Heart Disease (Logistic and K-NN)

Micah Mayanja

2024-07-17

Contents

Logistic regression & K-Nearest Neighbors.

Performing classification analysis on a data combining 5 popular heart disease datasets already available independently but not combined before. In this dataset, 5 heart datasets are combined over 11 common features which makes it the largest heart disease dataset available so far for research purposes.

The five datasets used for its curation are: Cleveland, Hungarian, Switzerland, Long Beach VA, and Statlog (Heart) Data Set.

```
#rm(list = ls())
setwd("C:/Users/micah/OneDrive/Documents/R/Heart disease")
data1 <- read.csv("~/R/Heart
disease/heart_statlog_cleveland_hungary_final.csv")

head(data1)

##   age sex chest.pain.type resting.bp.s cholesterol fasting.blood.sugar
## 1  40   1               2          140         289                   0
## 2  49   0               3          160         180                   0
## 3  37   1               2          130         283                   0
## 4  48   0               4          138         214                   0
## 5  54   1               3          150         195                   0
## 6  39   1               3          120         339                   0
##   resting.ecg max.heart.rate exercise.angina oldpeak ST.slope target
## 1           0            172               0     0.0        1      0
## 2           0            156               0     1.0        2      1
## 3           1             98               0     0.0        1      0
## 4           0            108               1     1.5        2      1
## 5           0            122               0     0.0        1      0
## 6           0            170               0     0.0        1      0
```

```r
dim(data1)
```

```
## [1] 1190   12
```

```r
sum(is.na(data1))
```

```
## [1] 0
```

The data contains 1190 observations with 12 variables. It should also be noted that the data has no missing values.

Investigate correlation among predictors.

```r
correlation_matrix <- cor(data1)

library(GGally)
```

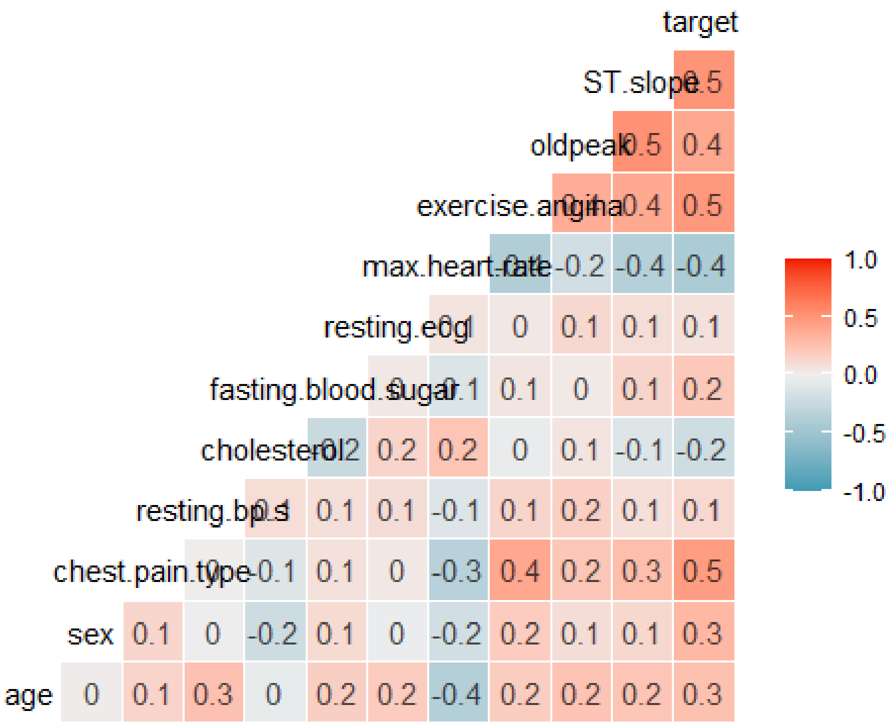
```
## Warning: package 'GGally' was built under R version 4.3.1
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
## Registered S3 method overwritten by 'GGally':
##   method from   
##   +.gg   ggplot2
```

```r
library(ggplot2)
ggcorr(data1, label = TRUE, label_alpha = .7)
```



Based off the correlation matrix plot, note the high correlation between ST.slope and old peak, exercise angina, max heart rate. Also note the high correlation between maximum heart rate and chest pain type.

Data types

Change the data types of categorical variables such as sex from numerical to factor variables and label categories.

```
table(data1$sex)

## 
##   0   1 
## 281 909

class(data1$sex)

## [1] "integer"

data1$sex <- factor(data1$sex,
                    levels=c(0,1),
                    labels = c("Female","Male"))
table(data1$sex)

## 
## Female   Male 
##    281    909
```

```
#table(data1$chest.pain.type)
data1$chest.pain <- factor(data1$chest.pain.type,
                           levels=c(1,2,3,4),
                           labels=c("typical angina","atypical angina",
                                    "non-anginal pain","asymptomatic"))
table(data1$chest.pain)
```

```
## 
##   typical angina  atypical angina non-anginal pain     asymptomatic 
##               66              216              283              625
```

```
#table(data1$fasting.blood.sugar)
data1$fasting.sugar <- factor(data1$fasting.blood.sugar,
                              levels=c(0,1),
                              labels=c("False","True"))
table(data1$fasting.sugar)
```

```
## 
## False  True 
##   936   254
```

```
data1$resting.ecg <- factor(data1$resting.ecg,
                            levels=c(0,1,2),
                            labels=c("Normal","ST-T abnormality",
                                     "Left ventricular hypertrophy"))
table(data1$resting.ecg)
```

```
## 
##                       Normal             ST-T abnormality 
##                          684                          181 
## Left ventricular hypertrophy 
##                          325
```

```
data1$exercise.angina <- factor(data1$exercise.angina,
                                levels=c(0,1),
                                labels=c("No","Yes"))
table(data1$exercise.angina)
```

```
## 
##  No Yes 
## 729 461
```

```
data1$ST.slope[data1$ST.slope == 0] <- 1
data1$ST.slope <- factor(data1$ST.slope,
                         levels=c(1,2,3),
                         labels=c("upsloping","flat","downsloping"))
table(data1$ST.slope)
```

```
## 
##   upsloping        flat downsloping 
##         527         582          81
```

```
data1$target <- factor(data1$target,
                       levels=c(0,1),
                       labels=c("No disease","Heart disease"))
table(data1$target)
```

```
## 
##     No disease Heart disease 
##            561           629
```

Descriptive Statistics

```
#Continuous variables

attach(data1)

par(mfrow=c(3,2))
hist(age)
hist(resting.bp.s, main="Histogram of resting blood pressure")
hist(cholesterol)
hist(max.heart.rate, main="Histogram of maximum heart rate achieved")
hist(oldpeak)

#Calculate means, medians, standard deviation and IQR
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.1
```

```
## 
## Attaching package: 'dplyr'
```

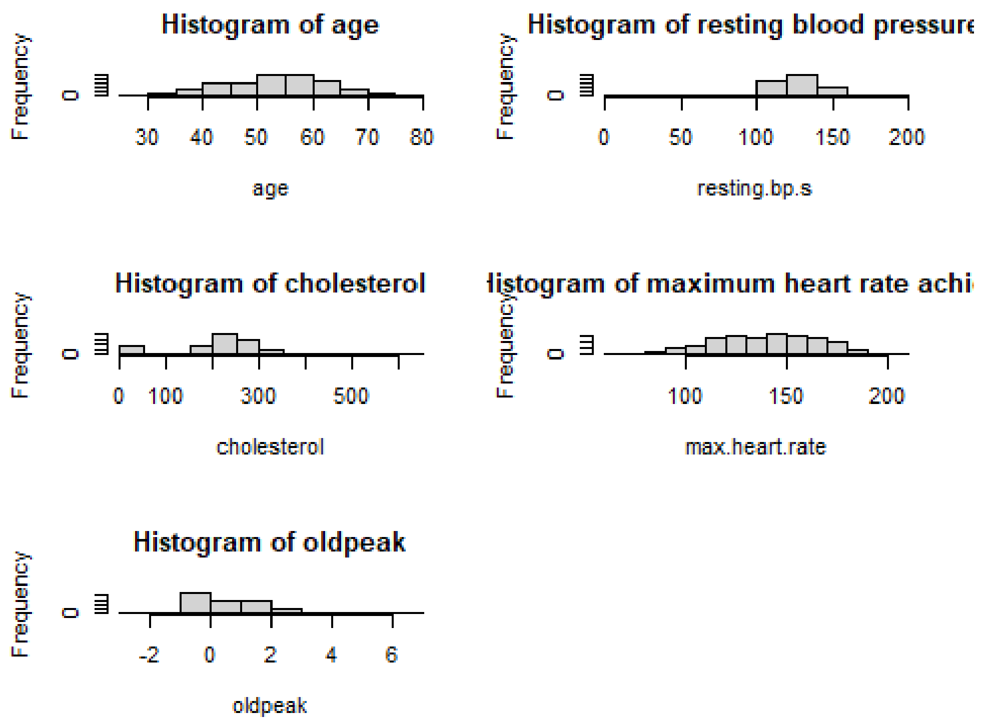
```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
descriptive <- data1 %>%
  select(age,resting.bp.s,cholesterol,max.heart.rate,oldpeak) %>%
  summarise_all(list(min,max,mean,sd,median,IQR))

print(descriptive)
```

```
##   age_fn1 resting.bp.s_fn1 cholesterol_fn1 max.heart.rate_fn1 oldpeak_fn1
## 1      28                0               0                 60        -2.6
##   age_fn2 resting.bp.s_fn2 cholesterol_fn2 max.heart.rate_fn2 oldpeak_fn2
## 1      77              200             603                202         6.2
##    age_fn3 resting.bp.s_fn3 cholesterol_fn3 max.heart.rate_fn3 oldpeak_fn3
## 1 53.72017         132.1538        210.3639           139.7328   0.9227731
##    age_fn4 resting.bp.s_fn4 cholesterol_fn4 max.heart.rate_fn4 oldpeak_fn4
## 1 9.358203         18.36882        101.4205           25.51764    1.086337
##   age_fn5 resting.bp.s_fn5 cholesterol_fn5 max.heart.rate_fn5 oldpeak_fn5
## 1      54              130             229              140.5         0.6
##   age_fn6 resting.bp.s_fn6 cholesterol_fn6 max.heart.rate_fn6 oldpeak_fn6
## 1      13               20           81.75                 39         1.6
```



```
#Factor variables
#Calculate proportions for multiple variables
combined_proportions <- data1 %>%
  tidyr::gather(key = "variable", value = "value", sex, chest.pain,
fasting.sugar,resting.ecg, exercise.angina,
                ST.slope,target) %>%
  group_by(variable, value) %>%
  summarize(count = n()) %>%
  group_by(variable) %>%
  mutate(proportion = count / sum(count))

## Warning: attributes are not identical across measure variables; they will
be
## dropped

## `summarise()` has grouped output by 'variable'. You can override using the
## `.groups` argument.

print(combined_proportions)

## # A tibble: 18 × 4
## # Groups:   variable [7]
##    variable        value                        count proportion
##    <chr>           <chr>                       <int>    <dbl>
##  1 ST.slope        downsloping                    81   0.0681
##  2 ST.slope        flat                          582   0.489
##  3 ST.slope        upsloping                     527   0.443
##  4 chest.pain      asymptomatic                  625   0.525
##  5 chest.pain      atypical angina               216   0.182
##  6 chest.pain      non-anginal pain              283   0.238
##  7 chest.pain      typical angina                 66   0.0555
##  8 exercise.angina No                            729   0.613
##  9 exercise.angina Yes                           461   0.387
## 10 fasting.sugar   False                         936   0.787
## 11 fasting.sugar   True                          254   0.213
## 12 resting.ecg     Left ventricular hypertrophy  325   0.273
## 13 resting.ecg     Normal                        684   0.575
## 14 resting.ecg     ST-T abnormality              181   0.152
## 15 sex             Female                        281   0.236
## 16 sex             Male                          909   0.764
## 17 target          Heart disease                 629   0.529
## 18 target          No disease                    561   0.471
```

Logistic Regression

```
logit1 <-
glm(target~sex+chest.pain+fasting.sugar+resting.ecg+exercise.angina+ age+

resting.bp.s+cholesterol+max.heart.rate+oldpeak,data=data1,family = binomial)

summary(logit1)

##
## Call:
## glm(formula = target ~ sex + chest.pain + fasting.sugar + resting.ecg +
##     exercise.angina + age + resting.bp.s + cholesterol + max.heart.rate +
##     oldpeak, family = binomial, data = data1)
##
## Coefficients:
##                                          Estimate Std. Error z value
Pr(>|z|)
## (Intercept)                            -1.5840687  1.1012453  -1.438
0.1503
## sexMale                                 1.3100306  0.2083787   6.287
3.24e-10
## chest.painatypical angina              -0.2692564  0.3805399  -0.708
0.4792
## chest.painnon-anginal pain              0.0126788  0.3396094   0.037
0.9702
## chest.painasymptomatic                  1.6176821  0.3317116   4.877
1.08e-06
## fasting.sugarTrue                       0.9337392  0.2141872   4.359
1.30e-05
## resting.ecgST-T abnormality            -0.1022413  0.2546271  -0.402
```

```
0.6880
## resting.ecgLeft ventricular hypertrophy  0.3008586  0.1969282   1.528
0.1266
## exercise.anginaYes                       1.1486696  0.1873302   6.132
8.69e-10
## age                                      0.0171370  0.0101248   1.693
0.0905
## resting.bp.s                             0.0053326  0.0047236   1.129
0.2589
## cholesterol                             -0.0029118  0.0009465  -3.076
0.0021
## max.heart.rate                          -0.0164854  0.0038229  -4.312
1.62e-05
## oldpeak                                  0.6419186  0.0894538   7.176
7.18e-13
##
## (Intercept)
## sexMale                                 ***
## chest.painatypical angina
## chest.painnon-anginal pain
## chest.painasymptomatic                  ***
## fasting.sugarTrue                       ***
## resting.ecgST-T abnormality
## resting.ecgLeft ventricular hypertrophy
## exercise.anginaYes                      ***
## age                                     .
## resting.bp.s
## cholesterol                             **
## max.heart.rate                          ***
## oldpeak                                 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1645.80  on 1189  degrees of freedom
## Residual deviance:  948.59  on 1176  degrees of freedom
## AIC: 976.59
##
## Number of Fisher Scoring iterations: 5
```

```
logit2 <- glm(target~sex+chest.pain+fasting.sugar+exercise.angina+age
              +cholesterol+max.heart.rate+oldpeak,data=data1,family =
binomial)
summary(logit2)
```

```
##
## Call:
## glm(formula = target ~ sex + chest.pain + fasting.sugar + exercise.angina
+
##     age + cholesterol + max.heart.rate + oldpeak, family = binomial,
##     data = data1)
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.2677854  0.9610527  -1.319   0.1871
## sexMale                     1.2898138  0.2071517   6.226 4.77e-10 ***
## chest.painatypical angina  -0.3557215  0.3779937  -0.941   0.3467
## chest.painnon-anginal pain -0.0476018  0.3381876  -0.141   0.8881
## chest.painasymptomatic      1.5529701  0.3295448   4.712 2.45e-06 ***
## fasting.sugarTrue           0.9424443  0.2127570   4.430 9.44e-06 ***
## exercise.anginaYes          1.1484854  0.1854946   6.191 5.96e-10 ***
## age                         0.0221528  0.0097354   2.275   0.0229 *
## cholesterol                -0.0024700  0.0009143  -2.701   0.0069 **
## max.heart.rate             -0.0153385  0.0037406  -4.101 4.12e-05 ***
## oldpeak                     0.6511051  0.0886588   7.344 2.07e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1645.80  on 1189  degrees of freedom
## Residual deviance:  952.53  on 1179  degrees of freedom
## AIC: 974.53
##
## Number of Fisher Scoring iterations: 5
```

The anova() function can also be used to compare nested logistic regression models to determine if adding additional predictors significantly improves the model fit.

```
anova(logit2,logit1,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + chest.pain + fasting.sugar + exercise.angina +
##     age + cholesterol + max.heart.rate + oldpeak
## Model 2: target ~ sex + chest.pain + fasting.sugar + resting.ecg +
exercise.angina +
##     age + resting.bp.s + cholesterol + max.heart.rate + oldpeak
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1179     952.53
## 2      1176     948.59  3   3.9419   0.2678
```

P-value > 0.05, therefore, the complex model does not significantly improve the fit compared to the simplier model.

```
# Get the coefficients of the model
coefficients <- summary(logit2)$coefficients

# Transform the coefficients to odds ratios
```

```
odds_ratios <- exp(coefficients[,"Estimate"])
odds_ratios
```

```
##                 (Intercept)                    sexMale
##                   0.2814542                  3.6321101
##  chest.painatypical angina chest.painnon-anginal pain
##                   0.7006677                  0.9535134
##      chest.painasymptomatic          fasting.sugarTrue
##                   4.7254846                  2.5662465
##          exercise.anginaYes                        age
##                   3.1534132                  1.0224000
##                 cholesterol             max.heart.rate
##                   0.9975331                  0.9847785
##                     oldpeak
##                   1.9176589
```

```
# Confidence intervals
confidence_intervals <- exp(confint(logit2))
```

```
## Waiting for profiling to be done...
```

```
confidence_intervals
```

```
##                                 2.5 %    97.5 %
## (Intercept)                0.04230963 1.8391194
## sexMale                    2.43336409 5.4864693
## chest.painatypical angina  0.33442881 1.4768618
## chest.painnon-anginal pain 0.49392466 1.8655904
## chest.painasymptomatic     2.49668065 9.1156647
## fasting.sugarTrue          1.69851791 3.9142978
## exercise.anginaYes         2.19576124 4.5468569
## age                        1.00313139 1.0421973
## cholesterol                0.99572405 0.9993036
## max.heart.rate             0.97753338 0.9919863
## oldpeak                    1.61682066 2.2895689
```

Split into train and test data

```
set.seed(2)
train_indices <- sample(seq_len(nrow(data1)),size = 0.7*nrow(data1))
train <- data1[train_indices,]
test <- data1[-train_indices,]

glm.fit <- glm(target~sex+chest.pain+fasting.sugar+exercise.angina+age
              +cholesterol+max.heart.rate+oldpeak,data=train,family =
binomial)

glm.prob =predict(glm.fit,test, type="response")

#Compute the predictions using test data.
glm.pred=rep("No",357)
#Probability above 0.5 is predicted as Up
```

```
glm.pred[glm.prob>.50]="Yes"
table(glm.pred,test$target)

##
## glm.pred No disease Heart disease
##      No         139            25
##      Yes         37           156

#correctly predicting heart disease / no heart disease (83%)
(139+156)/357

## [1] 0.8263305

#prediction error (17%)
1 - (139+156)/357

## [1] 0.1736695
```

Using the trained model, the probability of predicting correctly (heart disease / no heart disease) is 0.83. Therefore, the prediction error is approximately 17%.

K-Nearest Neighbors

```
library(class)
train.x <-
cbind(train$sex,train$chest.pain,train$fasting.sugar,train$exercise.angina,

train$age,train$cholesterol,train$max.heart.rate,train$oldpeak)
test.x <-
cbind(test$sex,test$chest.pain,test$fasting.sugar,test$exercise.angina,
                test$age,test$cholesterol,test$max.heart.rate,test$oldpeak)
train.heart <- train$target

#Prediction accuracy with k = 1
set.seed(1)
knn.pred<-knn(train.x,test.x,train.heart,k=1)
table(knn.pred,test$target)

##
## knn.pred        No disease Heart disease
##   No disease           130            35
##   Heart disease         46           146

mean(knn.pred == test$target)

## [1] 0.7731092
```

77% highest K-NN prediction accuracy.

Therefore, the Logistic Model (83%) predicts better than the non-parametric K-Nearest Neighbor model (77%).