Full Model Docs

Steph Locke

16 February 2017

We're working with data on who survived the Titanic.

## Source

We're collecting our data from a SQLite database. The titanic3 data was originally pulled in from the PASWR package and is the third major version of that dataset. It contains more features than the basic titanic dataset available in the datasets package.

library(DBI)  
library(RSQLite)  
titanicdb <- dbConnect(SQLite(),dbname= "../data-raw/titanic.sqlite")

## Data

We're using just a single table of data that has already been collated. Here is a quick overview of the data.

titanic\_all<-dbReadTable(titanicdb, "titanic")  
knitr::kable(head(titanic\_all))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pclass | survived | name | sex | age | sibsp | parch | ticket | fare | cabin | embarked | boat | body | home.dest |
| 1st | 1 | Allen, Miss. Elisabeth Walton | female | 29.0000 | 0 | 0 | 24160 | 211.3375 | B5 | Southampton | 2 | NA | St Louis, MO |
| 1st | 1 | Allison, Master. Hudson Trevor | male | 0.9167 | 1 | 2 | 113781 | 151.5500 | C22 C26 | Southampton | 11 | NA | Montreal, PQ / Chesterville, ON |
| 1st | 0 | Allison, Miss. Helen Loraine | female | 2.0000 | 1 | 2 | 113781 | 151.5500 | C22 C26 | Southampton |  | NA | Montreal, PQ / Chesterville, ON |
| 1st | 0 | Allison, Mr. Hudson Joshua Crei | male | 30.0000 | 1 | 2 | 113781 | 151.5500 | C22 C26 | Southampton |  | 135 | Montreal, PQ / Chesterville, ON |
| 1st | 0 | Allison, Mrs. Hudson J C (Bessi | female | 25.0000 | 1 | 2 | 113781 | 151.5500 | C22 C26 | Southampton |  | NA | Montreal, PQ / Chesterville, ON |
| 1st | 1 | Anderson, Mr. Harry | male | 48.0000 | 0 | 0 | 19952 | 26.5500 | E12 | Southampton | 3 | NA | New York, NY |

knitr::kable(summary(titanic\_all))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | pclass | survived | name | sex | age | sibsp | parch | ticket | fare | cabin | embarked | boat | body | home.dest |
|  | Length:1309 | Min. :0.000 | Length:1309 | Length:1309 | Min. : 0.1667 | Min. :0.0000 | Min. :0.000 | Length:1309 | Min. : 0.000 | Length:1309 | Length:1309 | Length:1309 | Min. : 1.0 | Length:1309 |
|  | Class :character | 1st Qu.:0.000 | Class :character | Class :character | 1st Qu.:21.0000 | 1st Qu.:0.0000 | 1st Qu.:0.000 | Class :character | 1st Qu.: 7.896 | Class :character | Class :character | Class :character | 1st Qu.: 72.0 | Class :character |
|  | Mode :character | Median :0.000 | Mode :character | Mode :character | Median :28.0000 | Median :0.0000 | Median :0.000 | Mode :character | Median : 14.454 | Mode :character | Mode :character | Mode :character | Median :155.0 | Mode :character |
|  | NA | Mean :0.382 | NA | NA | Mean :29.8811 | Mean :0.4989 | Mean :0.385 | NA | Mean : 33.295 | NA | NA | NA | Mean :160.8 | NA |
|  | NA | 3rd Qu.:1.000 | NA | NA | 3rd Qu.:39.0000 | 3rd Qu.:1.0000 | 3rd Qu.:0.000 | NA | 3rd Qu.: 31.275 | NA | NA | NA | 3rd Qu.:256.0 | NA |
|  | NA | Max. :1.000 | NA | NA | Max. :80.0000 | Max. :8.0000 | Max. :9.000 | NA | Max. :512.329 | NA | NA | NA | Max. :328.0 | NA |
|  | NA | NA | NA | NA | NA's :263 | NA | NA | NA | NA's :1 | NA | NA | NA | NA's :1188 | NA |

## Defensive stuff

### Store a copy (one-off)

We will use the cache file to keep the data file size small.

cache\_file<-"../data-raw/rawdatacache.Rdata"  
if(!file.exists(cache\_file)) {  
 titanic\_cache<-titanic\_all  
 save(titanic\_cache,file = cache\_file)  
 rm(titanic\_cache)  
 }

### Check for changes

Check that the original data is still the same as the cache data otherwise throw an error so you'll get things out.

load(cache\_file)  
if(!identical(titanic\_all, titanic\_cache)) stop("Hey, the data has changed, you should check that out!")

We could change things and rerun with:

dbWriteTable(titanicdb, "titanic", head(titanic\_all), append=TRUE)

Restore your original data with your get\_data.R file.

We want to split our data into a training and a test sample for predicting survival of the Titanic. There are two sampling strategies we'll consider here. Consider splitting data first so that you aren't using information from the test data when you, for example, scale a variable in the data. This is basically doing some fitting before you split the data.

## Random sample

n\_titanic\_all <- nrow(titanic\_all)  
prop\_train <- 0.7  
n\_train <- floor(prop\_train \* n\_titanic\_all)  
  
# Sample RowIDs  
rows\_train\_s <- sample.int(n\_titanic\_all, n\_train)  
  
titanic\_train\_s <- titanic\_all[rows\_train\_s, ]  
titanic\_test\_s <- titanic\_all[-rows\_train\_s, ]

### Class Distribution

pval<-t.test(titanic\_train\_s$survived, titanic\_test\_s$survived, conf.level = .95)$p.value   
pval

## [1] 0.2063763

t.test(titanic\_train\_s$survived, titanic\_test\_s$survived, conf.level = .95)$p.value >= .05

## [1] TRUE

#### How often would they be sufficiently similar?

You ideally want the class (Yes/No) distribution to be similar in training and testing data.

pass\_ttest<-function(x, prop\_train = .07){  
 n\_x\_all<-length(x)  
 n\_train<-floor(prop\_train \* n\_x\_all)  
   
 # Sample RowIDs  
 rows\_train<-sample.int(n\_x\_all, n\_train)  
   
 t.test(x[rows\_train], x[-rows\_train], conf.level = .95)$p.value >= .05  
}  
  
sum(replicate(1000,pass\_ttest(titanic\_all$survived)))/1000

## [1] 0.94

#### How do we get the same sample every time?

For reproducibility, you need to set the seed, then check that your sample satisfies the similarity criteria (from the same distribution)

pass\_ttest <- function(x, prop\_train = .07) {  
 set.seed(6767)  
 n\_x\_all <- length(x)  
 n\_train <- floor(prop\_train \* n\_x\_all)  
   
 # Sample RowIDs  
 rows\_train <- sample.int(n\_x\_all, n\_train)  
   
 t.test(x[rows\_train], x[-rows\_train], conf.level = .95)$p.value >= .05  
}  
  
sum(replicate(1000, pass\_ttest(titanic\_all$survived))) / 1000

## [1] 1

## Maintaining class ratio

Using caret package: helps with data selection, feature prep and other useful functions.

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

rows\_train\_c <-  
createDataPartition(titanic\_all$survived, p = prop\_train, list = FALSE)  
titanic\_train\_c <- titanic\_all[rows\_train\_c, ]  
titanic\_test\_c <- titanic\_all[-rows\_train\_c, ]

### Testing the class ratio

pass\_ttest<-function(x, prop\_train = .07){  
   
 # Sample RowIDs  
 rows\_train<-caret::createDataPartition(x, p=prop\_train, list=FALSE)  
   
 t.test(x[rows\_train], x[-rows\_train], conf.level = .95)$p.value >= .05  
}  
  
sum(replicate(1000,pass\_ttest(titanic\_all$survived)))/1000

## [1] 0.944

This gives us fewer cases where the class ratio is significantly different between training and sample datasets. Why is it not substantially much higher? Only so many permutations of 1309 rows for sampling - better results over more data.

## Reproducible sampling

We saw that we can set the seed within in a function to always yield the same value from our sampling process. We need to do this to select our training data so that the results don't shift each time we run the analysis.

## Defensive sampling

### Final sample

library(caret)  
set.seed(8787)  
rows\_train\_c <-  
createDataPartition(titanic\_all$survived, p = prop\_train, list = FALSE)  
titanic\_train\_c <- titanic\_all[rows\_train\_c, ]  
titanic\_test\_c <- titanic\_all[-rows\_train\_c, ]

### Storing a copy (one-off)

cache\_file<-"../data-processed/sampleIDs.Rdata"  
if(!file.exists(cache\_file)){  
 sample\_cache<- rows\_train\_c  
 save(sample\_cache,file=cache\_file)  
 rm(sample\_cache)  
}

### Check for changes

load(cache\_file)  
if(!identical(rows\_train\_c, sample\_cache)) stop("Hey, the sample has changed, you should check that out!")