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)

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

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.

## 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] 0.6988223

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?

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] 0.95

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

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

library(caret)  
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] 0.953

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!")

Now that we have our data split into two sections, we can start processing features.

## Our outcome variable

Some stats packages don't like booleans, others don't like factor labels with spaces. Let's process our outcome variable and put it at the end of the data.frame, as is also expected.

convertsurvived<-function(x) {  
 factor(x,levels = c("0","1")  
 ,labels=c("Died","Survived")  
 )}  
   
survived\_tr <- convertsurvived(titanic\_train\_c$survived)  
titanic\_train\_c$survived<-NULL  
titanic\_train\_c$survived<-survived\_tr

## Factors

In some cases with strings, if you want to make these a categorical variable, you need to make sure to incorporate levels from both sets of data.

stringcols<- sapply(titanic\_train\_c, is.character)  
lapply(titanic\_train\_c[,stringcols],function(x) head(unique(x)))

## $pclass  
## [1] "1st" "2nd" "3rd"  
##   
## $name  
## [1] "Allison, Master. Hudson Trevor" "Allison, Mrs. Hudson J C (Bessi"  
## [3] "Anderson, Mr. Harry" "Andrews, Miss. Kornelia Theodos"  
## [5] "Andrews, Mr. Thomas Jr" "Appleton, Mrs. Edward Dale (Cha"  
##   
## $sex  
## [1] "male" "female"  
##   
## $ticket  
## [1] "113781" "19952" "13502" "112050" "11769" "PC 17609"  
##   
## $cabin  
## [1] "C22 C26" "E12" "D7" "A36" "C101" ""   
##   
## $embarked  
## [1] "Southampton" "Cherbourg" "Queenstown"   
##   
## $boat  
## [1] "11" "" "3" "10" "D" "4"   
##   
## $home.dest  
## [1] "Montreal, PQ / Chesterville, ON" "New York, NY"   
## [3] "Hudson, NY" "Belfast, NI"   
## [5] "Bayside, Queens, NY" "Montevideo, Uruguay"

convertclass<-function(x){ ordered(x , labels=c("3rd","2nd","1st"), levels=c("3rd","2nd","1st"))}  
  
titanic\_train\_c$pclass <- convertclass(titanic\_train\_c$pclass )  
  
prop.table(table(titanic\_train\_c$pclass, titanic\_train\_c$survived),margin = 1)

##   
## Died Survived  
## 3rd 0.7449799 0.2550201  
## 2nd 0.5550000 0.4450000  
## 1st 0.3926941 0.6073059

convertsex<-function(x) factor(x, labels=c("male","female"))  
titanic\_train\_c$sex <- convertsex(titanic\_train\_c$sex)  
prop.table(table(titanic\_train\_c$sex, titanic\_train\_c$survived),1)

##   
## Died Survived  
## male 0.2647975 0.7352025  
## female 0.8104027 0.1895973

convertcabin<-function(x){factor(substr(x,1,1),levels=c("",LETTERS[1:7],"T")) }  
titanic\_train\_c$cabin<-convertcabin(titanic\_train\_c$cabin)  
prop.table(table(titanic\_train\_c$cabin, titanic\_train\_c$survived),1)

##   
## Died Survived  
## 0.6913408 0.3086592  
## A 0.5263158 0.4736842  
## B 0.3250000 0.6750000  
## C 0.4237288 0.5762712  
## D 0.3030303 0.6969697  
## E 0.2068966 0.7931034  
## F 0.3750000 0.6250000  
## G 0.5000000 0.5000000  
## T 1.0000000 0.0000000

convertembarked<-function(x){factor(x,levels = c("Southampton","Cherbourg" , "Queenstown" ))}  
titanic\_train\_c$embarked<-convertembarked(titanic\_train\_c$embarked)  
prop.table(table(titanic\_train\_c$embarked, titanic\_train\_c$survived),1)

##   
## Died Survived  
## Southampton 0.6600000 0.3400000  
## Cherbourg 0.4594595 0.5405405  
## Queenstown 0.6585366 0.3414634

titanic\_train\_c$name <- NULL  
titanic\_train\_c$ticket <- NULL  
  
# Note this could be different if we wanted to geocode and then further categorise this variable  
titanic\_train\_c$home.dest<-NULL  
  
# Boat appears to have too many values to be useful  
titanic\_train\_c$boat<-NULL

## Processing numbers

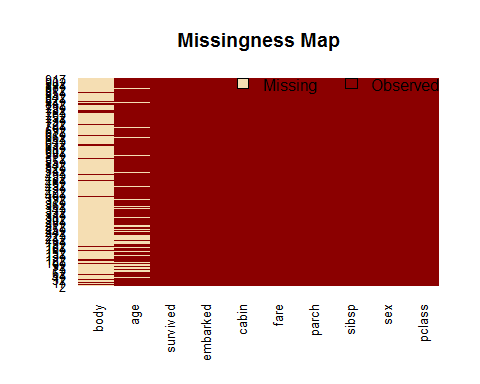
We should scale numbers so that values are on similar scales to prevent individual variables overpowering the model to a larger scale

procVar<-caret::preProcess(titanic\_train\_c)  
titanic\_train\_c<-predict(procVar, titanic\_train\_c)  
  
knitr::kable(summary(titanic\_train\_c))

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | pclass | sex | age | sibsp | parch | fare | cabin | embarked | body | survived |
|  | 3rd:498 | male :321 | Min. :-2.0306 | Min. :-0.4582 | Min. :-0.4755 | Min. :-0.61237 | :716 | Southampton:650 | Min. :-1.4801 | Died :568 |
|  | 2nd:200 | female:596 | 1st Qu.:-0.6184 | 1st Qu.:-0.4582 | 1st Qu.:-0.4755 | 1st Qu.:-0.46023 | C : 59 | Cherbourg :185 | 1st Qu.:-0.8966 | Survived:349 |
|  | 1st:219 | NA | Median :-0.1381 | Median :-0.4582 | Median :-0.4755 | Median :-0.34052 | B : 40 | Queenstown : 82 | Median :-0.0917 | NA |
|  | NA | NA | Mean : 0.0000 | Mean : 0.0000 | Mean : 0.0000 | Mean : 0.00000 | D : 33 | NA | Mean : 0.0000 | NA |
|  | NA | NA | 3rd Qu.: 0.6852 | 3rd Qu.: 0.5033 | 3rd Qu.:-0.4755 | 3rd Qu.:-0.04009 | E : 29 | NA | 3rd Qu.: 0.8541 | NA |
|  | NA | NA | Max. : 3.1552 | Max. : 7.2341 | Max. : 6.9573 | Max. : 9.25961 | A : 19 | NA | Max. : 1.7797 | NA |
|  | NA | NA | NA's :182 | NA | NA | NA | (Other): 21 | NA | NA's :836 | NA |

## Missings

library(Amelia)  
missmap(titanic\_train\_c)



Looking at the data dictionary (?titanic3), the column body represents a number associated with a retrieved dead body. That would of course be correlated with not surviving so let's remove it!

titanic\_train\_c$body<-NULL

## Applying to test samples

Once processed for our training, we should apply changes to our test data. Consolidating transformations like this allows us to think about our production requirements in relation to the data.

survived\_t <- convertsurvived(titanic\_test\_c$survived)  
titanic\_test\_c$survived<-NULL  
titanic\_test\_c$survived<-survived\_t  
  
titanic\_test\_c$pclass <- convertclass(titanic\_test\_c$pclass )  
titanic\_test\_c$sex <- convertsex(titanic\_test\_c$sex)  
titanic\_test\_c$cabin<-convertcabin(titanic\_test\_c$cabin)  
titanic\_test\_c$embarked<-convertembarked(titanic\_test\_c$embarked)  
titanic\_test\_c$name <- NULL  
titanic\_test\_c$home.dest <- NULL  
titanic\_test\_c$boat <- NULL  
titanic\_test\_c$ticket <- NULL  
titanic\_train\_c$body<-NULL  
  
titanic\_test\_c<-predict(procVar, titanic\_test\_c)