

R Programming

Sun Wasan

Machine Learning

Logistic Regression model

Predict diabetes by

\$ pregnant
\$ glucose
\$ pressure
\$ triceps
\$ insulin
\$ mass
\$ pedigree
\$ age

Data Overview

Hide

	pregnant <dbl>	glucose <dbl>	pressure <dbl>	triceps <dbl>	insulin <dbl>	mass <dbl>	pedigree <dbl>	age <dbl>	diabetes <fctr>
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

6 rows

Hide

```
library(mlbench)

data("PimaIndiansDiabetes")
print(head(PimaIndiansDiabetes))

library(tidyverse)
library(caret)
library(tidyr)

df <- PimaIndiansDiabetes

#train test split
train_test_split <- function(data , size = 0.8){
  n <- nrow(data)

  id <- sample(1:n , size = n*size)
  train_data <- data[id, ]
  test_data <- data[-id, ]

  return(list(train_data,test_data))
}

data_split <- train_test_split(df)
train_data <- data_split[[1]]
test_data <- data_split[[2]]

#train model
logis <- train(diabetes ~ . ,
               method = 'glm',
               data = train_data)
res <- predict(logis , test_data)

cm <- confusionMatrix(res,test_data$diabetes)

precision <- cm$byClass['Pos Pred Value']
recall <- cm$byClass['Sensitivity']
f_measure <- 2 * ((precision * recall) / (precision + recall))
print(cm)
```

Confusion Matrix and Statistics

		Reference	
Prediction	neg	pos	
neg	89	16	
pos	11	38	
Accuracy : 0.8247			
95% CI : (0.7553, 0.8812)			
No Information Rate : 0.6494			
P-Value [Acc > NIR] : 1.198e-06			
Kappa : 0.6066			
Mcnemar's Test P-Value : 0.4414			
Sensitivity : 0.8900			
Specificity : 0.7037			
Pos Pred Value : 0.8476			
Neg Pred Value : 0.7755			
Prevalence : 0.6494			
Detection Rate : 0.5779			
Detection Prevalence : 0.6818			
Balanced Accuracy : 0.7969			
'Positive' Class : neg			

Hide

```
paste("F1-Score :",f_measure)

[1] "F1-Score : 0.868292682926829"
```

Visualization by ggplot2

Hide

```
library(ggplot2)
library(tidyverse)
library(patchwork)
library(ggcorrplot)

dummy <- model.matrix( ~., df)[-1]

corr <- round(cor(dummy), 2)
p1 <- ggcorrplot(corr)
p2 <- ggplot(data.frame(dummy)) , aes(glucose , diabetespos))+
  geom_point()+
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE)

p3 <- ggplot(data.frame(dummy)) , aes(mass , diabetespos))+
  geom_point()+
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"), se = FALSE)

p4 <- ggplot(data.frame(dummy)) , aes(pregnant , diabetespos))+
  geom_point()+
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE)

p5 <- ggplot(data.frame(dummy)) , aes(age , diabetespos))+
  geom_point()+
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE)

print(p1 | (p2+p3)/(p4+p5))
```

