Heart Attack Analysis & Prediction Dataset

Berkay ÇAYAN

Heart Attack Analysis & Prediction Dataset

```r
# Packages

library(tidyverse)
library(caret)
library(DALEX)
library(ROCR)
```

Problem

The problem is to predict the occurrence of a heart attack based on certain attributes. This is a classification problem where we need to predict whether a person is likely to have a heart attack or not.

```r
# Dataset
data <- read.csv("/cloud/project/heart.csv")

str(data)
```

```
'data.frame':   303 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
 $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
 $ trtbps  : int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
```

```
$ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
$ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
$ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
$ exng    : int  0 0 0 0 1 0 0 0 0 0 ...
$ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
$ slp     : int  0 0 2 2 2 1 1 2 2 2 ...
$ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
$ thall   : int  1 2 2 2 2 1 2 3 3 2 ...
$ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Features

The dataset contains 14 attributes, which are as follows:

1. -age: age of the patient (numerical)
2. sex: gender of the patient (categorical - 1: male, 0: female)
3. cp: chest pain type (categorical - 0: typical angina, 1: atypical angina, 2: non-anginal pain, 3: asymptomatic)
4. trtbps: resting blood pressure (in mm Hg) (numerical)
5. chol: cholesterol level (in mg/dl) (numerical)
6. fbs: fasting blood sugar > 120 mg/dl (categorical - 1: true, 0: false)
7. restecg: resting electrocardiographic results (categorical - 0: normal, 1: having ST-T wave abnormality, 2: showing probable or definite left ventricular hypertrophy)
8. thalachh: maximum heart rate achieved (numerical)
9. exng: exercise induced angina (categorical - 1: yes, 0: no)
10. oldpeak: ST depression induced by exercise relative to rest (numerical)
11. slp: slope of the peak exercise ST segment (categorical - 0: upsloping, 1: flat, 2: downsloping)
12. caa: number of major vessels (0-3) colored by flourosopy (numerical)
13. thall: thal rate (categorical - 1: normal, 2: fixed defect, 3: reversible defect)
14. output: target variable - 0: less chance of heart attack, 1: more chance of heart attack (categorical)

Logistic Regression Model

```r
# Create training and test datasets
set.seed(123)
index <- sample(1 : nrow(data), round(nrow(data) * 0.80))
train <- data[index, ]
test  <- data[-index, ]
```

```r
# Create logistic regression model
lr_model <- glm(output ~ ., data = train, family = "binomial")
summary(lr_model)
```

```
Call:
glm(formula = output ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6127  -0.4311   0.1795   0.5835   2.4205

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.555347   2.794683   1.630  0.10310
age         -0.013283   0.025289  -0.525  0.59941
sex         -1.540661   0.509861  -3.022  0.00251 **
cp           0.769378   0.195309   3.939 8.17e-05 ***
trtbps      -0.017916   0.011003  -1.628  0.10348
chol        -0.003182   0.004177  -0.762  0.44617
fbs          0.132748   0.566248   0.234  0.81465
restecg      0.658516   0.389009   1.693  0.09049 .
thalachh     0.019761   0.011439   1.727  0.08408 .
exng        -1.077694   0.448609  -2.402  0.01629 *
oldpeak     -0.678520   0.240961  -2.816  0.00486 **
slp          0.263729   0.411859   0.640  0.52195
caa         -0.646435   0.201346  -3.211  0.00132 **
thall       -1.015510   0.330194  -3.075  0.00210 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 333.48  on 241  degrees of freedom
Residual deviance: 173.07  on 228  degrees of freedom
AIC: 201.07

Number of Fisher Scoring iterations: 6
```

```r
predicted_probs <- predict(lr_model, test[,-14], type = "response")
head(predicted_probs)
```

```
        2         3        12        15        19        28 
0.7079596 0.9397678 0.9823383 0.9748455 0.6193038 0.9260086 
```

```r
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)
head(predicted_classes)
```

```
 2  3 12 15 19 28 
 1  1  1  1  1  1 
```

```r
table(train$output) / dim(train)
```

```
        0         1 
0.4545455 9.4285714 
```

```r
conf_matrix <- table(Predicted = predicted_classes, Actual = test$output)
```

```r
TP <- sum(predicted_classes[which(test$output == "1")] == 1)
FP <- sum(predicted_classes[which(test$output == "1")] == 0)
TN <- sum(predicted_classes[which(test$output == "0")] == 0)
FN <- sum(predicted_classes[which(test$output == "0")] == 1)
recall      <- TP / (TP + FN)
specificity <- TN / (TN + FP)
precision   <- TP / (TP + FP)
accuracy    <- (TN + TP) / (TP + FP + TN + FN)
recall
```

```
[1] 0.7948718
```

```
specificity
```

```
[1] 0.9090909
```

```
precision
```

```
[1] 0.9393939
```

```
accuracy
```

```
[1] 0.8360656
```

This code shows the four metrics used to evaluate the performance of the classification model: Recall, Specificity, Precision, and Accuracy.

Recall represents the rate of true positives detected by the model. In this example, the Recall value is calculated as 0.79487. This means that the model correctly detected approximately 79% of the true positives.

Specificity represents the rate of true negatives detected by the model. In this example, the Specificity value is calculated as 0.90909. This means that the model correctly detected approximately 91% of the true negatives.

Precision represents the rate of true positives out of all the positive classifications made by the model. In this example, the Precision value is calculated as 0.93939. This means that approximately 94% of the positive classifications made by the model are indeed true positives.

Accuracy represents the rate of correct classifications out of all the examples. In this example, the Accuracy value is calculated as 0.83606. This means that the model was able to correctly classify approximately 83% of all examples.

Imbalance Problem

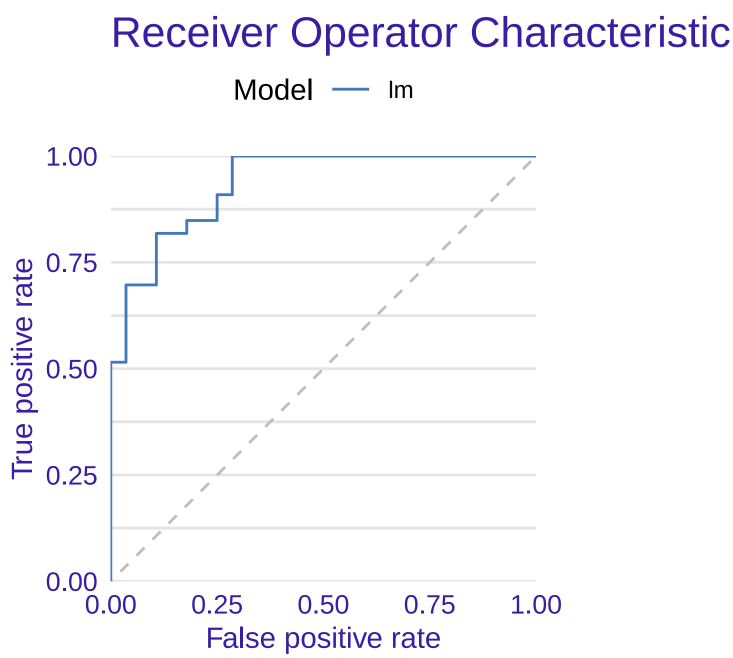
```
table(train$output) / dim(train)[1]
```

```
        0         1 
0.4545455 0.5454545 
```

The table looks pretty balanced, but not harmful to make under-overasmpling.

```r
explain_lr <- DALEX::explain(model   = lr_model,                # trained model
                             data    = test[, -14],           # test set without target
                             y       = test$output == "1", # observed values of target
                                                              # with reference class
                             type    = "classification",      # type of task
                             verbose = FALSE)                 # remove some messages
```

```r
performance_lr <- model_performance(explain_lr)
plot(performance_lr, geom = "roc")
```




```r
performance_lr
```

```
Measures for:  classification
recall     : 0.9393939
precision  : 0.7948718
f1         : 0.8611111
accuracy   : 0.8360656
auc        : 0.9339827
```

```
Residuals:
         0%         10%         20%         30%         40%         50% 
-0.92500426 -0.66440247 -0.23803316 -0.05420990 -0.00710291  0.01057154 
        60%         70%         80%         90%        100% 
 0.02908870  0.05916045  0.19665257  0.30022165  0.59491469 
```

- Accuracy is the proportion of correct predictions made by the model among all cases. The accuracy value for this model is 0.8360656, indicating that the model predicted about 84% of the cases correctly.
- AUC is the Area Under the Curve of the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate against the False Positive Rate. The AUC value for this model is 0.9339827, which provides a single number to evaluate the overall performance of the model.