Can I use the predict() function with a Logistic Regression Model in R?

Authored by

stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *Can I use the predict() function with a Logistic Regression Model in R?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=149547

ARABPSYCHOLOGY.COM

The predict() function in R can be used with a Logistic Regression Model to make predictions based on the model's fitted coefficients. This function takes in the model object as well as new data to generate predicted values. It is a useful tool for evaluating the performance of the model and making predictions on new data. However, it is important to note that the predict() function is only applicable for models with a binary outcome or those that have been converted to a binomial family using the glm() function.

Use predict() with Logistic Regression Model in R

Once we've fit a in R, we can use the predict() function to predict the response value of a new observation that the model has never seen before.

This function uses the following syntax:

predict(object, newdata, type="response")

where:

object: The name of the logistic regression modelnewdata: The name of the new data frame to make predictions fortype: The type of prediction to make

The following example shows how to use this function in practice.

Example: Using predict() with a Logistic Regression Model in R

ARABPSYCHOLOGY.COM

For this example, we'll use the built-in R dataset called :

#view first six rows of *mtcars* dataset
head(mtcars)

mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4
Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4
Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1
Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1
Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2
Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

We'll fit the following logistic regression model in which we use the variables disp and hp to predict the response variable am (the transmission type of the car: 0 = automatic, 1 = manual):

#fit logistic regression model
model <- glm(am ~ disp + hp, data=mtcars, family=binomial)

#view model summary
summary(model)

ARABPSYCHOLOGY.COM

Call:
glm(formula = am ~ disp + hp, family = binomial, data = mtcars)

Deviance Residuals:
Min 1Q Median 3Q Max
-1.9665 -0.3090 -0.0017 0.3934 1.3682

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.40342 1.36757 1.026 0.3048
disp -0.09518 0.04800 -1.983 0.0474 *
hp 0.12170 0.06777 1.796 0.0725 .

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 43.230 on 31 degrees of freedom
Residual deviance: 16.713 on 29 degrees of freedom
AIC: 22.713

Number of Fisher Scoring iterations: 8

We can then create a new data frame that contains

ARABPSYCHOLOGY.COM

information about eight cars the model has never seen before and use the predict() function to predict the probability that a new car has an automatic transmission (am=0) or a manual transmission (am=1):

```
#define new data frame
newdata = data.frame(disp=c(200, 180, 160, 140, 120, 120, 100, 160),
hp=c(100, 90, 108, 90, 80, 90, 80, 90),
am=c(0, 0, 0, 1, 0, 1, 1, 1))

#view data frame
newdata

#use model to predict value of am for all new cars
newdata$am_prob <- predict(model, newdata, type="response")

#view updated data frame
newdata

disp hp am am_prob
1 200 100 0 0.004225640
2 180 90 0 0.008361069
3 160 108 0 0.335916069
```

ARABPSYCHOLOGY.COM

4 140 90 1 0.275162866

5 120 80 0 0.429961894

6 120 90 1 0.718090728

7 100 80 1 0.835013994

8 160 90 1 0.053546152

Here's how to interpret the output:

The probability that car 1 has a manual transmission is .004.The probability that car 2 has a manual transmission is .008.The probability that car 3 has a manual transmission is .336.

And so on.

We can also use the table() function to create a confusion matrix that displays the actual am values vs. the predicted values by the model:

```
#create vector that contains 0 or 1 depending on predicted value of am
am_pred = rep(0, dim(newdata))
am_pred = 1

#create confusion matrix
```

ARABPSYCHOLOGY.COM

```
table(am_pred, newdata$am)

am_pred 0 1
0 4 2
1 0 2
```

Lastly, we can use the mean() function to calculate the percentage of observations in the new data frame that the model correctly predicted the value of am for:

```
#calculate percentage of observations the model correctly predicted response value for
mean(am_pred == newdata$am)

0.75
```

We can see that the model correctly predicted the am value for 75% of the cars in the new data frame.

ARABPSYCHOLOGY.COM