How can I split my data into training and test sets using R, and what are the three methods available for doing so?

Authored by

stats writer

June 28, 2024

Splitting data into training and test sets is an essential step in machine learning and data analysis. In R, there are three methods available for dividing data into training and test sets: random sampling, stratified sampling, and k-fold cross-validation.

Random sampling involves randomly selecting a portion of the data as the training set and the remaining as the test set. This method is suitable for large datasets and ensures that the training and test sets have similar characteristics.

Stratified sampling is used when the data has imbalanced classes. It involves dividing the data into groups based on a certain variable and then randomly sampling from each group to create the training and test sets. This method helps to ensure that the training and test sets have a similar distribution of classes.

K-fold cross-validation is a technique where the data is divided into k subsets, and each subset is used as the test set while the remaining subsets are used for training. This process is repeated k times, and the results are averaged to obtain a more reliable estimate of the model's performance.

Overall, these methods allow for the proper evaluation of a model's performance and help to prevent overfitting. It is crucial to carefully select the appropriate method based on the characteristics of the data to ensure accurate and reliable results.

Split Data into Training & Test Sets in R (3 Methods)

Often when we fit to datasets, we first split the dataset into a training set and a test set.

There are three common ways to split data into training and test sets in R:

Method 1: Use Base R

#make this example reproducible

set.seed(1)

```
#use 70% of dataset as training set and 30% as test set
sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.7,0.3))
train <- df
test <- df
```

Method 2: Use caTools package

```
library(caTools)

#make this example reproducible
set.seed(1)

#use 70% of dataset as training set and 30% as test set
sample <- sample.split(df$any_column_name, SplitRatio = 0.7)
train <- subset(df, sample == TRUE)
test <- subset(df, sample == FALSE)
```

Method 3: Use dplyr package

```
library(dplyr)

#make this example reproducible
set.seed(1)
```

```
#create ID column
df$id <- 1:nrow(df)

#use 70% of dataset as training set and 30% as test set
train <- df %>% dplyr::sample_frac(0.70)
test <- dplyr::anti_join(df, train, by = 'id')
```

The following examples show how to use each method in practice with the built-in in R.

Example 1: Split Data Into Training & Test Set Using Base R

The following code shows how to use base R to split the iris dataset into a training and test set, using 70% of the rows as the training set and the remaining 30% as the test set:

```
#load iris dataset
data(iris)

#make this example reproducible
set.seed(1)

#Use 70% of dataset as training set and remaining 30% as testing set
sample <- sample(c(TRUE, FALSE), nrow(iris),
```

```
replace=TRUE, prob=c(0.7,0.3))
train <- iris
test <- iris

#view dimensions of training set
dim(train)

106 5

#view dimensions of test set
dim(test)

44 5
```

From the output we can see:

The training set is a data frame with 106 rows and 5 columns.The test is a data frame with 44 rows and 5 columns.

Since the original data frame had 150 total rows, the training set contains roughly 106 / 150 = 70.6% of the original rows.

We can also view the first few rows of the training set if we'd like:

#view first few rows of training set
head(train)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
3 4.7 3.2 1.3 0.2 setosa
5 5.0 3.6 1.4 0.2 setosa
8 5.0 3.4 1.5 0.2 setosa
9 4.4 2.9 1.4 0.2 setosa

Example 2: Split Data Into Training & Test Set Using caTools

The following code shows how to use the caTools package in R to split the iris dataset into a training and test set, using 70% of the rows as the training set and the remaining 30% as the test set:

library(caTools)

#load iris dataset
data(iris)

#make this example reproducible

```
set.seed(1)

#Use 70% of dataset as training set and remaining 30% as testing set
sample <- sample.split(iris$Species, SplitRatio = 0.7)
train <- subset(iris, sample == TRUE)
test <- subset(iris, sample == FALSE)

#view dimensions of training set
dim(train)

105 5

#view dimensions of test set
dim(test)

45 5
```

From the output we can see:

The training set is a data frame with 105 rows and 5 columns.The test is a data frame with 45 rows and 5 columns.

Example 3: Split Data Into Training & Test Set Using dplyr

The following code shows how to use the caTools

package in R to split the iris dataset into a training and test set, using 70% of the rows as the training set and the remaining 30% as the test set:

```
library(dplyr)

#load iris dataset
data(iris)

#make this example reproducible
set.seed(1)

#create ID variable
iris$id <- 1:nrow(iris)

#Use 70% of dataset as training set and remaining 30% as testing set
train <- iris %>% dplyr::sample_frac(0.7)
test <- dplyr::anti_join(iris, train, by = 'id')

#view dimensions of training set
dim(train)

105 6

#view dimensions of test set
```

dim(test)

45 6

From the output we can see:

The training set is a data frame with 105 rows and 6 columns.The test is a data frame with 45 rows and 6 columns.

Note that these training and test sets contain one extra 'id' column that we created.

Be sure not to use this column (or drop it entirely from the data frames) when fitting your machine learning algorithm.

Additional Resources