Logistic regression model with Titanic dataset

Rungrot Watthanakitkuson

2024-08-03

1. Load packages

```
library(conflicted)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
library(lattice)
library(caret)
```

2. Read titanic.csv into RStudio

```
titanic <- tibble(read_csv("titanic.csv"))
```

```
## Rows: 1310 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): name, sex, ticket, cabin, embarked, boat, home.dest
## dbl (7): pclass, survived, age, sibsp, parch, fare, body
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

3. Data Preparation

3.1 Remove body column

```
titanic$body <- NULL
```

3.2 Add PassengerId column

```
titanic <- tibble::rowid_to_column(titanic, "PassengerId")
```

3.3 Rename columns in titanic

```
titanic <- titanic %>%
  mutate(Pclass = pclass,
         Survived = survived,
         Name = name,
         Gender = sex,
         Age = age,
         SibSp = sibsp,
         Parch = parch,
         Ticket = ticket,
         Fare = fare,
         Cabin = cabin,
         Embarked = embarked) %>%
  select(PassengerId, Pclass,
         Survived,
         Name,
         Gender,
         Age,
         SibSp,
         Parch,
         Ticket,
         Fare,
         Cabin,
         Embarked)
```

3.4 Change data in Gender column

```
titanic$Gender <- factor(titanic$Gender,
                         levels = c("male","female"))

titanic$Survived <- factor(titanic$Survived,
                           levels = c(1,0),
                           labels = c("Yes","No"))
```

3.5 Build checking missing values function for each dataset

```
have_missing_values <- function(data) {
  ncol = 1
  while(ncol <= ncol(data)) {
    if (sum(is.na(data[ncol])) > 0) {
      print(colnames(data[ncol]))
      ncol <- ncol + 1
    } else {
      ncol <- ncol + 1
    }
  }
}
```

3.6 Check missing values in titanic

```
have_missing_values(titanic)
```

```
## [1] "Pclass"
## [1] "Survived"
## [1] "Name"
## [1] "Gender"
## [1] "Age"
## [1] "SibSp"
## [1] "Parch"
## [1] "Ticket"
## [1] "Fare"
## [1] "Cabin"
## [1] "Embarked"
```

3.7 Clean missing values

```
clean_titanic <- titanic %>%
                  drop_na()
```

4. Split Data

4.1 Build train-test data spliting function

```
train_test_split <- function(data) {
  set.seed(0)
  n <- nrow(data)
  id <- sample(n, size = 0.8*n)
  train_data <- data[id, ]
  test_data <- data[-id, ]
  return(list(train_data, test_data))
}
```

```
split_data <- train_test_split(clean_titanic)
```

5. Train Data

```
glm_model <- train(Survived ~ Gender + Age,
                   data = split_data[[1]],
                   method = "glm",
                   family = "binomial")
```

```
summary(glm_model)
```

```
##
## Call:
## NULL
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.00871    0.49721  -2.029  0.04248 *
## Genderfemale -2.98074    0.44607  -6.682 2.35e-11 ***
## Age           0.03316    0.01195   2.775  0.00553 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 267.52  on 215  degrees of freedom
## Residual deviance: 188.89  on 213  degrees of freedom
## AIC: 194.89
##
## Number of Fisher Scoring iterations: 5
```

6. Scoring Model

```r
probSurvived <- predict(glm_model, newdata = split_data[[2]])
```

7. Model Evaluation

```r
confusionMatrix(probSurvived,
                split_data[[2]]$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Yes No
##        Yes  25  6
##        No    6 17
##
##                Accuracy : 0.7778
##                  95% CI : (0.644, 0.8796)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 0.00143
##
##                   Kappa : 0.5456
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.8065
##             Specificity : 0.7391
##          Pos Pred Value : 0.8065
##          Neg Pred Value : 0.7391
##              Prevalence : 0.5741
##          Detection Rate : 0.4630
##    Detection Prevalence : 0.5741
##       Balanced Accuracy : 0.7728
##
##        'Positive' Class : Yes
##
```