Logistic Regression Bootcamp Homework

Nirucha P

2024-01-03

Assignment

1. Use the titanic dataset to create Logistic Regression Model.
2. This model is used to predict the probability of survival of people in Titanic boats.
3. Export this document to pdf with R Markdown.

Install packaged

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
```

```
library(titanic)
```

```
## Warning: package 'titanic' was built under R version 4.3.2
```

Load raw data

```
data("titanic_train")
glimpse(titanic_train)
```

```
## Rows: 891
## Columns: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1~
```

```
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3, 3~
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl~
## $ Sex         <chr> "male", "female", "female", "female", "male", "male", "mal~
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 14, ~
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1, 0~
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0~
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37~
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.8625,~
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", "G6", "C~
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", "S", "S"~
```

Table 1: Explain variables in titanic dataset

Variable	Definition	Key
PassengerId	The unique number of passengers	
Survived	The probability of survival	0 = No, 1 = Yes
Pclass	Ticket class	1 = Upper, 2 = Middle, 3 = Lower
Name	The fullname of passengers	
Sex	Gender	
Age	Age in years	
SibSp	Number of siblings / spouses aboard the Titanic	
Parch	Number of parents / children aboard the Titanic	
Ticket	Ticket number	
Fare	Passenger fare	
Cabin	Cabin number	
Embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Data Cleaning

1. Check some missing values in the diamonds dataset.

```
if(sum(is.na(titanic_train)) > 0){
  print("This dataset has some missing values.")

} else{
  print("This dataset doesn't have any missing values.")
}
```

```
## [1] "This dataset has some missing values."
```

2. Drop NA (missing values).

```
titanic_train <- na.omit(titanic_train)
cat("Number of rows after cleaned :",nrow(titanic_train))
```

```
## Number of rows after cleaned : 714
```

Prepare Data

1. Change column Sex from string to factor.

```
titanic_train$Sex  <- factor(titanic_train$Sex,
                             level = c("male", "female"),
                             label = c(0, 1))
glimpse(titanic_train)
```

```
## Rows: 714
## Columns: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19~
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1~
## $ Pclass      <int> 3, 1, 3, 1, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 3, 2, 2, 3~
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl~
## $ Sex         <fct> 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1~
## $ Age         <dbl> 22, 38, 26, 35, 35, 54, 2, 27, 14, 4, 58, 20, 39, 14, 55, ~
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 1, 0, 0, 0~
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0, 0~
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37~
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 51.8625, 21.0750~
## $ Cabin       <chr> "", "C85", "", "C123", "", "E46", "", "", "", "G6", "C103"~
## $ Embarked    <chr> "S", "C", "S", "S", "S", "S", "S", "S", "C", "S", "S", "S"~
```

2. Split the data into two parts with a random sampling method. We use 70% for the training set and 30% for the testing set.

```
set.seed(95)
n <- nrow(titanic_train)
id <- sample(1:n, size = n*0.7)
train_data <- titanic_train[id, ]
test_data <- titanic_train[-id, ]

cat("The training set has",nrow(train_data),", and the testing set has",nrow(test_data), "rows.")
```

```
## The training set has 499 , and the testing set has 215 rows.
```

Create Train and Test Model

We use the Pclass, Age, and Sex columns to predict the probability of survival (Survived Column).

Train Model

```
# Calculate the line of best fit a logistic regression
logit_model <- glm(Survived ~ Pclass + Age + Sex,
                   data = train_data,
                   family = "binomial")

# Predict with the training set
train_data$prob_survived <- predict(logit_model, type = "response")
```

```r
# Cut off at 0.5 of probability
train_data$pred_survived <- ifelse(train_data$prob_survived >= 0.5, 1, 0)

train_data %>%
  select(Pclass, Age, Sex, pred_survived) %>%
  head(5)
```

```
##     Pclass Age Sex pred_survived
## 513      1  36   0             0
## 660      1  58   0             0
## 371      1  25   0             1
## 259      1  35   1             1
## 104      3  33   0             0
```

Test Model

```r
# Predict with the testing set
test_data$prob_survived <- predict(logit_model, newdata = test_data, type = "response")

# Cut off at 0.5 of probability
test_data$pred_survived <- ifelse(test_data$prob_survived >= 0.5, 1, 0)

test_data %>%
  select(Pclass, Age, Sex, pred_survived) %>%
  head(5)
```

```
##    Pclass Age Sex pred_survived
## 1       3  22   0             0
## 3       3  26   1             1
## 5       3  35   0             0
## 10      2  14   1             1
## 11      3   4   1             1
```

Model Evaluation

Calculate the average accuracy of the train and test models to determine whether the generated model is overfitting or not.

```r
# Train Model
avg_acc_train <- train_data$Survived == train_data$pred_survived
cat("Average accuracy of the train model :", mean(avg_acc_train))
```

```
## Average accuracy of the train model : 0.7815631
```

```r
# Test Model
avg_acc_test <- test_data$Survived == test_data$pred_survived
cat("Average accuracy of the test model :", mean(avg_acc_test))
```

```
## Average accuracy of the test model : 0.8093023
```

We observed that the average accuracy values of the train and test models are close to each other. Conclude that this logistic regression model does not overfit.

Confusion Matrix

Train set metrics calculation

```
conM_train <- table(train_data$pred_survived, train_data$Survived,
              dnn = c("Predicted", "Actual"))

acc_train <- (conM_train[1, 1] + conM_train[2, 2]) / sum(conM_train)
prec_train <- conM_train[2, 2] / (conM_train[2, 1] + conM_train[2, 2])
rec_train <- conM_train[2, 2] / (conM_train[1, 2] + conM_train[2, 2])
f1_train <- 2*((prec_train * rec_train) / (prec_train + rec_train))
```

Test set metrics calculation

```
conM_test <- table(test_data$pred_survived, test_data$Survived,
              dnn = c("Predicted", "Actual"))

acc_test <- (conM_test[1, 1] + conM_test[2, 2]) / sum(conM_test)
prec_test <- conM_test[2, 2] / (conM_test[2, 1] + conM_test[2, 2])
rec_test <- conM_test[2, 2] / (conM_test[1, 2] + conM_test[2, 2])
f1_test <- 2*((prec_test * rec_test) / (prec_test + rec_test))
```

Table 2: Confusion Matrix comparison

	Accuracy	Precision	Recall	F1.score
Train set	0.7815631	0.7365591	0.6954315	0.7154047
Test set	0.8093023	0.8023256	0.7419355	0.7709497