Logistic_Regression_homework

Napat Teekasuk

2023-07-23

Logistic Regression

load packages

```
library(titanic)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

DROP NA (missing value)

```
titanic_train <- na.omit(titanic_train)
nrow(titanic_train)
```

```
## [1] 714
```

SPLIT DATA

```
set.seed(36)
n <- nrow(titanic_train)
id <- sample(1:n, size = n*0.7)
train_data <- titanic_train[id, ]
test_data <- titanic_train[-id, ]

nrow(train_data)
```

```
## [1] 499
```

```
nrow(test_data)
```

```
## [1] 215
```

Train Model

```r
train_model <- glm(Survived ~ Age + Pclass + Sex,
                   data = train_data,
                   family = "binomial")

summary(train_model)
```

```
## 
## Call:
## glm(formula = Survived ~ Age + Pclass + Sex, family = "binomial", 
##     data = train_data)
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  5.42395    0.62322   8.703  < 2e-16 ***
## Age         -0.04059    0.00938  -4.327 1.51e-05 ***
## Pclass      -1.39169    0.17123  -8.128 4.38e-16 ***
## Sexmale     -2.60347    0.25535 -10.196  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 675.80  on 498  degrees of freedom
## Residual deviance: 437.29  on 495  degrees of freedom
## AIC: 445.29
## 
## Number of Fisher Scoring iterations: 5
```

```r
train_data$prob <- predict(train_model, type = "response")
train_data$pred <- ifelse(train_data$prob >= 0.5, 1, 0)
```

Test Model

```r
test_data$prob <- predict(train_model, newdata = test_data, type = "response")
test_data$pred <- ifelse(test_data$prob >= 0.5, 1, 0)
```

confusion matrix

```r
conM <- table(train_data$pred, train_data$Survived, dnn = c("Predicted","Actual") )
train_Accuracy <- (conM[1,1] + conM[2,2]) / sum(conM)
train_Percision <- conM[2,2] / (conM[2,1] + conM[2,2])
train_Recall <- conM[2,2] / (conM[1,2] + conM[2,2])
train_F1score <- 2*((train_Percision*train_Recall) / (train_Percision+train_Recall))

cat("Train Data",
    "\nAccuracy:", train_Accuracy,
    "\nPercision:",train_Percision,
    "\nRecall:",train_Recall,
    "\nF1score:", train_F1score)
```

```
## Train Data 
## Accuracy: 0.8076152
## Percision: 0.7823834
## Recall: 0.7365854
## F1score: 0.758794
```

```
conM2 <- table(test_data$pred, test_data$Survived, dnn = c("Predicted","Actual") )
test_Accuracy <- (conM2[1,1] + conM2[2,2]) / sum(conM2)
test_Percision <- conM2[2,2] / (conM2[2,1] + conM2[2,2])
test_Recall <- conM2[2,2] / (conM2[1,2] + conM2[2,2])
test_F1score <- 2*((test_Percision*test_Recall) / (test_Percision+test_Recall))

cat("Test Data",
    "\nAccuracy:", test_Accuracy,
    "\nPercision:",test_Percision,
    "\nRecall:",test_Recall,
    "\nF1score:", test_F1score)
```

```
## Test Data
## Accuracy: 0.7813953
## Percision: 0.7261905
## Recall: 0.7176471
## F1score: 0.7218935
```