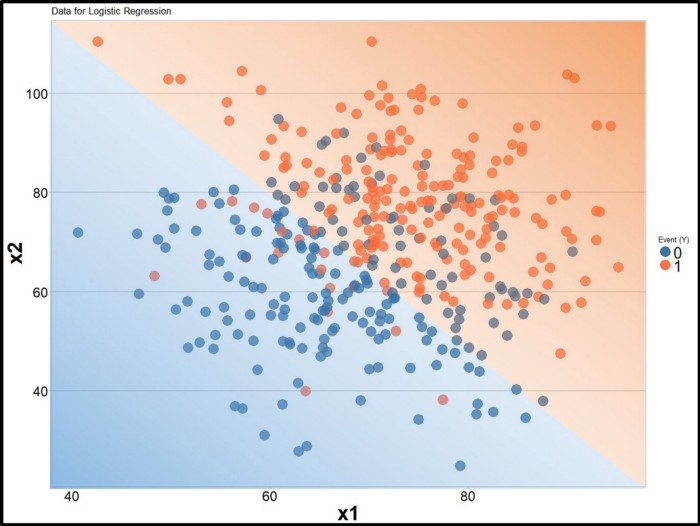
**LOGISTIC REGRESSION IN PYTHON**

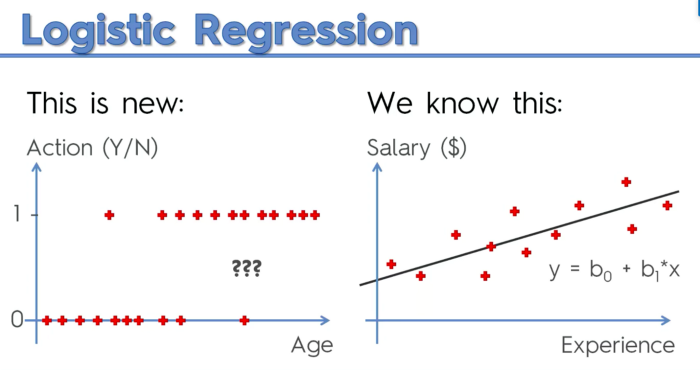


In statistics, the **logistic model** (or **logit model**) is a statistical model that is usually taken to apply to a binary dependent variable. In regression analysis, **logistic regression** or **logit regression** is estimating the parameters of a logistic model. More formally, a logistic model is one where the log-odds of the probability of an event is a linear combination of independent or predictor variables. The two possible dependent variable values are often labelled as “0” and “1”, which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. The binary logistic regression model can be generalized to more than two levels of the dependent variable: categorical outputs with more than two values are modelled by multinomial logistic regression, and if the multiple categories are ordered, by ordinal logistic regression, for example the proportional odds ordinal logistic model.

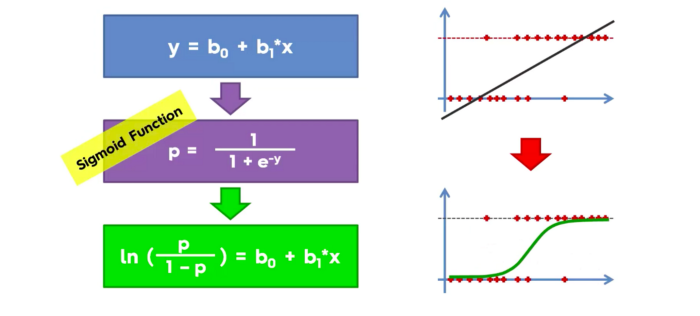
Logistic regression was developed by statistician David Cox in 1958.The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor. The model itself simply models probability of output in terms of input, and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other. The coefficients are generally not computed by a closed-form expression, unlike linear least squares; see § Model fitting.

So here is an example: did the person perform an action? (let’s say accept an offer or respond to an email). Intuitively we can understand that there is some sort of correlation between the customer whom will accept the offer and ones that will refuse it. The best method to solve this problem is to predict the probability of likelyhood of these 2 events happening.

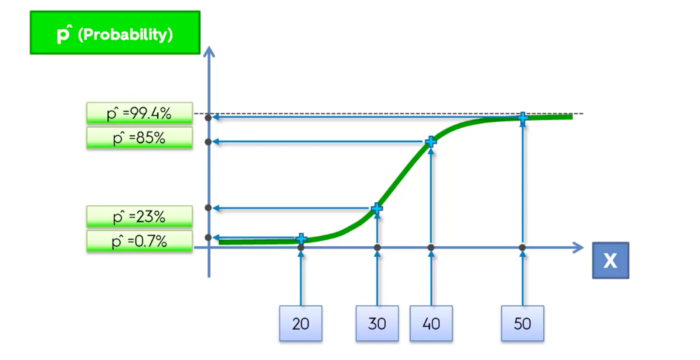
Right away you we see from the chart we have two values, 0 to 1:



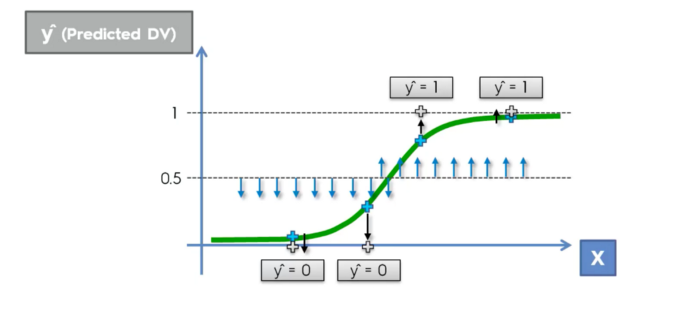
If we apply to the linear regression function the *sigmoid function* and solve for Y, we will get the green box function that we give us the logistic regression function:



The slope line from the linear regression corresponds to the best fitting line that can fit the data set. We can use this method to predict probabilities. Lets take 4 random values and project them to the logistic curve and get the fitted values:



Now if we project them to the left we will get the related probabilities and establish an arbitarry probability to estimate our outcome:



To implement LR in Python we need toimport the linear model library from sklearn and use the LogisticRegression class to create an object which is gonna be our classifier, and fit it to the training set:

Next we will introduce a new variable y\_pred vector of predictions and apply the predict method:

Next step is to create the confusion matrix from sklearn.matrix library we import the confusion\_matrix class, to verify the accuracy of our predictions:

Final step is to visualise the training set results and test set results through the ListedColorMap class:

And we are done!