How can I use the createDataPartition() function in R to split my dataset into training and testing sets?

Authored by

stats writer

June 25, 2024

The createDataPartition() function in R is a useful tool for splitting a dataset into training and testing sets. This function allows for a random partition of the data, ensuring that the resulting sets are representative of the entire dataset. To use this function, first load the "caret" package in R. Then, specify the dataset and the desired split ratio. The function will return a vector of indices that can be used to subset the dataset into training and testing sets. This process helps to prevent overfitting and allows for better model evaluation. Overall, the createDataPartition() function is a valuable tool for preparing data for machine learning and statistical analysis in R.

Use createDataPartition() Function in R

You can use the createDataPartition() function from the caret package in R to partition a data frame into training and testing sets for model building.

This function uses the following basic syntax:

createDataPartition(y, times = 1, p = 0.5, list = TRUE, ...)

where:

y: vector of outcomestimes: number of partitions to createp: percentage of data to use in training setlist: whether to store results in list or not

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

Example: Using createDataPartition() in R

Suppose we have some data frame in R with 1,000 rows that contains information about hours studied by students and their corresponding score on a final exam:

```
#make this example reproducible
set.seed(0)

#create data frame
df <- data.frame(hours=runif(1000, min=0, max=10),
score=runif(1000, min=40, max=100))

#view head of data frame
head(df)

  hours    score
1 8.966972 55.93220
2 2.655087 71.84853
3 3.721239 81.09165
4 5.728534 62.99700
5 9.082078 97.29928
6 2.016819 47.10139
```

Suppose we would like to fit a that uses hours studied to predict final exam score.

Suppose we would like to train the model on 80% of the rows in the data frame and test it on the remaining 20% of rows.

The following code shows how to use the createDataPartition() function from the caret package to split the data frame into training and testing sets:

```
library(caret)

#partition data frame into training and testing sets
train_indices <- createDataPartition(df$score, times=1, p=.8, list=FALSE)

#create training set
df_train <- df

#create testing set
df_test <- df

#view number of rows in each set
nrow(df_train)
```

800

```
nrow(df_test)
```

200

We can see that our training dataset contains 800 rows, which represents 80% of the original dataset.

Similarly, we can see that our test dataset contains 200 rows, which represents 20% of the original dataset.

We can also view the first few rows of each set:

#view head of training set
head(df_train)

hours score
1 8.966972 55.93220
2 2.655087 71.84853
3 3.721239 81.09165
4 5.728534 62.99700
5 9.082078 97.29928
7 8.983897 42.34600

#view head of testing set
head(df_test)

hours score
6 2.016819 47.10139

12 2.059746 96.67170

18 7.176185 92.61150

23 2.121425 89.17611

24 6.516738 50.47970

25 1.255551 90.58483