What is the formula to calculate the R-Squared value for a generalized linear model (glm) in R?

Authored by

stats writer

June 29, 2024

The formula to calculate the R-Squared value for a Generalized Linear Model (GLM) in R is a statistical measure that quantifies the goodness of fit of the model. It represents the proportion of the total variation in the response variable that is explained by the model. The formula is expressed as the square of the correlation coefficient between the predicted values and the actual values of the response variable. It is used to evaluate the performance of a GLM and can range from 0 to 1, with higher values indicating a better fit. This formula is commonly used in statistical analysis and can help determine the effectiveness of the GLM in predicting the response variable.

Calculate R-Squared for glm in R

Often when we fit a linear regression model, we use R-squared as a way to assess how well a model fits the data.

R-squared represents the proportion of the variance in the that can be explained by the predictor variables in a regression model.

This number ranges from 0 to 1, with higher values indicating a better model fit.

However, there is no such R-squared value for general linear models like models and models.

Instead, we can calculate a metric known as McFadden's R-Squared, which ranges from 0 to just under 1, with higher values indicating a better model fit.

We use the following formula to calculate McFadden's R-Squared:

McFadden's R-Squared = 1 - (log likelihoodmodel / log likelihoodnull)

where:

log likelihoodmodel: Log likelihood value of current fitted modellog likelihoodnull: Log likelihood value of null model (model with intercept only)

In practice, values over 0.40 indicate that a model fits the data very well.

The following example shows how to calculate McFadden's R-Squared for a logistic regression model in R.

Example: Calculating McFadden's R-Squared in R

For this example, we'll use the Default dataset from the ISLR package. We can use the following code to load and view a summary of the dataset:

#install and load ISLR package

```
install.packages('ISLR')library(ISLR)

#define dataset
data <- ISLR::Default

#view summary of dataset
summary(data)

 default    student       balance        income
 No :9667   No :7056   Min.   :   0.0   Min.   :  772
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
                       Median : 823.6   Median :34553
                       Mean   : 835.4   Mean   :33517
                       3rd Qu.:1166.3   3rd Qu.:43808
                       Max.   :2654.3   Max.   :73554

#find total observations in dataset
nrow(data)

10000
```

This dataset contains the following information about 10,000 individuals:

default: Indicates whether or not an individual defaulted.student: Indicates whether or not an

individual is a student.balance: Average balance carried by an individual.income: Income of the individual.

We will use student status, bank balance, and income to build a logistic regression model that predicts the probability that a given individual defaults:

#fit logistic regression model
model <- glm(default~student+balance+income, family='binomial', data=data)

#view model summary
summary(model)

Call:
glm(formula = default ~ balance + student + income,
family = "binomial",
data = data)

Deviance Residuals:
Min 1Q Median 3Q Max
-2.4691 -0.1418 -0.0557 -0.0203 3.7383

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01 4.923e-01 -22.080 < 2e-16 *

balance 5.737e-03 2.319e-04 24.738 < 2e-16 *
studentYes -6.468e-01 2.363e-01 -2.738 0.00619 **
income 3.033e-06 8.203e-06 0.370 0.71152

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

(Dispersion parameter for binomial family taken to be 1)

**Null deviance: 2920.6 on 9999 degrees of freedom
Residual deviance: 1571.5 on 9996 degrees of freedom
AIC: 1579.5**

Number of Fisher Scoring iterations: 8

Next, we'll use the following formula to calculate McFadden's R-squared value for this model:

**#calculate McFadden's R-squared for model
with(summary(model), 1 - deviance/null.deviance)**

0.4619194

McFadden's R-squared value turns out to be 0.4619194. This value is fairly high, which indicates that our model fits the data well and has high predictive power.

Also note that we could use the pR2() function from the pscl package to calculate McFadden's R-square value for the model as well:

```
#install and load pscl package
install.packages('pscl')
library(pscl)

#calculate McFadden's R-squared for model
pR2(model)

McFadden
0.4619194
```

Notice that this value matches the one calculated earlier.

Additional Resources

The following tutorials explain how to perform other common tasks in R: