Titanic Survival Analysis

Langtao Chen

Update: Mar 2, 2021

Contents

In this example, we'll predict survival of passengers in Titanic by using validation set, k-fold cross-validation, and repeated k-fold cross-validation.

1. Dataset

We use the Titanic passenger survival data set in the titanic R package..

```r
# Clean the environment
rm(list = ls())

titanic <- read.csv('titanic_train.csv')
str(titanic)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

The dataset includes the following columns:

- PassengerId: Passenger ID

- Survived: Passenger Survival Indicator
- Pclass: Passenger Class
- Name: Name
- Sex: Sex
- Age: Age
- SibSp: Number of Siblings/Spouses Aboard
- Parch: Number of Parents/Children Aboard
- Ticket: Ticket Number
- Fare: Passenger Fare
- Cabin: Cabin
- Embarked: Port of Embarkation

```
head(titanic)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                  Name    Sex Age SibSp Parch
## 1                             Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                              Heikkinen, Miss. Laina female  26     0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                            Allen, Mr. William Henry   male  35     0     0
## 6                                    Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

```
summary(titanic)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891        
##  Class :character   1st Qu.:  7.91   Class :character   Class :character  
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character  
##                     Mean   : 32.20                                        
##                     3rd Qu.: 31.00                                        
##                     Max.   :512.33                                        
##
```

From the summary statistics, we found that there are missing values. Let's select the key variables in the dataset and remove missing values from the dataset.

```
library(dplyr)
```

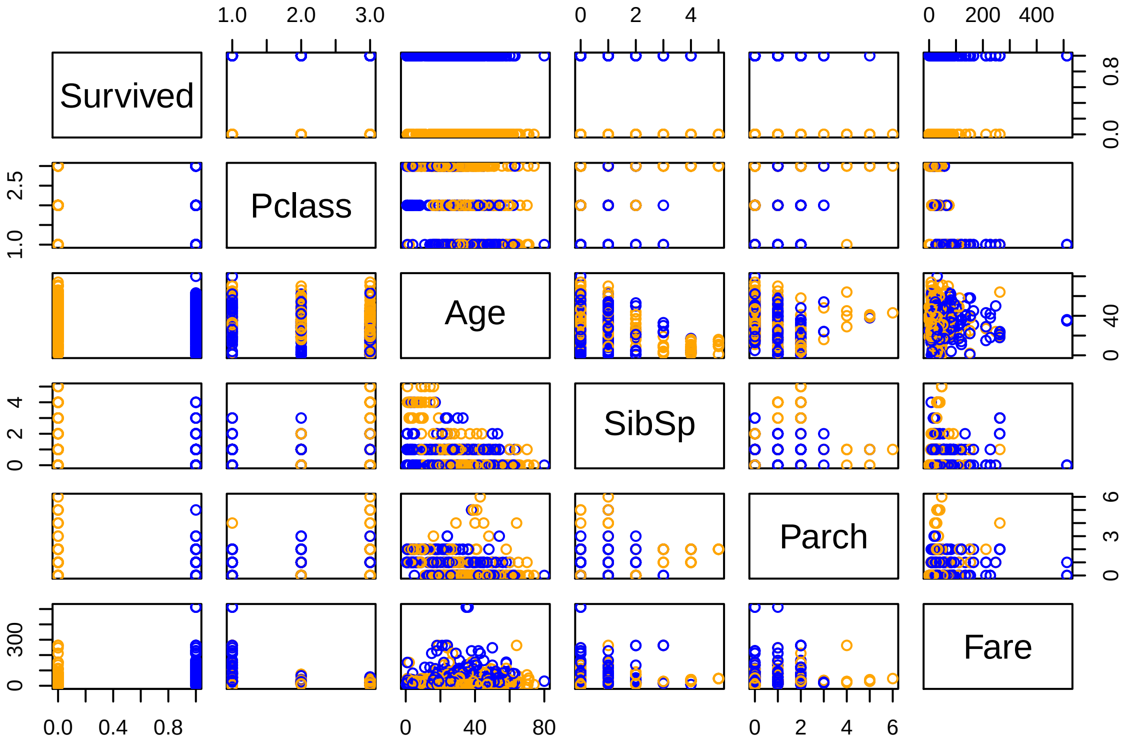
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
titanic <- titanic %>%
  select(Survived,Pclass,Sex,Age,SibSp,Parch,Fare) %>%
  na.omit

str(titanic)
```

```
## 'data.frame':    714 obs. of  7 variables:
##  $ Survived: int  0 1 1 1 0 0 0 1 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : chr  "male" "female" "female" "female" ...
##  $ Age     : num  22 38 26 35 35 54 2 27 14 4 ...
##  $ SibSp   : int  1 1 0 1 0 0 3 0 1 1 ...
##  $ Parch   : int  0 0 0 0 0 0 1 2 0 1 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  - attr(*, "na.action")= 'omit' Named int [1:177] 6 18 20 27 29 30 32 33 37 43 ...
##   ..- attr(*, "names")= chr [1:177] "6" "18" "20" "27" ...
```

Draw a scatterplot matrix.

```
pairs(~Survived + Pclass + Age + SibSp + Parch + Fare,
      data = titanic,
      col=ifelse(titanic$Survived==1, 'blue', 'orange'))
```



2. Validation Set Approach

We use a single 80/20% split.

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
set.seed(1234)

trainIndex <- createDataPartition(titanic$Survived, p = .8, list = FALSE)
train_data <- titanic[ trainIndex,]
test_data <- titanic[-trainIndex,]
```

```
# Fit a logistic regression model on the training dataset
logit_fit <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare,
                 family = binomial, data = train_data)
summary(logit_fit)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare, family = binomial, data = train_data)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max  
## -2.7975  -0.6247  -0.4026   0.6396   2.4422  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  5.372468   0.666469   8.061 7.56e-16 ***
## Pclass      -1.230464   0.180070  -6.833 8.30e-12 ***
## Sexmale     -2.569496   0.241454 -10.642  < 2e-16 ***
## Age         -0.045838   0.009154  -5.007 5.52e-07 ***
## SibSp       -0.330883   0.132982  -2.488   0.0128 *  
## Parch       -0.110692   0.136343  -0.812   0.4169    
## Fare         0.002605   0.002844   0.916   0.3598    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 770.89  on 571  degrees of freedom
## Residual deviance: 512.14  on 565  degrees of freedom
## AIC: 526.14
## 
## Number of Fisher Scoring iterations: 5
```

```
# Predict on the test dataset
pred_prob <- predict(object=logit_fit, newdata = test_data, type='response')
pred_class <- ifelse(pred_prob > 0.5, 1, 0)

confusionMatrix(factor(pred_class),factor(test_data$Survived), positive = "1")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 69 15
##          1 13 45
##                                           
##                Accuracy : 0.8028          
##                  95% CI : (0.7278, 0.8648)
##     No Information Rate : 0.5775          
##     P-Value [Acc > NIR] : 1.123e-08       
##                                           
##                   Kappa : 0.5941          
##                                           
##  Mcnemar's Test P-Value : 0.8501          
##                                           
##             Sensitivity : 0.7500          
##             Specificity : 0.8415          
##          Pos Pred Value : 0.7759          
##          Neg Pred Value : 0.8214          
##              Prevalence : 0.4225          
##          Detection Rate : 0.3169          
##    Detection Prevalence : 0.4085          
##       Balanced Accuracy : 0.7957          
##                                           
##        'Positive' Class : 1               
```

```
##
```

3. K-Fold Cross-Validation

3.1. A Simple Implementation Using caret Package

We can use the train() method in caret package to easily train a regression (prediction) or classification model using k-fold cross-validation. Refer to the following link for all available models supported by the train() method.

http://topepo.github.io/caret/available-models.html

```
## Train a logistic regression model with 10-fold cross-validation
fitControl <- trainControl(method = "cv",number = 10)

set.seed(123)
logit_fit2 <- train(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare,
                    data = titanic,
                    trControl = fitControl,
                    method="glm", family=binomial(link='logit'))

print(logit_fit2)
```

```
## Generalized Linear Model
##
## 714 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 642, 642, 643, 643, 643, 643, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7969288  0.574362
```

```
confusionMatrix(logit_fit2)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    0    1
##          0 50.6 11.5
##          1  8.8 29.1
##
##  Accuracy (average) : 0.7969
```

As you can see from the above result, the train() method in caret package by default only supports two performance measures (the overall accuracy and kappa coefficient) for cross-validation classification. If we need to check other measures, we can directly implement the k-fold cross-validation.

3.2. Directly Implement K-Fold Cross-Validation

Let's manually implement k-fold cross-validation.

In this example, let's choose logistic regression as the predictive model, and balanced accuracy as the performance measure.

$Balanced\ Accuracy = \frac{Sensitivity+Specificity}{2}$

```
# Implement k-fold cross-validation
k.folds <- function(k) {
    folds <- createFolds(titanic$Survived, k = k, list = TRUE, returnTrain = TRUE)
    accuracies <- c()

    for (i in 1:k) {
        model <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare,
                     data = titanic[folds[[i]],],family=binomial(link='logit'))

        pred_prob_cv <- predict(object = model, newdata = titanic[-folds[[i]],], type = "response")
        pred_class_cv <- ifelse(pred_prob_cv > 0.5, 1, 0)

        accuracies <- c(accuracies,
                        confusionMatrix(factor(pred_class_cv),
                                        factor(titanic[-folds[[i]], ]$Survived), positive = "1")$byClas
    }

    accuracies
}
```

```
# Execute the k-fold cross-validation
set.seed(123)
accuracies_cv <- k.folds(5)
accuracies_cv
```

```
## Balanced Accuracy Balanced Accuracy Balanced Accuracy Balanced Accuracy
##         0.7596491         0.8343663         0.7404421         0.8017136
## Balanced Accuracy
##         0.7879455
```

```
# Calculate the average balanced accuracy
cat('Balanced Accuracy:\n Mean = ', mean(accuracies_cv),"; ",
    'Standard Deviation = ',sd(accuracies_cv), ";\n",
    '95% Confidence Interval = [',
    mean(accuracies_cv) - sd(accuracies_cv) * 1.96, ", ",
    mean(accuracies_cv) + sd(accuracies_cv) * 1.96,"]")
```

```
## Balanced Accuracy:
##  Mean =  0.7848233 ;  Standard Deviation =  0.03658199 ;
##  95% Confidence Interval = [ 0.7131226 ,  0.856524 ]
```

4. Repeated K-Fold Cross-Validation

The mean and standard estimates in k-fold cross-validation is not very robust. We can repeat the k-fold cross-validation mulitple times to get more robust estimates.

Repeated k-fold cross-validation is repeating k-fold cross-validation multiple times, with different folds split in each repetition.

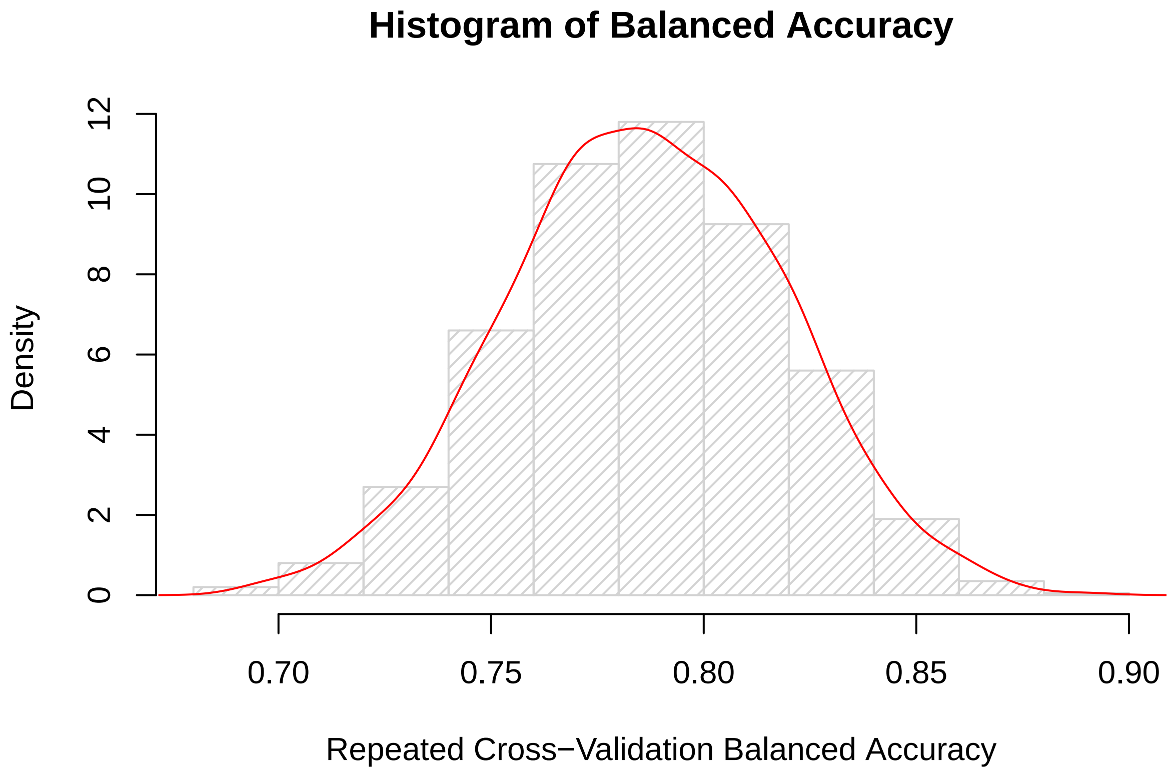
4.1. Directly Implement Repeated K-Fold Cross-Validation

```
# Execute the repeated k-fold cross-validation
set.seed(123)

v <- c()
v <- replicate(200, k.folds(5))

accuracies_rcv <- c()

for (i in 1 : 200) {
  accuracies_rcv <- c(accuracies_rcv, v[,i])
}

lci <- mean(accuracies_rcv) - sd(accuracies_rcv) * 1.96
uci <- mean(accuracies_rcv) + sd(accuracies_rcv) * 1.96

cat('Balanced Accuracy:\n Mean = ', mean(accuracies_rcv),"; ",
    'Standard Deviation = ',sd(accuracies_rcv), ";\n",
    '95% Confidence Interval = [',
    mean(accuracies_rcv) - sd(accuracies_rcv) * 1.96, ", ",
    mean(accuracies_rcv) + sd(accuracies_rcv) * 1.96,"]")
```

```
## Balanced Accuracy:
##  Mean =  0.7863668 ;  Standard Deviation =  0.03204394 ;
##  95% Confidence Interval = [ 0.7235607 ,  0.8491729 ]
```

Let's show the distribution of balanced accuracy in all repeated k-fold cross-validations.

```
hist(accuracies_rcv, prob = TRUE, density = 20,
     main = "Histogram of Balanced Accuracy",
     xlab = "Repeated Cross-Validation Balanced Accuracy")

lines(density(accuracies_rcv), col="red")
```



4.2. Use trainControl() to Configure Repeated CV

As mentioned above, the train() method in caret package by default only supports two performance measures (the overall accuracy and kappa coefficient) for cross-validation classification.

An alternative way is to set the summary function as twoClassSummary, which supports sensitivity, specificity, and ROC curve.

```
## Train a logistic regression model with repeated 5-fold cross-validation
fitControl_rcv <- trainControl(method = "repeatedcv",
                               number = 5,
                               repeats = 200,
                               classProbs = TRUE,
                               summaryFunction = twoClassSummary)

set.seed(123)
logit_fit_rcv <- train(factor(ifelse(Survived==1, 'Yes', 'No'), levels = c('Yes','No')) ~
                         Pclass + Sex + Age + SibSp + Parch + Fare,
                       data = titanic,
                       trControl = fitControl_rcv,
                       method="glm", family=binomial(link='logit'),
                       metric = "ROC")

print(logit_fit_rcv)
```

```
## Generalized Linear Model
##
```

```
## 714 samples
##   6 predictor
##   2 classes: 'Yes', 'No'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 200 times)
## Summary of sample sizes: 571, 571, 571, 572, 571, 571, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.8544638  0.7171552  0.8544238
```

```
confusionMatrix(logit_fit_rcv)
```

```
## Cross-Validated (5 fold, repeated 200 times) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction  Yes   No
##        Yes 29.1  8.6
##        No  11.5 50.7
##
##  Accuracy (average) : 0.7987
```

```
cat('Balanced Accuracy = ',
    sum(logit_fit_rcv$results['Spec'],logit_fit_rcv$results['Sens'])/2)
```

```
## Balanced Accuracy =  0.7857895
```

You can find the caret train result is very similar to the result of the direct implementation.