How do I create a confusion matrix in R?

Authored by

stats writer

April 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I create a confusion matrix in R?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=140234

ARABPSYCHOLOGY.COM

A confusion matrix is a useful tool for evaluating the performance of a predictive model in machine learning. It is a table that summarizes the predictions made by the model against the actual outcomes. In R, creating a confusion matrix involves using the "confusionMatrix" function from the "caret" package. This function takes in the predicted and actual values as inputs and generates a table with the number of correct and incorrect predictions for each class. This allows for a quick and easy assessment of the model's accuracy and error rates. Additionally, the confusion matrix can be used to calculate other performance metrics such as precision, recall, and F1 score. Overall, understanding how to create a confusion matrix in R can aid in the evaluation and improvement of machine learning models.

Create a Confusion Matrix in R (Step-by-Step)

is a type of regression we can use when the response variable is binary.

One common way to evaluate the quality of a logistic regression model is to create a confusion matrix, which is a 2×2 table that shows the predicted values from the model vs. the actual values from the test dataset.

		Predicted	
		0	**1**
Actual	**0**	30	12
	1	8	56

The following step-by-step example shows how to create a confusion matrix in R.

ARABPSYCHOLOGY.COM

Step 1: Fit the Logistic Regression Model

For this example we'll use the Default dataset from the ISLR package. We'll use student status, bank balance, and annual income to predict the probability that a given individual defaults on their loan.

The following code shows how to fit a logistic regression model to this dataset:

```
#load necessary packages
library(caret)
library(InformationValue)
library(ISLR)

#load dataset
data <- Default

#split dataset into training and testing set
set.seed(1)
sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE, prob=c(0.7,0.3))
train <- data
test <- data

#fit logistic regression model
```

ARABPSYCHOLOGY.COM

```
model <- glm(default~student+balance+income, family="binomial", data=train)
```

Step 2: Create the Confusion Matrix

Next, we'll use the confusionMatrix() function from the caret package to create a confusion matrix:

```
#use model to predict probability of default
predicted <- predict(model, test, type="response")

#convert defaults from "Yes" and "No" to 1's and 0's
test$default <- ifelse(test$default=="Yes", 1, 0)

#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$default, predicted)

#create confusion matrix
confusionMatrix(test$default, predicted)

0 1
0 2912 64
1 21 39
```

ARABPSYCHOLOGY.COM

Step 3: Evaluate the Confusion Matrix

We can also calculate the following metrics using the confusion matrix:

Sensitivity: The "true positive rate" - the percentage of individuals the model correctly predicted would default.Specificity: The "true negative rate" - the percentage of individuals the model correctly predicted would *not* default.Total misclassification rate: The percentage of total incorrect classifications made by the model.

The following code shows how to calculate these metrics:

```
#calculate sensitivity
sensitivity(test$default, predicted)
```

0.3786408

```
#calculate specificity
specificity(test$default, predicted)
```

0.9928401

ARABPSYCHOLOGY.COM

#calculate total misclassification error rate
misClassError(test$default, predicted, threshold=optimal)

0.027

The total misclassification error rate is 2.7% for this model.

ARABPSYCHOLOGY.COM