Logistic Regression

SRIRAM VIVEK

2025-02-26

Cleaning the data

```r
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 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
titanic_data <- read.csv('/Users/sriram/Desktop/SEMESTER 2/AMS 580/Logistic Regression/Titanic2.csv')


titanic_cleaned <- titanic_data %>%
  select(-PassengerId, -Name, -Ticket, -Cabin)

titanic_cleaned <- titanic_cleaned %>%
  filter(!is.na(Age))

n_passengers <- nrow(titanic_cleaned)
cat("Number of passengers after cleaning:", n_passengers, "\n")
```

```
## Number of passengers after cleaning: 714
```

```r
titanic_cleaned <- titanic_cleaned %>%
  mutate(Survived = as.factor(Survived),
         Pclass = as.factor(Pclass))

str(titanic_cleaned)
```

```
## 'data.frame':    714 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 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 ...
##  $ Embarked: chr  "S" "C" "S" "S" ...
```

Using the random seed 123 to divide the cleaned data into 80% training and 20% testing.

```r
set.seed(123)
training_samples <- createDataPartition(titanic_cleaned$Survived, p = 0.8, list = FALSE)
train_data <- titanic_cleaned[training_samples, ]
test_data <- titanic_cleaned[-training_samples, ]
```

Fitting a logistic regression model with all the other 7 predictors using the training data.

```r
model <- glm(Survived ~ ., data = train_data, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.128749   0.604277   6.833 8.34e-12 ***
## Pclass2     -1.012074   0.370945  -2.728 0.006365 **
## Pclass3     -2.076496   0.384989  -5.394 6.90e-08 ***
## Sexmale     -2.556944   0.242699 -10.535  < 2e-16 ***
## Age         -0.031944   0.009200  -3.472 0.000516 ***
## SibSp       -0.350734   0.144212  -2.432 0.015012 *
## Parch       -0.115459   0.139457  -0.828 0.407717
## Fare         0.002723   0.003202   0.851 0.395005
## EmbarkedQ   -1.080480   0.641079  -1.685 0.091910 .
## EmbarkedS   -0.718802   0.320819  -2.241 0.025057 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 772.45  on 571  degrees of freedom
## Residual deviance: 518.19  on 562  degrees of freedom
## AIC: 538.19
```

```
##
## Number of Fisher Scoring iterations: 5
```

Predicting the response variable "Survived" (whether the subject survived or not) for the testing data based on the fitted model followed by generating a confusion matrix, reporting overall accuracy, sensitivity and specificity.

```
probabilities <- predict(model, test_data, type = "response")
predicted_classes <- ifelse(probabilities > 0.5, 1, 0)

confusion_matrix <- table(Predicted = predicted_classes, Actual = test_data$Survived)
print(confusion_matrix)
```

```
##          Actual
## Predicted  0  1
##         0 71 15
##         1 13 43
```

```
accuracy <- mean(predicted_classes == test_data$Survived)
sensitivity <- sum(predicted_classes == 1 & test_data$Survived == 1) / sum(test_data$Survived == 1)
specificity <- sum(predicted_classes == 0 & test_data$Survived == 0) / sum(test_data$Survived == 0)

cat("Overall Accuracy:", accuracy, "\n")
```

```
## Overall Accuracy: 0.8028169
```

```
cat("Sensitivity:", sensitivity, "\n")
```

```
## Sensitivity: 0.7413793
```

```
cat("Specificity:", specificity, "\n")
```

```
## Specificity: 0.8452381
```

Testing the above model to predict the survival of additional passengers.

```
additional_passengers <- data.frame(
  Pclass = as.factor(c(3, 1, 2)),
  Sex = c("male", "female", "male"),
  Age = c(24, 68, 41),
  SibSp = c(1, 0, 1),
  Parch = c(0, 0, 2),
  Fare = c(8.42, 24.34, 41.93),
  Embarked = c("Q", "C", "S")
)

additional_probabilities <- predict(model, additional_passengers, type = "response")
additional_predicted_classes <- ifelse(additional_probabilities > 0.5, 1, 0)

cat("Predicted survival for additional passengers:", additional_predicted_classes, "\n")
```

```
## Predicted survival for additional passengers: 0 1 0
```

```r
cat("Passenger 892: ", "Not Survived\n")
```

```
## Passenger 892:  Not Survived
```

```r
cat("Passenger 893: ", "Survived\n")
```

```
## Passenger 893:  Survived
```

```r
cat("Passenger 894: ", "Not Survived")
```

```
## Passenger 894:  Not Survived
```