

Linear Regression

Chanchawat Pakdeesri

2025-05-19

Logistic Regression

(classification problem) method = “glm”

```
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
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
```

```
## Attaching package: 'caret'
```

```
##
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
## lift
```

load data

```
library(mlbench)
```

```
data("PimaIndiansDiabetes")
```

```
df <- PimaIndiansDiabetes
```

```
df %>%
```

```
  select(age, diabetes) %>%
```

```
  group_by(diabetes) %>%
```

```
  summarise(avg_age = mean(age, na.rm=TRUE),
            median_age = median(age))
```

```
## # A tibble: 2 x 3
```

```
##   diabetes avg_age median_age
```

```
##   <fct>      <dbl>      <dbl>
```

```
## 1 neg          31.2         27
```

```
## 2 pos          37.1          36
## check / inspect data
sum(complete.cases(df))

## [1] 768
nrow(df)

## [1] 768
glimpse(df)

## Rows: 768
## Columns: 9
## $ pregnant <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, 5, 7, 0, 7, 1, 1-
## $ glucose <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, 125, 110, 168, 139, ~
## $ pressure <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92, 74, 80, 60, 72, 0, ~
## $ triceps <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, 0, 23, 19, 0, 47, 0-
## $ insulin <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, 0, 846, 175, 0, 230-
## $ mass <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35.3, 30.5, 0.0, 37-
## $ pedigree <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.248, 0.134, 0.158-
## $ age <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, 34, 57, 59, 51, 3-
## $ diabetes <fct> pos, neg, pos, neg, pos, neg, pos, neg, pos, pos, neg, pos, n-
head(df)

##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1         6      148       72      35         0 33.6    0.627  50      pos
## 2         1       85       66      29         0 26.6    0.351  31      neg
## 3         8      183       64       0         0 23.3    0.672  32      pos
## 4         1       89       66      23        94 28.1    0.167  21      neg
## 5         0      137       40      35       168 43.1    2.288  33      pos
## 6         5      116       74       0         0 25.6    0.201  30      neg
tail(df)

##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 763         9       89       62       0         0 22.5    0.142  33      neg
## 764        10      101       76      48       180 32.9    0.171  63      neg
## 765         2      122       70      27         0 36.8    0.340  27      neg
## 766         5      121       72      23       112 26.2    0.245  30      neg
## 767         1      126       60       0         0 30.1    0.349  47      pos
## 768         1       93       70      31         0 30.4    0.315  23      neg
```

split data

80:20

```
set.seed(42)
n <- nrow(df)
id<- sample(1:n,0.8*n)
train_df <- df[id,]
test_df <- df[-id,]## check train_data %>% head
```

train a logistic regression model

```
set.seed(42)

train_ctrl <- trainControl(method = "cv",
                           number = 5)
logit_model <- train(diabetes ~ age + glucose + pressure,
                    data = train_df,
                    method = "glm")
```

score

```
p_test <- predict(logit_model,
                  newdata = test_df)
```

evaluate

```
acc <- mean(test_df$diabetes == p_test)
conf_matrix <- confusionMatrix(p_test, test_df$diabetes,
                              positive = "pos", # default : NULL
                              dnn = c("Prediction", "Reference"))

accuracy <- conf_matrix$overall["Accuracy"]
sensitivity <- conf_matrix$byClass["Sensitivity"]
specificity <- conf_matrix$byClass["Specificity"]
f1_malignant <- conf_matrix$byClass["F1"][1] # F1 for "malignant"
f1_benign <- conf_matrix$byClass["F1"][2] # F1 for "benign"
precision_malignant <- conf_matrix$byClass["Precision"][1]
precision_benign <- conf_matrix$byClass["Precision"][2]
recall_malignant <- conf_matrix$byClass["Recall"][1]
recall_benign <- conf_matrix$byClass["Recall"][2]

# Print selected metrics
cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.7597403
cat("Sensitivity:", sensitivity, "\n")

## Sensitivity: 0.5454545
cat("Specificity:", specificity, "\n")

## Specificity: 0.8787879
cat("F1 Score (Malignant):", f1_malignant, "\n")

## F1 Score (Malignant): 0.6185567
cat("F1 Score (Benign):", f1_benign, "\n")

## F1 Score (Benign): NA
cat("Precision (Malignant):", precision_malignant, "\n")

## Precision (Malignant): 0.7142857
```

```
cat("Precision (Benign):", precision_benign, "\n")
```

```
## Precision (Benign): NA
```

```
cat("Recall (Malignant):", recall_malignant, "\n")
```

```
## Recall (Malignant): 0.5454545
```

```
cat("Recall (Benign):", recall_benign, "\n")
```

```
## Recall (Benign): NA
```