and the test data and the reason for splitting this into training and test data is the following. In this case, if you look at it, we built a classifier based on some data and then we tested it on some other data, but we have no way of knowing whether these results are right or wrong, right. So, we just have to take the results as it is.

So, ideally what you would like to do is you would like to use some portion of the data to build a classifier and then you want to retain some portion of the data for testing and the reason for retaining this is because the labels are already known in this. So, if I just give this portion of the data to the classifier the classifier will come up with some classification, now that can be compared with the already established labels for those data points, right. So, from verifying how good your classifier is, it is always a good idea to split this into training and testing data; what proportion of data you use for training, what proportion of data use for testing, and so on are things to think about.

Also, there are many different ways of doing this validation as one would call it with this data; there are techniques such as k fold validation and so on. So, there are many ways of splitting the data into train and test and then verifying how good your classifier is. Nonetheless, the most important idea to remember is that one should always look at data and partition the data into training and testing, so that you get results that are consistent.

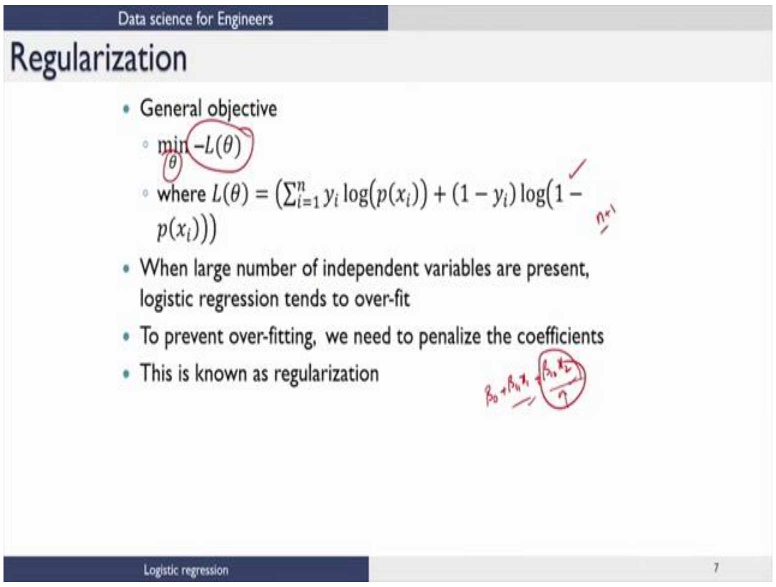
(Refer Slide Time: 10:21)



So, if one were to draw these points again that that we use this in this exercise. So, these are all class 1 data points, these are class 0 data points and this is your hyperplane that a logistic regression model figured out and these are the test points that we tried with this classifier. So, you can see that in this case everything seems to be working well, but as I said before you can look at results like this in two dimensions quite easily.

However, when there are multiple dimensions it is very difficult to visualize where the data point lies and so on. Nonetheless so, it gives you an idea of what logistics regression doing. It is actually doing a linear classification here. However, based on the distance in some sense from this hyperplane we also assign a probability for the data being in a particular class.

(Refer Slide Time: 11:37)



Data science for Engineers

Regularization

- General objective
  - $\min_{\theta} -L(\theta)$
  - where $L(\theta) = \left(\sum_{i=1}^{n} y_i \log(p(x_i)) + (1 - y_i)\log(1 - p(x_i))\right)$ ✓ $n+1$
- When large number of independent variables are present, logistic regression tends to over-fit
- To prevent over-fitting, we need to penalize the coefficients
- This is known as regularization

$\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$

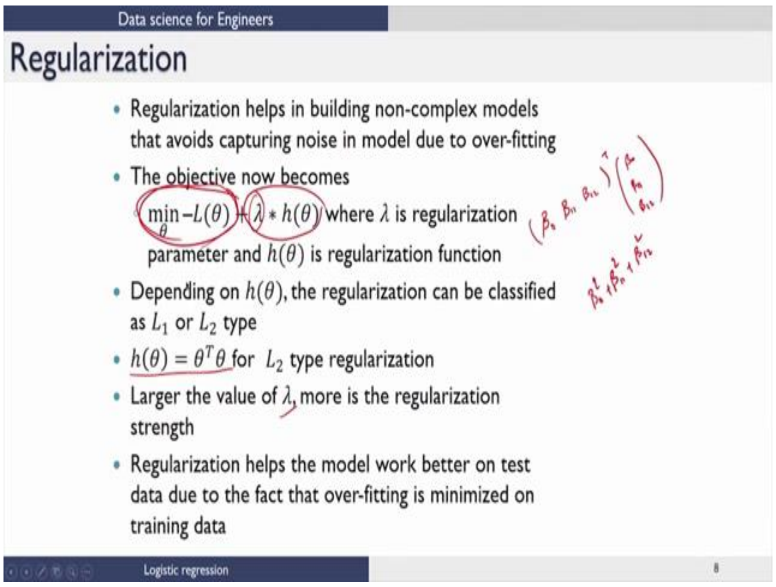
Logistic regression 7

Now, there is one more idea that we want to talk about in logistic regression. This idea is what is called as regularization. The idea here is the following. If you notice the objective function that we used in the general logistics regression which is what we call as a log likelihood objective function. Here theta again speaks to the constants in the hyperplane are the decision variables and this is the form of the equation that we saw in the previous lecture and in the beginning of this lecture also I believe. Now, if you have n variables in your problem or n features or n attributes then the number of decision variables that you are identifying are n + 1.

So, one constant for each variable and the constant if this n becomes very large then there are large number of variables that represent, but then what happens is this logistic regression models can over fit because there are so many parameters that you could tend to over fit the data. So, to proven this what we want to do is somehow we want to say though you have this n + 1 decision variables to use, one would want these decision variables to be used sparingly. So, whenever you use a coefficient for a variable for the classification problem then we want ensure that you get the maximum value for using that variable in the classification problem.

So, in other words, if let us say there are two variables I say $\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$, then for this classifier I am using both let us say variables x_1 and x_2 as being important. What one would like to do is make sure that I use these only if they really contribute to the solution or to the efficacy of the solution. So, one might say that for every term that you use you should get something in return or in other words if you use a term and get nothing in return I want to penalize this term. So, I want to penalize these coefficients. This is what is typically called as regularization.

(Refer Slide Time: 14:20)


Data science for Engineers

Regularization

- Regularization helps in building non-complex models that avoids capturing noise in model due to over-fitting
- The objective now becomes $\min_{\theta} -L(\theta) + \lambda * h(\theta)$ where λ is regularization parameter and $h(\theta)$ is regularization function
- Depending on $h(\theta)$, the regularization can be classified as L_1 or L_2 type
- $h(\theta) = \theta^T\theta$ for L_2 type regularization
- Larger the value of λ, more is the regularization strength
- Regularization helps the model work better on test data due to the fact that over-fitting is minimized on training data

$(\beta_0\ \beta_{11}\ \beta_{12})^T \begin{pmatrix}\beta_0 \\ \beta_{11} \\ \beta_{12}\end{pmatrix}$

$\beta_0^2 + \beta_{11}^2 + \beta_{12}^2$

Logistic regression 8

So, regularization avoids building complex models or it helps in building non-complex models so, that your fitting effects can be reduced. So, how do we penalize this? So, notice that what we are trying to do is we are trying to minimize a log likelihood. So, what we do here is we add another term to the objective and lambda is called the regularization parameter and this h theta is some regularization function.

So, what we want to do is when I choose the values of theta to be very large, I want this function to be large, so that the penalty is more or whenever I choose a variable, right away a penalty kicks in. And this penalty should be offset by the improvement I have in this term of the objective function, so that is the basic idea behind regularization.

Now, this function could be of many types if you use this function to be theta transpose theta then this is called L 2 type regularization. So, in the previous example this will turn out to be theta would be $[\beta_0\ \beta_{11}\ \beta_{12}]$, transpose times $[\beta_0\ \beta_{11}\ \beta_{12}]$. So, in this case h of theta will become $\beta_0^2+\beta_{11}^2+\beta_{12}^2$.

Now, there are other types of regularization that you can use. You can use this is what is called the L 2 type or L 2 norm. You can also use something called an L 1 type or an L 1 norm. And larger the value of this coefficient that is multiplying this and the more is regularization strength that is you are penalizing for use of variables lot more. And, one general rule is regularization helps the model work better with test data because you avoid over fitting on the trained data, so that is in general something that one can keep in mind, as one does these kinds of problems.

So, with this the portion on logistic regression comes to an end. What we are going to do next is we are going to show you an example case study, where logistics regression is used for a solution. However, before we do this case study; since all the case studies on classification and clustering, will involve looking at the output from the arc code. I am going to take a typical output from the arc code and there are several results that will show up, these are called performance measures of a classifier.

I am going to describe what these performance measures are and how you should interpret these performance measures once you use a particular technique for any case study. So, in the next lecture we will talk about these performance measures. And then following that will be the lecture on a case study using logistic regression, right.

So, thank you for listening to this lecture. And, I will see you in the next lecture.