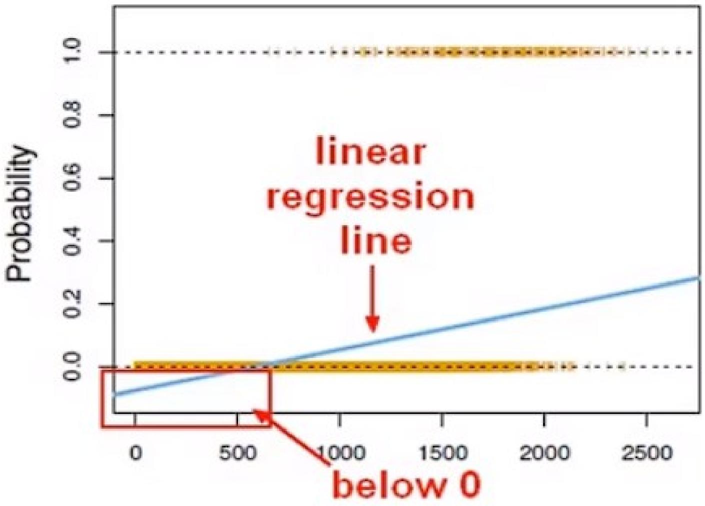
The logistic regression as a method for classification for machine learning and statistics.

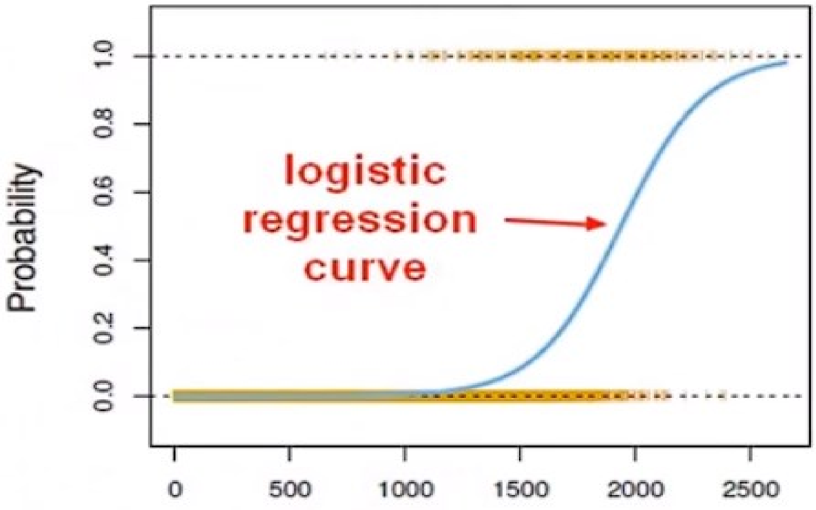
Classification is the problem of identifying to which of a set of categories a new observation belongs to base on the training data. Some examples of classification problems include trying to detect spam vs hams emails trying to detect. Whether someone's going to defaulter on their loan or not, or even trying to diagnose a disease for example trying to tell if someone has cancer or not.

The examples mentioned above are all examples of binary classification, which means that there are only two classes. Regression problems, aim to predict continuous values like home prices. When trying to predict discrete categories, classification problems can be solved with the help of logistic regression. In binary classification, the two classifications 0 and 1 are the standard.

Normal linear regression will not be good fit.

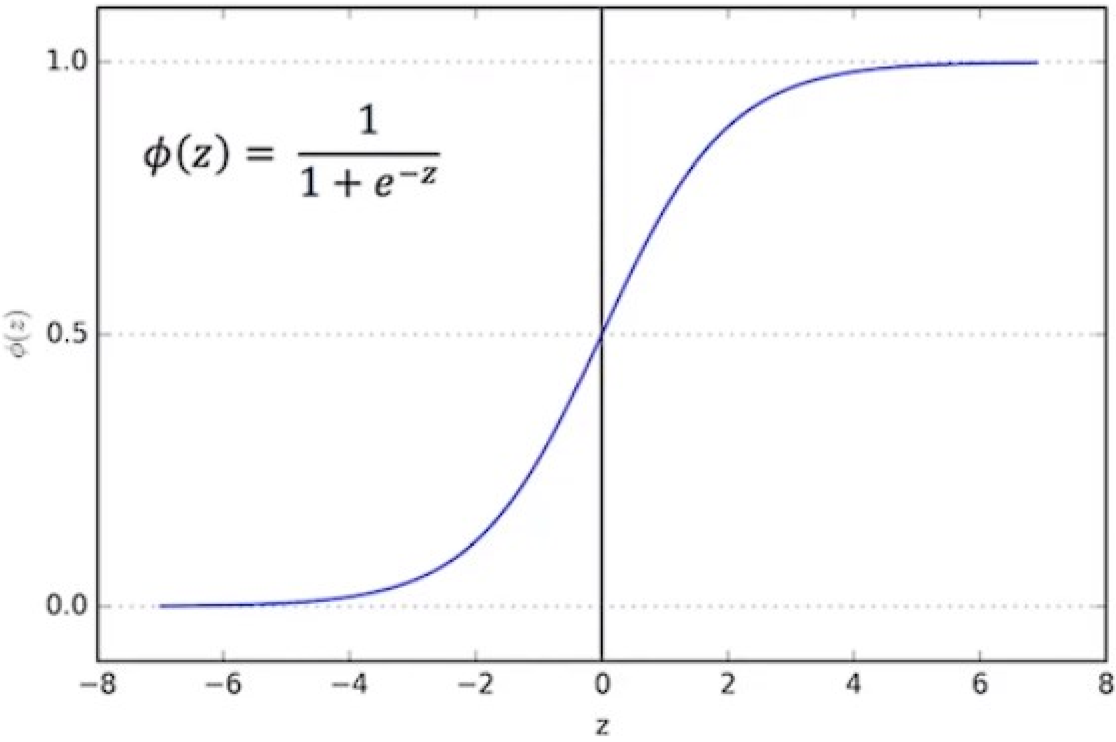


So, transform linear regression to a logistic regression curve.

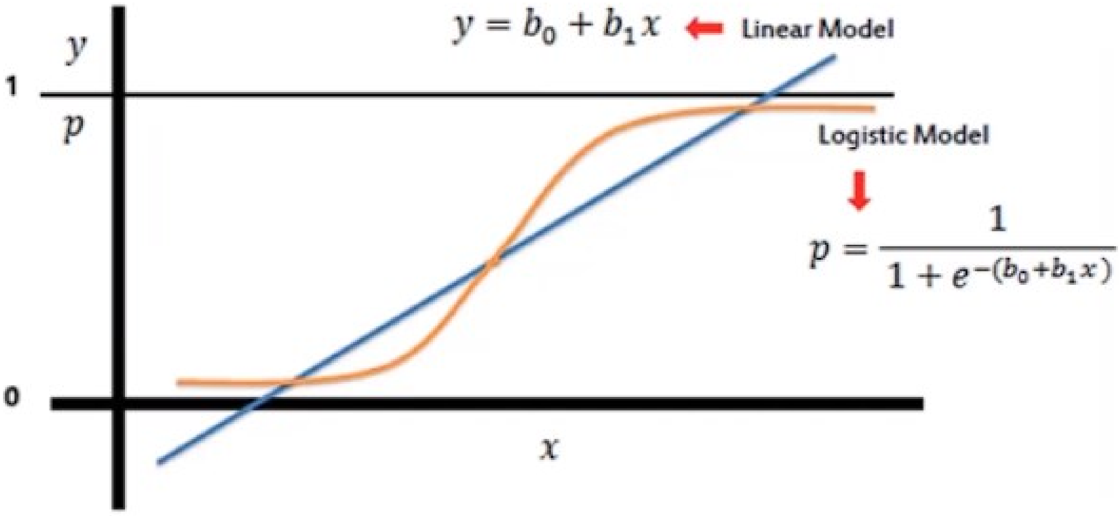


Sigmoid Function

The sigmoid function also known as logistic function is going to be the key to understanding using logistic regression to perform a classification. This function's main feature is that it takes any value as input and outputs a value between 0 and 1.



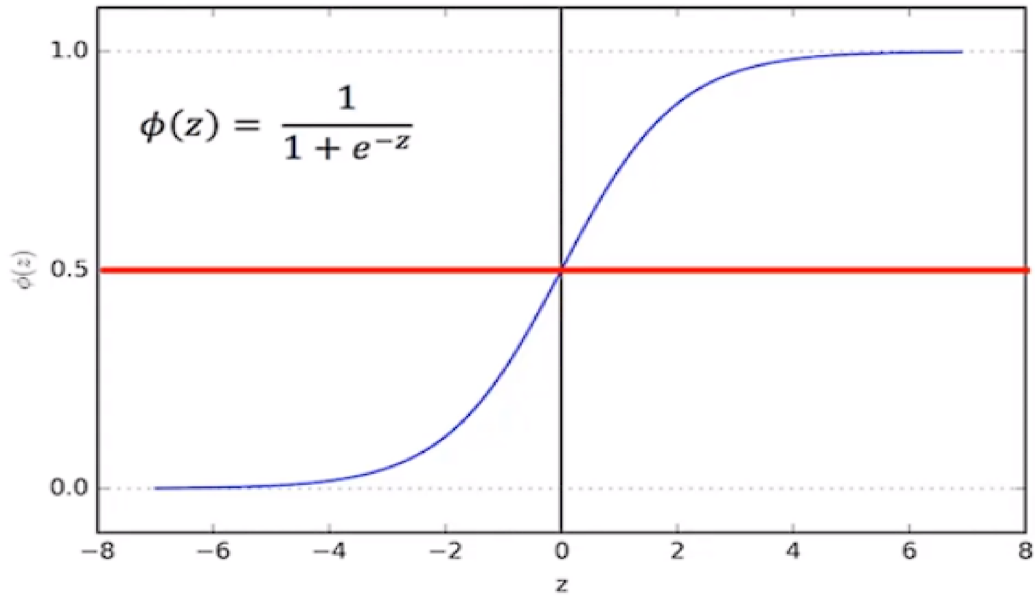
The key feature of sigmoid function is the output along the vertical axis will always be between 0 and 1, regardless of the value you provide for Z.



In linear model followed a basic y = x plus B principle i.e. y equals beta plus Beta 1 times X.

Finally, we can transform this linear regression into a logistic model by taking that linear model and putting it into the sigmoid function. This means that the value of the linear model output is not important. It will always fall between 0 and 1 when it is put into the sigmoid function or logistic model.

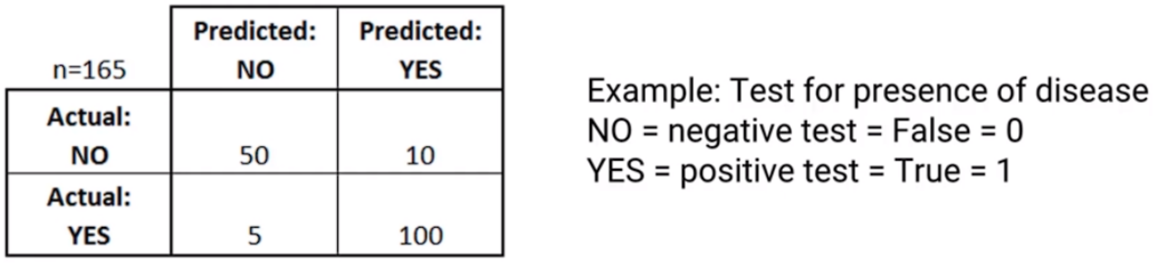
The fundamental idea behind all of this is that, regardless of what we enter on the horizontal access on the vertical axis, we will always receive a probability between 0 and 1. These results to a probability from 0 to 1 of belonging in the same class. That means we can set a cutoff point usually at 0.5 and we'll say if anything below results in 0.5 will go to class 0. Anything above belongs to class 1. Therefore, we will use that 0.5 probability as the cutoff value.



Confusion matrix

After you train a logistic regression model to classify some training data. And you want to evaluate your model's performance on some test data. confusion matrix can be used to evaluate classification models. A confusion matrix is a table that's frequently used for clarifying how well a classification model performs when applied to a set of test data whose true values are known. Although at times related terminology can be confusing, the confusion matrix itself is quite simple to understand.

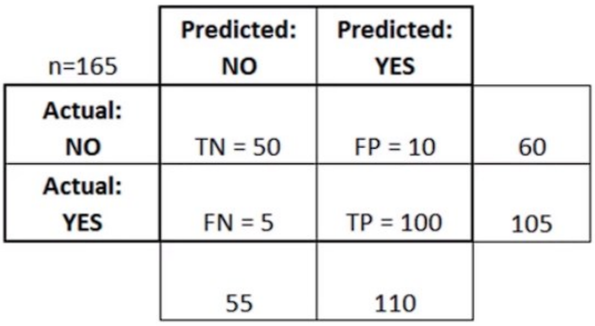
For example:



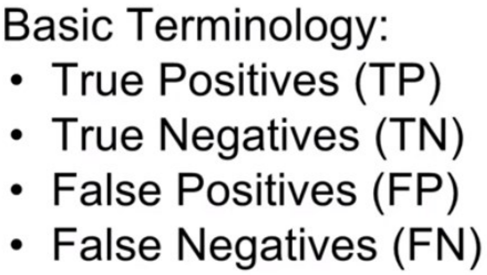
n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Example: Test for presence of disease
NO = negative test = False = 0
YES = positive test = True = 1

There are two possible predicted classes. (Yes and no). The classifier made a total of 165 predictions meaning 165 patients were tested for the presence of the disease. In that, out of those 165 cases the classifier predicted a Yes =110 times and No=55 times. In reality meaning we already have label test data 105 patients in the sample have the disease and 60 patients do not.



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	



Basic Terminology:

- True Positives (TP)
- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)

TP (True positive): They have disease and they do have the disease. TP =100
TN (True Negative): They do have disease and in reality, they do have the disease. TN =50
FP (False positive): They have disease and they do not have the disease. (This also known as type1 error)
FN (False Negative): They do not have disease and in reality, they do have the disease. (This also known as type2 error)

The Accuracy can be calculated by (TP+TN)/total= 150/165= 0.91), In this case our model is 91% accurate. And for the error rate (FP+FN)/total= 15/165= 0.09. Overall, this is 9% error rate or misclassification rate.