Final Project: Heart Disease Prediction Model

Madhav Sigdel, Bishwas Ghimire and Surendra Deuja

11/26/2019

Introduction and Objective

Early diagnosis can be a game changer when it comes to treating life-threatening diseases such as heart related disease. Researchers and doctors are always trying to come up with markers or symptoms and other tell-tale signs that can alert them to a patient's increased health risk.
Clearly, a multitude of variables are involved in a patient's health outcome such as the presence or absense of heart-related complication. A significant subset of these variables can be quantified in one way or another and encoded as numbers (or categories). This naturally qualifies as a situation where the sheer amount of information about a patient can be easily overwhleming to a single doctor or a small group of human scientists in order to come to a meaningful reliable conclusion or prediction, so any help from computers is desirable in this scenatio if it is reasable. This is eactly where data science tools and algorithms can come in handy, which is what motivates our work in this project.

Our main goal is to develop a machine learning model to predict the presence (or risk) of heart disease based on variables like age, sex, blood pressure, chest pain type, maximum heart rate, resting electrocardiographic results, etc. that are likely to have an impact on heart condition.

Data Exploration

We begin by importing the dataset.

```
#import hear data
heart.data = read.csv(file="heart.csv", header=TRUE)
```

Here's some more information about the dataset.

age: age in years
sex: (1 = male; 0 = female)
cp: chest pain type
trestbps: resting blood pressure (in mm Hg on admission to the hospital)
chol: serum cholestoral in mg/dl
fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
restecg: resting electrocardiographic results
thalach: maximum heart rate achieved
exang: exercise induced angina (1 = yes; 0 = no)

oldpeak: ST depression induced by exercise relative to rest
slope: the slope of the peak exercise ST segment
ca: number of major vessels (0-3) colored by flourosopy
thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
target: 1 or 0

Since we can see that some of the explanatory variables are categorical, we convert them into factor variables.

```
#covert factor variables
cols = c('sex','fbs','cp','restecg','exang','ca','slope','thal','target')
heart.data[cols] = lapply(heart.data[cols], factor)
```

Next thing to do is explore the dataset. Let's quickly display the head to look at different columns and their entries.

```
head(heart.data)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

Let's quickly look at the summary statistics.

```
summary(heart.data)
```

```
##       age        sex     cp         trestbps          chol       fbs
##  Min.   :29.00   0: 96   0:143   Min.   : 94.0   Min.   :126.0   0:258
##  1st Qu.:47.50   1:207   1: 50   1st Qu.:120.0   1st Qu.:211.0   1: 45
##  Median :55.00           2: 87   Median :130.0   Median :240.0
##  Mean   :54.37           3: 23   Mean   :131.6   Mean   :246.3
##  3rd Qu.:61.00                   3rd Qu.:140.0   3rd Qu.:274.5
##  Max.   :77.00                   Max.   :200.0   Max.   :564.0
##  restecg    thalach      exang      oldpeak     slope   ca      thal
target
##  0:147   Min.   : 71.0   0:204   Min.   :0.00   0: 21   0:175   0:  2
0:138
##  1:152   1st Qu.:133.5   1: 99   1st Qu.:0.00   1:140   1: 65   1: 18
1:165
##  2:  4   Median :153.0           Median :0.80   2:142   2: 38   2:166
```

```
##           Mean   :149.6           Mean   :1.04           3: 20   3:117
##           3rd Qu.:166.0           3rd Qu.:1.60           4:  5
##           Max.   :202.0           Max.   :6.20
```

Data Visualization

R's ggplot2 package allows us to make plots that are colored by target classes, which can be helpful. For instance, we can see if the effect of cholesterol on heart disease is different for different ages.

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

gg = ggplot(heart.data, aes(x=age, y=chol)) +
  geom_point(aes(col=target))
plot(gg)
```

We can see in the plot that dots with one color (target = 1 or heart disease present) are clustered to one side and another color dots (target = 0 or heart disease not present) are clustered to the other side. Shortly speaking, whether a certain level of cholesterol is indicative of heart problem in a patient or not seems to depend on how old the patient is.

Let's plot another one of these plots for maximum heart rate achieved by the patient.

```
gg = ggplot(heart.data, aes(x=age, y=thalach)) +
  geom_point(aes(col=target))
plot(gg)
```

Train-test Split

After our preliminary exploration of the dataset, we can begin model development, but before we begin building any model, let's sequester a portion of the dataset for cross-validation. Below, we do a 80-20 train-test split on our original dataset.

```
################################################
#### Train-Test Split for Cross Validation #####

#define the size of the training set
train.idx <- floor((nrow(heart.data)/5)*4)

#sample rows
heart.data <- heart.data[sample(nrow(heart.data)), ]
#get training set
heart.train <- heart.data[1:train.idx, ]
#set aside test set
heart.test <- heart.data[(train.idx+1):nrow(heart.data), ]
```

Logistic Regression Model

Using the training data, we will develop a model that can help us predict our target variable, and we will cross-validate the model's performance on the sequestered test data. Since our target is a binary categorical variable, we need a binary classification algorithm. We will use a logistic regression model, which is the classification counterpart of a linear regression model. To build the logistic regression model, we can use R's glm function with family = 'binomial'.

```
my.model = glm(target~age+sex+cp+thalach+exang+oldpeak+ca + I(age*thalach),
data = heart.train, family = "binomial")
summary(my.model)
```

```
##
## Call:
## glm(formula = target ~ age + sex + cp + thalach + exang + oldpeak +
##     ca + I(age * thalach), family = "binomial", data = heart.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4776  -0.3345   0.1091   0.4716   2.4933
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -19.353139  11.634410  -1.663 0.096224 .
## age                0.301189   0.197182   1.527 0.126644
## sex1              -1.539859   0.493350  -3.121 0.001801 **
## cp1                0.675284   0.628151   1.075 0.282359
## cp2                2.121495   0.568878   3.729 0.000192 ***
## cp3                2.596694   0.826984   3.140 0.001690 **
## thalach            0.143303   0.075987   1.886 0.059309 .
## exang1            -1.341582   0.541731  -2.476 0.013269 *
## oldpeak           -0.801288   0.245910  -3.258 0.001120 **
## ca1               -2.663462   0.580081  -4.592  4.4e-06 ***
## ca2               -2.885146   0.774316  -3.726 0.000194 ***
## ca3               -1.888752   0.868853  -2.174 0.029717 *
## ca4                0.116694   1.585122   0.074 0.941314
## I(age * thalach)  -0.001980   0.001305  -1.517 0.129210
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 334.14  on 241  degrees of freedom
## Residual deviance: 150.64  on 228  degrees of freedom
## AIC: 178.64
##
## Number of Fisher Scoring iterations: 6
```

```
# cv.err = cv.glm(heart.train,mymodel,K=10)
# cv.err$delta
```

Interpretation

Let's talk about interpreting the model results. The model shown here is a logistic regression model that can predict the binary target variable, which is the presence or absence of heart disease. The coefficients in the model can be interpreted as the marginal effect on log odds of the target variable (log(p/(1-p)) where p is the probability ranging from 0 to 1). For instance, unit increase in age increases the log odds of having heart disease by 0.5, or, being male reduces the log odds by 1.51 points.

Model Development Strategy

One thing that needs to be mentioned here is that the model shown here is not the first model we picked. We started by taking into account all the variables that are avaiable to us. We then dropped the variables that are not statistically significant by looking at the p-values and deviance reduction. We confirmed that the variables dropped are statistically insignificant by using likelihood-ratio test. An exmaple is shown below:

```
###If model0 is a logistic regression submodel of model, then
###LRtest compares them with a likelihood ratio test and returns
###the p-value.
LRtest=function(model0,model){
  1-pchisq(model0$deviance-model$deviance,
           df=model0$df.residual-model$df.residual)}

big.model = glm(target~age+sex+cp+thalach+exang+oldpeak+ca +
I(age*thalach)+slope+chol+restecg+fbs, data = heart.train, family =
"binomial")

small.model = my.model
LRtest(small.model,big.model)

## [1] 0.0113276
```

The p-value shows that we don't have evidence to reject the null hypothesis that variables 'slope', 'chol', 'restecg', 'fbs' are statistically insignificant. Hence, we can drop these from our model. We visually explore the dataset to look for any suggestions of interaction among different variables. When we see something visually, we check to see if it is statistically significant. Notice that we have decided to include an interaction term (age * thalach) in our final model.

Analysis of Deviance

To quickly see how each varaible is reducing the residual deviance, we can use 'anova'.

```r
anova(my.model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                241     334.14
## age               1   15.481       240     318.66 8.333e-05 ***
## sex               1   21.193       239     297.47 4.152e-06 ***
## cp                3   61.083       236     236.39 3.451e-13 ***
## thalach           1   20.322       235     216.06 6.545e-06 ***
## exang             1    8.313       234     207.75  0.003937 **
## oldpeak           1   15.529       233     192.22 8.125e-05 ***
## ca                4   39.192       229     153.03 6.357e-08 ***
## I(age * thalach)  1    2.395       228     150.64  0.121735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Diagnostics (using group plots of π versus x-groups)

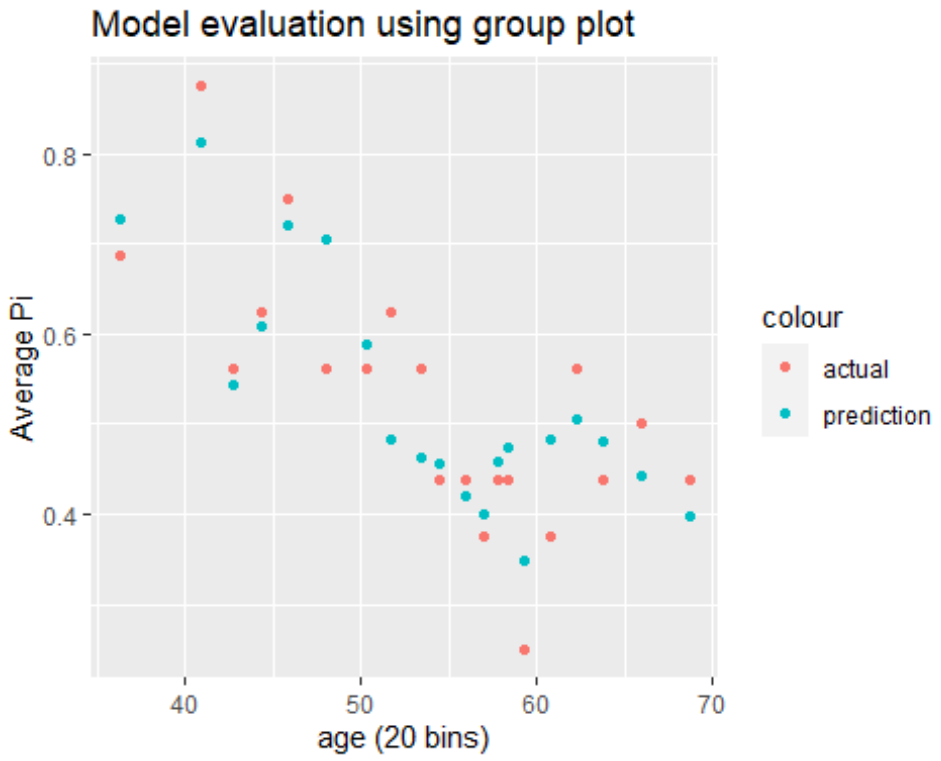
Since we are working with logistic regression, the residuals don't have the same meaning and role in model diagnostics as they do for linear regression because our target variable is now binary with two distint categories. When we make the prediction using our model, we would expect to output either a 0 or a 1, signifying either a 'yes' or a 'no' for the presence or absense of heart disease. Thus, we cannot do possibly do things like checking for the normality of errors as it would make no sense to do that. The error would itself always be a 0 (if we get the target right) or a 1 (if we miss the target).

Without getting into the details, in logistic regression, the output of the model is always a number between 0 and 1 and can actually be interpreted as the probability that the target variable is 1.

Similarly, we can make group plots where we can group an independent variable like age in our case into many bins and plot the fraction of the targets in that bin that are true ('1' or 'yes'). This number would rougly mean the probablity that the target is 1 if an individual belongs to that particular age bin.

Since we cannot analyze the residuals, in order to ensure that our model is making reasonable and sensible predictions, we will look at the group plot of the fraction of true target values for each age bin and compare that to the group plot of the mean predicted target probabilities for that bin.

```r
#sigmoid maps a real number to a number between 0 and 1
pi = function(g){
  return(exp(g)/(1+exp(g)))}
#logit calculates the logit of a number p between 0 and 1.
logit=function(p){
  log(p/(1-p))}

###Given a vector of numerical values x, a response vector Y, and a
###number of groups k, groupplot returns a list.
###x=mean of x groups
###Pi=mean of Y values in each group (with Wilson's adjustment)
###g=logit of Pi
groupplot=function(x,Y,k){
  sortframe=sort(x,index=TRUE)
  x=sortframe$x
  Y=Y[sortframe$ix]
  xmeans=1:k
  Pi=1:k
  s=floor(length(x)/k)  #groupsize

  for(i in 1:k){
    index=(((i-1)*s+1):(i*s))
    xmeans[i]=mean(x[index])
    Pi[i]=(sum(Y[index])+2)/(s+4)}

  g=logit(Pi)
  return(list(x=xmeans,g=g,Pi=Pi))}

as.numeric.factor = function(x) {as.numeric(levels(x))[x]}

#number of bins
bins = 20

group.true =
groupplot(heart.train$age,as.numeric.factor(heart.train$target),bins)

y.hat = predict(my.model,newdata = heart.train, type = 'response' )
group.pred = groupplot(heart.train$age,y.hat,bins)

x1 = group.true$x
y1 = group.true$Pi
x2 = group.pred$x
y2 = group.pred$Pi
df <- data.frame(x1, y1, x2, y2)

ggplot(df) + geom_point(aes(x = x1, y = y1, col='actual')) + geom_point(aes(x
= x2, y = y2,col='prediction')) +
labs(x = 'age (20 bins)', y = 'Average Pi', title ='Model evaluation using
group plot')
```



As we can see in the plot above, actual and predicted values follow a similar trend, which tells us the model is making reasonable prediction. We have to note that our model is multivariate, and we are only looking at a single variable, age in this plot. However, the predicted probabilities from the logistic model take into account the whole row of data with all relevant features, so the predicted mean 'Pi' values for each age bin incorporate the model's performance in predicting the target variable based on all dependent variables.

Cross Validation

To cross-validate our model, we will evaluate its performance on the test data that it has not seen before. Our performance metric will be the classification accuracy score which is the fraction of the total targets in the test data that are correctly predicted (1 as 1 and 0 as 0).

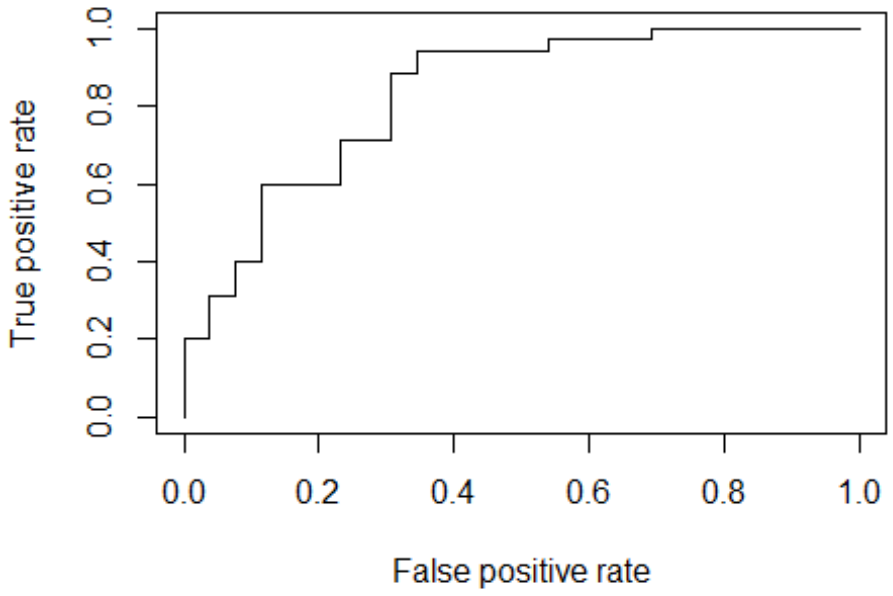
```
test.hat = predict(my.model,newdata=heart.test,type='response')
test.hat = ifelse(test.hat> 0.5,1,0)
miss= mean(test.hat != heart.test$target)
print(paste('Classification Accuracy:',1-miss))

## [1] "Classification Accuracy: 0.754098360655738"
```

The classification accuracy is around 80 %, which is not bad. The model certainly does a decent job in predicting the presence or absense of heart disease given the required variables.

ROC and AUC

Since this is a binary classification problem, another performance metric that we can use is the ROC (receiver operating characteristic) curve which helps analyze how the model reponds to changing the threshold probability. It is basically a plot of true positive rate (also called sensitivity or recall) against the false positive rate (1-specificity).

```
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.6.3

test.hat = predict(my.model,newdata=heart.test,type='response')
pr = prediction(test.hat, heart.test$target)
prf = performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc

## [1] 0.8318681
```

The AUC (area under the curve) is 0.8922, which is pretty good.

Conclusion

Looking at our diagnostic results and performance metrics, we can claim that the logistic regression model we developed is robust and does a pretty reasonable job of predicting heart disease correctly with a classification accuracy of approximately 80%.

Member Contributions

Madhav Sigdel:

Bishwas Ghimire:

Surendra Deuja: