Logistic Regression
10/18/2025

Linear regression is to estimate the coefficients of a linear combination of a group of n explanatory variables (x) and a response variable (y), where w are the coefficients (or weights).

$$y = \sum_{i=1}^{n} w_i x_i + w_0$$

The ordinary least squares (OLS) method is often used to solve these coefficients. In R, the $glm()$ function uses iteratively re-weighted least squares (IWLS) to estimate the coefficients and provide the statistics. This function allows the user to specify the error distribution function of the response variable including the Gaussian, Binomial, Poisson and Gamma. For a Gaussian error distribution, the $glm()$ function becomes the ordinary $lm()$ function for most linear regression problems.

In a logistic regression, however, the response variable is no longer a simple measurement variable, but a logarithmic odds ratio. An odds ratio is the ratio of the probability of success (p) divided by the probability of failure ($1 - p$). The logarithmic odds ratio is also called "logit" by the statisticians.

$$log\left(\frac{p}{1-p}\right) = \sum_{i=1}^{n} w_i x_i + w_0$$

$$p = \frac{e^{logit}}{1 + e^{logit}} = \frac{1}{1 + e^{-logit}}$$

The above function for computing p is also known as the sigmoid function due to its "S" shape.

An Online Advertising Example

Figure 1 shows a portion of a dataset on the effectiveness of online advertising. The column of "Purchased" is the realized probability of success of the advertising and Logit is its computed logarithmic odds ratio. A small value (0.005) is added to the probability to avoid infinite values.

	UserID	Gender	Age	Salary	Purchased	Logit
1	15624510	0	19	19000	0	-5.303305
2	15810944	0	35	20000	0	-5.303305
3	15668575	1	26	43000	0	-5.303305
4	15603246	1	27	57000	0	-5.303305
5	15804002	0	19	76000	0	-5.303305
6	15728773	0	27	58000	0	-5.303305
7	15598044	1	27	84000	0	-5.303305
8	15694829	1	32	150000	1	5.303305
9	15600575	0	25	33000	0	-5.303305

Figure 1, Dataset of Online Advertising Effectiveness

If one directly uses the success probability ("Purchased") as the response variable for regression, function $glm()$ must be used and Binomial distribution selected because the success probability follows the Binomial distribution. Equivalently, one may uses the computed Logit as the response variable and uses either $glm()$ or $lm()$ function. The Gaussian distribution should be selected. Figure 2 shows the results of direct uses of the success probability with $glm()$ function.

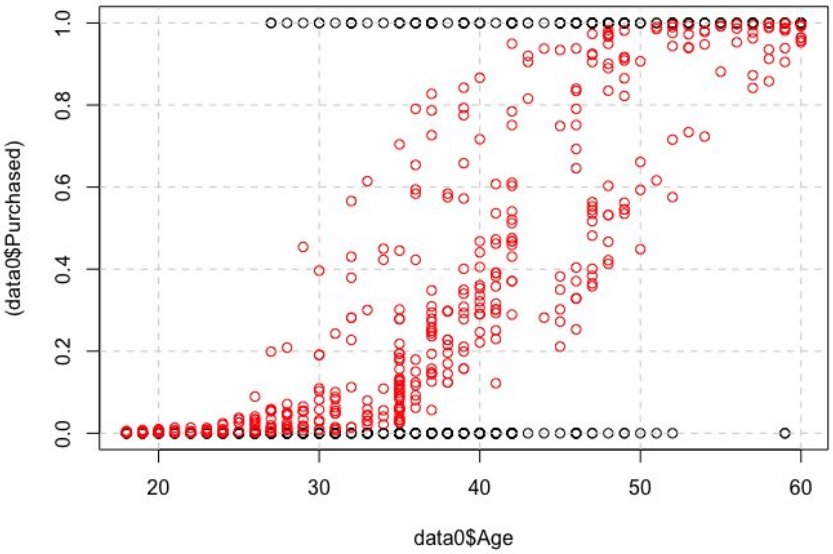

Figure 2, Regression with glm() function showing the probability of Purchased (black circles) and the fitted values (red circles).

The logit values along with the $lm()$ model yield similar results. In order to compare with results in Figure 2, the logit values must be converted to the probabilities using the sigmoid function (see Figure 3). Minor differences between the models exist; however the overall results are largely comparable.

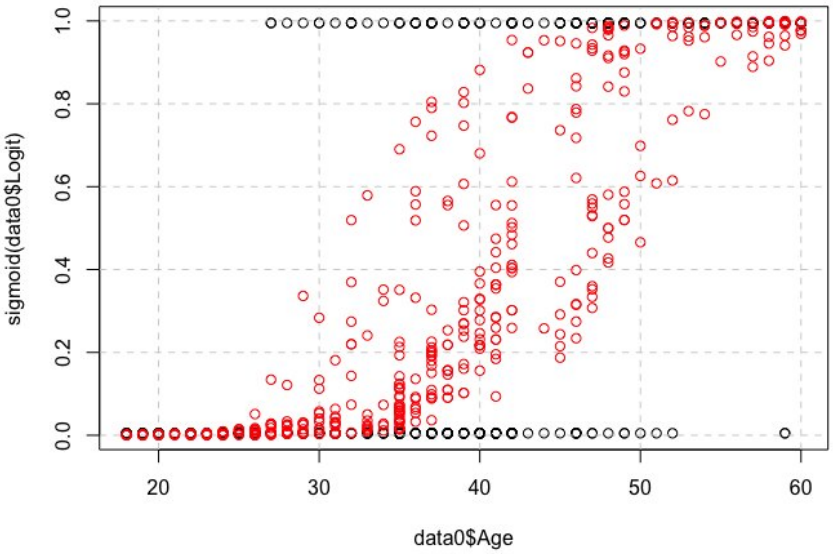

Figure 3, Regression with lm() function. The probability of Purchased (black circles) and the fitted values (red circles) are computed using the sigmoid function.

Feed-Forward Neural Network

In addition to the linear model $lm()$ and generic linear model $glm()$, the neural network from the library *nnet* can be used for logistic regression (as a classifier). This is a feed-forward neural network with a single hidden layer. The activation function is logistic by default. The optimization procedure follows the least squares method. Figure 4 shows the results.

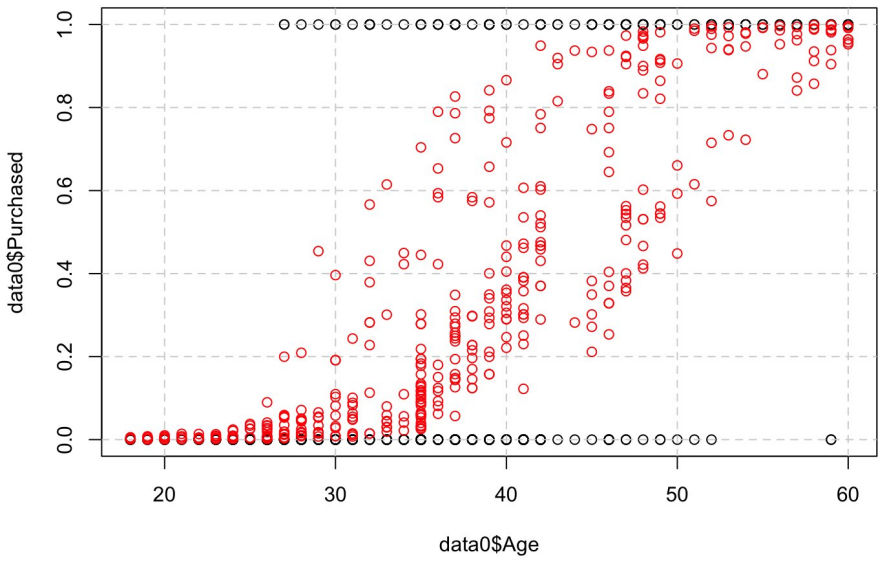

Figure 4, Regression with the multinom() function from the nnet library showing the probability of Purchased (black circles) and the fitted values (red circles).

Appendix R Script

```
### logistic regression ###
library(nnet)
eps=0.005
sigmoid=function(x){1/(1+exp(-x))}
logit=function(x){log((x+eps)/(1-x+eps))}
fn="advert.csv"; tbl=read.csv(fn, stringsAsFactors=TRUE); data0=data.frame(tbl)
data0$Logit=logit(data0$Purchased)

#logit0=glm(Logit ~ Female+Age+Salary, family=binomial, data = data0)
#logit0=lm(Logit ~ Female+Age+Salary, , data = data0)
logit0=multinom(Purchased ~ Female+Age+Salary, data = data0)

summary(logit0)

#plot(data0$Age, logit(data0$Purchased)); grid(lty=2)   # plot for glm() model
#points(data0$Age, logit(logit0$fitted), col="red")

#plot(data0$Age, sigmoid(data0$Logit)); grid(lty=2)   # plot for lm() model
#points(data0$Age, sigmoid(logit0$fitted), col="red")

plot(data0$Age, data0$Purchased); grid(lty=2)  # plot for multinom() model
points(data0$Age, (logit0$fitted.values), col="red")
```