Logistic regression

Logistic Regression is one of the simplest and commonly used Machine Learning algorithms for two-class classification. Logistic regression describes and estimates the relationship between one dependent binary variable and independent variables.

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function. Even though logistic/logit regression is frequently used for binary classes, it can be used for categorical dependent variables with more than 2 classes. In this case it's called Multinomial Logistic Regression

Multiclass classification with logistic regression can be done either through

- the one-vs-rest scheme in which for each class a binary classification problem of data belonging or not to that class is done or

- changing the loss function to cross-entropy loss.

It is named as Logistic Regression because it's underlying technique is quite the same as Linear Regression. The term Logistic is taken from the Logit function that is used in this method of classification.

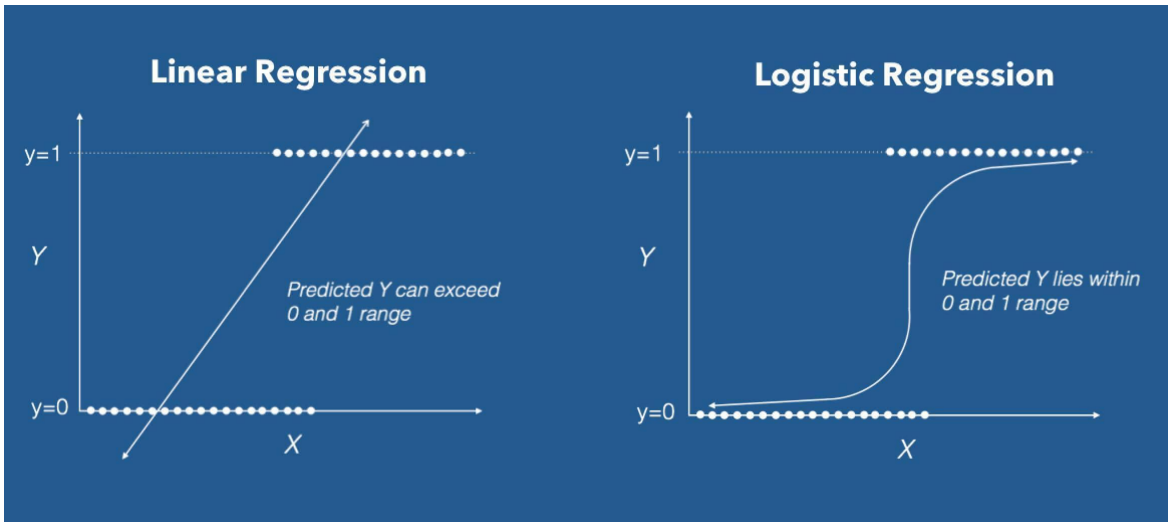
Linear Regression Vs. Logistic Regression

Linear regression is unbounded and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

Linear regression gives you a continuous output, but logistic regression provides a constant output. Linear regression model can generate the predicted probability as any number ranging from negative to positive infinity whereas probability of an outcome can only lie between 0< P(x)<1.

Why not linear regression?

If you use linear regression to model a binary response variable, the resulting model may not restrict the predicted Y values within 0 and 1 and would predict probability as any number ranging from negative to positive infinity



Linear regression tries to predict the data by finding a linear straight line equation but log reg does not look at the relationship between the variables as a straight line. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome.

Why cost function which has been used for linear cannot be used for logistic?

Linear regression uses mean squared error as its cost function. If this is used for logistic regression, then it will be a non-convex function of parameters (theta). Gradient descent will converge into global minimum only if the function is convex.

Estimation of Regression Coefficients

Unlike linear regression model that uses Ordinary Least Square for parameter estimation, we use Maximum Likelihood Estimation. The maximum likelihood estimate is that set of regression coefficients for which the probability of getting the data we have observed is maximum.
If we have binary data, the probability of each outcome is simply π if it was a success, and 1−π otherwise. Therefore, we have the likelihood function:

$$\mathcal{L}(\beta; \boldsymbol{y}) = \prod_{i=1}^{N} (\frac{\pi_i}{1-\pi_i})^{y_i} (1-\pi_i)$$

To determine the value of parameters, log of likelihood function is taken as it does not change the properties of the function. The log-likelihood is differentiated and using iterative techniques like Newton method, values of parameters that maximize the log-likelihood are determined.

Important Terms

Odds Ratio

Describes the ratio between the probability that a certain, positive, event occurs and the probability that it doesn't occur – where positive refers to the "event that we want to predict"

Given a probability p, the corresponding odds are calculated as p / (1 – p). For example, if p=0.75, the odds are 3 to 1: 0.75/0.25

Decision Boundary

Decision boundary helps to differentiate probabilities into positive class and negative class. Decision boundary can be linear (a straight line) or non-linear (any shape but not a straight line).

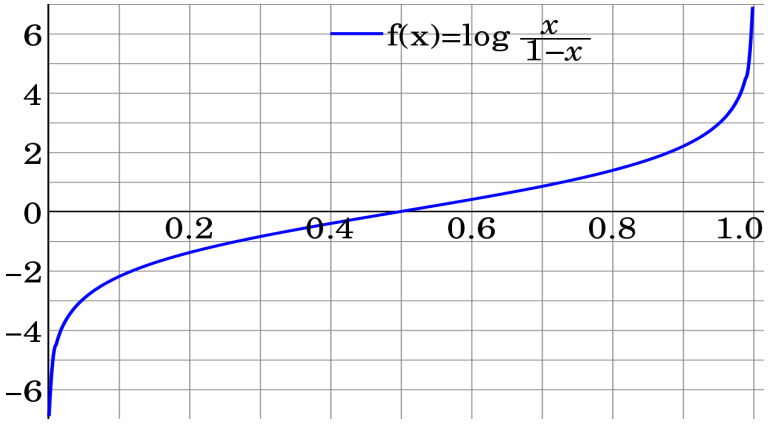
Logit function

The logit function is simply the logarithm of the odds. For binary data, the goal is to model the probability *p* that one of two outcomes occurs. The logit function (log[*p*/(1-*p*)]) varies between -∞ for y=0 and +∞ for y=1. The more likely it is that the positive event occurs, the larger the odds' ratio. Now, if we take the natural log of this odds' ratio, we get the following.

$\log[p/(1-p)] = b_0 + b_1x_1 + \ldots + b_kx_k$

All of this helps us understand that indeed the model is still a linear combination of the inputs, but that this linear combination relates to the log-odds of the default class.

Logic function plot:



Note: Logit function's equation is transformed into sigmoid function equation. The value of the logit function heads towards infinity as p approaches 1 and towards negative infinity as it approaches 0. The logit function is useful in analytics because it maps probabilities which are values in the range [0, 1] to the full range of real numbers.

p	odds	logodds
.001	.001001	-6.906755
.01	.010101	-4.59512
.15	.1764706	-1.734601
.2	.25	-1.386294
.25	.3333333	-1.098612
.3	.4285714	-.8472978
.35	.5384616	-.6190392
.4	.6666667	-.4054651
.45	.8181818	-.2006707
.5	1	0
.55	1.222222	.2006707
.6	1.5	.4054651
.65	1.857143	.6190392
.7	2.333333	.8472978
.75	3	1.098612
.8	4	1.386294
.85	5.666667	1.734601
.9	9	2.197225
.999	999	6.906755
.9999	9999	9.21024

Sigmoid function

In order to map predicted values to probabilities, we use the sigmoid function. The sigmoid function maps arbitrary real values back to the range [0, 1].

The score calculated by summation of multiplication of input and the corresponding weights is known as the logits. Sigmoid function is the inverse of the logit function. We can say sigmoid function existed from the below equation

$\log[p/(1-p)] = b_0 + b_1x_1 + \ldots + b_kx_{kj}$

So to convert above equation to sigmoid function, th e above equation becomes

$p = 1/(1 + \exp\{-(b_0 + b_1x_1 + \ldots + b_kx_k)\})$

or equivalently

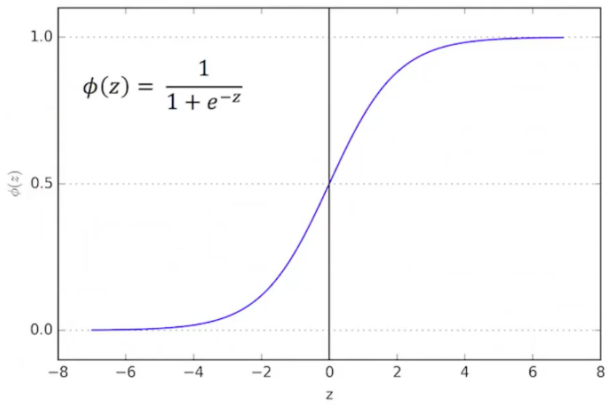
p = 1/ (1 + exp(-z))

p = output between 0 and 1 (probability estimate)

z = input to the function (your algorithm's prediction e.g. mx + b)

e = base of natural log

Here is a plot of the function:



The sigmoid might be useful if you want to transform a real valued variable into something that represents a probability

The biggest drawback of the sigmoid function for many analytics practitioners is the so-called vanishing gradient problem.

Assumptions of Logistic Regression

1. Target variable should be binary or ordinal

2. The dataset should not be unbalanced

3. There should be little or no multicollinearity between the features

Assumptions of Linear Regression not applicable for Logistics Regression

1. The error terms (residuals) do not need to be normally distributed

2. The error terms don't need to have constant variance

Example

Let's say we have a model that can predict whether a person is male or female based on their height (completely fictitious). Given a height of 150cm is the person male or female.

We have learned the coefficients of b0 = -100 and b1 = 0.6. Using the equation above we can calculate the probability of male given a height of 150cm or more formally P(male|height=150). We will use exp() for e, because that is what you can use if you type this example into your spreadsheet:

y = e ^ (b0 + b1*X) / (1 + e ^ (b0 + b1*X))

y = exp (-100 + 0.6*150) / (1 + EXP (-100 + 0.6*X))

y = 0.0000453978687

Or a probability of near zero that the person is a male. In practice we can use the probabilities directly. Because this is classification and we want a crisp answer, we can snap the probabilities to a binary class value.

For example:

0 if p(male) < 0.5

1 if p(male) >= 0.5