Logistic Regression, Random Forests, and SVM for Statistical Classification of Heart Disease

Natalia Gust-Bardon

September 2019

Preparing Our Data

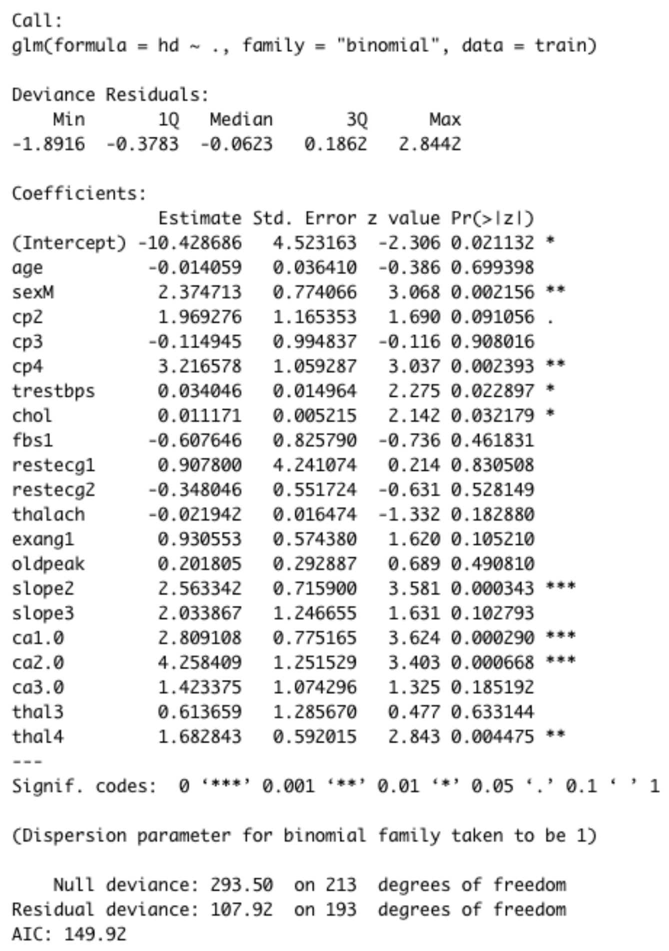
- We will use the Heart Disease Data Set from the UCI Machine Learning Repository (Download)
- The data set requires cleaning and some transformation
- We will prepare the training and test sets

Goal of the Project

- Our goal is to find the most accurate model predicting if a patient has a heart disease or not

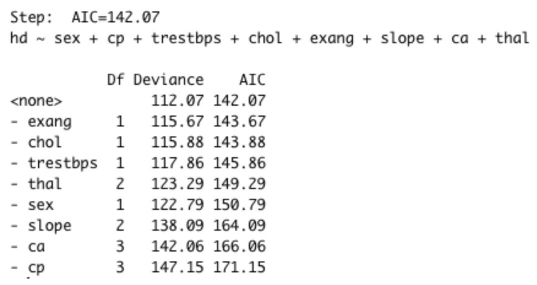
Logistic Regression

- We run the logistic regression on the training data set using the *glm()* function in R:



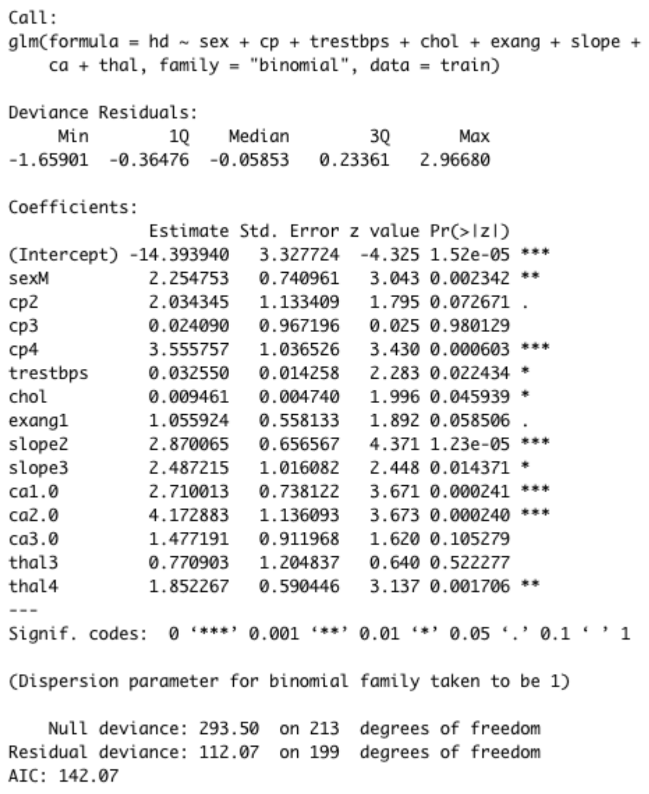
```
Call:
glm(formula = hd ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8916  -0.3783  -0.0623   0.1862   2.8442

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.428686   4.523163  -2.306 0.021132 *
age          -0.014059   0.036410  -0.386 0.699398
sexM          2.374713   0.774066   3.068 0.002156 **
cp2           1.969276   1.165353   1.690 0.091056 .
cp3          -0.114945   0.994837  -0.116 0.908016
cp4           3.216578   1.059287   3.037 0.002393 **
trestbps      0.034046   0.014964   2.275 0.022897 *
chol          0.011171   0.005215   2.142 0.032179 *
fbs1         -0.607646   0.825790  -0.736 0.461831
restecg1      0.907800   4.241074   0.214 0.830508
restecg2     -0.348046   0.551724  -0.631 0.528149
thalach      -0.021942   0.016474  -1.332 0.182880
exang1        0.930553   0.574380   1.620 0.105210
oldpeak       0.201805   0.292887   0.689 0.490810
slope2        2.563342   0.715900   3.581 0.000343 ***
slope3        2.033867   1.246655   1.631 0.102793
ca1.0         2.809108   0.775165   3.624 0.000290 ***
ca2.0         4.258409   1.251529   3.403 0.000668 ***
ca3.0         1.423375   1.074296   1.325 0.185192
thal3         0.613659   1.285670   0.477 0.633144
thal4         1.682843   0.592015   2.843 0.004475 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 293.50  on 213  degrees of freedom
Residual deviance: 107.92  on 193  degrees of freedom
AIC: 149.92
```

- Some independent variables do not appear to be significant, hence we perform variable selection. The lowest AIC was diagnosed with the following regressors:



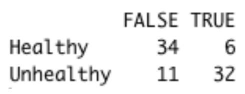
```
Step:  AIC=142.07
hd ~ sex + cp + trestbps + chol + exang + slope + ca + thal

           Df Deviance    AIC
<none>          112.07 142.07
- exang     1   115.67 143.67
- chol      1   115.88 143.88
- trestbps  1   117.86 145.86
- thal      2   123.29 149.29
- sex       1   122.79 150.79
- slope     2   138.09 164.09
- ca        3   142.06 166.06
- cp        3   147.15 171.15
```

- After reducing the number of variables, we get all the significant predictors

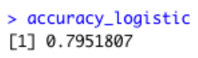


```
Call:
glm(formula = hd ~ sex + cp + trestbps + chol + exang + slope + 
    ca + thal, family = "binomial", data = train)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.65901  -0.36476  -0.05853   0.23361   2.96680  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -14.393940   3.327724  -4.325 1.52e-05 ***
sexM          2.254753   0.740961   3.043 0.002342 ** 
cp2           2.034345   1.133409   1.795 0.072671 .  
cp3           0.024090   0.967196   0.025 0.980129    
cp4           3.555757   1.036526   3.430 0.000603 ***
trestbps      0.032550   0.014258   2.283 0.022434 *  
chol          0.009461   0.004740   1.996 0.045939 *  
exang1        1.055924   0.558133   1.892 0.058506 .  
slope2        2.870065   0.656567   4.371 1.23e-05 ***
slope3        2.487215   1.016082   2.448 0.014371 *  
ca1.0         2.710013   0.738122   3.671 0.000241 ***
ca2.0         4.172883   1.136093   3.673 0.000240 ***
ca3.0         1.477191   0.911968   1.620 0.105279    
thal3         0.770903   1.204837   0.640 0.522277    
thal4         1.852267   0.590446   3.137 0.001706 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 293.50  on 213  degrees of freedom
Residual deviance: 112.07  on 199  degrees of freedom
AIC: 142.07
```

- We now use test data set to see how well the model predicts. For this purpose, we run the confusion matrix and count the accuracy. The classifier is correct in 79.5% cases.



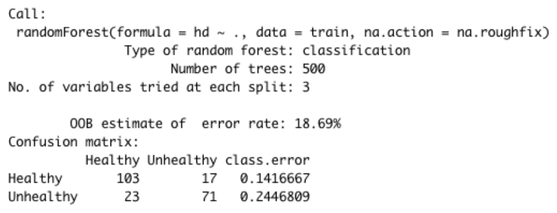
```
           FALSE TRUE
Healthy       34    6
Unhealthy     11   32
```



```
> accuracy_logistic
[1] 0.7951807
```

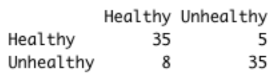
Random Forests

- We run Random Forests on the training data set using the *randomForest()* function

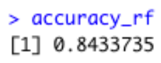


```
Call:
 randomForest(formula = hd ~ ., data = train, na.action = na.roughfix)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18.69%
Confusion matrix:
          Healthy Unhealthy class.error
Healthy       103        17   0.1416667
Unhealthy      23        71   0.2446809
```

- We now use our test data set to see how well the model predicts. For this purpose, we run the confusion matrix and count the accuracy. The classifier is correct in 84.3% cases



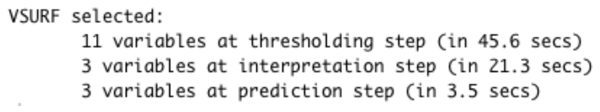
```
            Healthy Unhealthy
Healthy          35         5
Unhealthy         8        35
```



```
> accuracy_rf
[1] 0.8433735
```

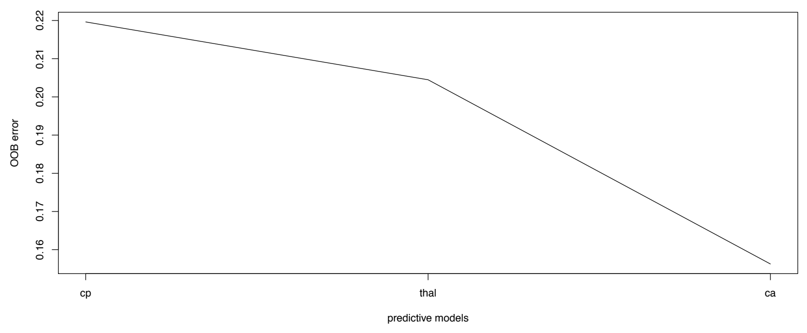
Random Forests: Reduced model

- We now test how the model behaves after removing variables that do not contribute much. For this purpose, we perform variable selection using VSURF() function

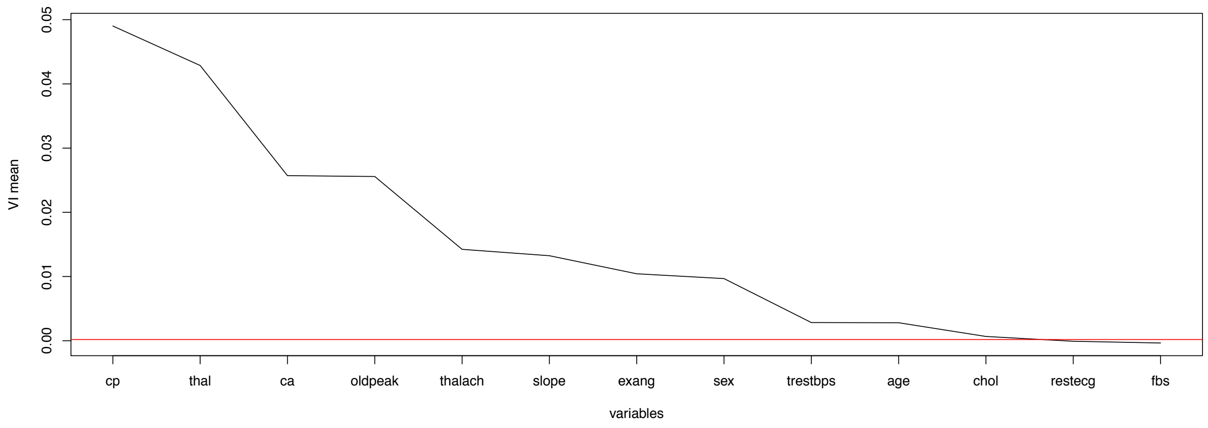


```
VSURF selected:
       11 variables at thresholding step (in 45.6 secs)
       3 variables at interpretation step (in 21.3 secs)
       3 variables at prediction step (in 3.5 secs)
```

- VSURF selected the following three variables for the prediction: cp, thal, and ca



- The variables can be classified into three groups ranging from the most to the least important. The first group contains cp, thal, ca, and oldpeak. The second group of less important variables consists of: thalach, slope, exang, sex, trestbps, age, and chol. Finally, the last group contains two insignificant variables: restecg and fbs. We select three variables chosen by the *varselect.ped* function (cp, thal, and ca). The oldpeak variable is not included as it does not improve the prediction.



- We run Random Forests on the reduced data set

```
Call:
 randomForest(formula = hd ~ ., data = new_train_data, na.action = na.roughfix)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 15.89%
Confusion matrix:
          Healthy Unhealthy class.error
Healthy       109        11  0.09166667
Unhealthy      23        71  0.24468085
```

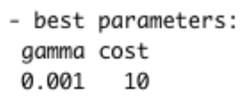
- We now use our reduced test data set to see how well the model predicts. For this purpose, we run the confusion matrix and count the accuracy. The classifier was correct in 80.7% cases

```
          Healthy Unhealthy
Healthy        32         8
Unhealthy       8        35
```

```
> accuracy
[1] 0.8072289
```

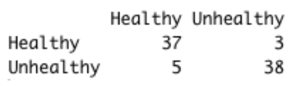
Support Vector Machines

- We use the *tune.svm()* function to find the best gamma and cost arguments. Subsequently, we use the *svm()* function to fit the model

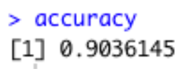


```
- best parameters:
 gamma cost
 0.001   10
```

- We now use our test data set to see how well the model predicts. The classifier is correct in 90.36% cases



```
          Healthy Unhealthy
Healthy        37         3
Unhealthy       5        38
```



```
> accuracy
[1] 0.9036145
```

Summary

- SVM has the highest accuracy of 90.36%
- Logistic regression has the lowest accuracy of 79.5%
- Random Forests with only three explanatory variables has the accuracy of 80.7%

Attachment

```r
# Data Preparation

url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data"

data <- read.csv(url, header=FALSE)

head(data)

colnames(data) <- c(
  "age",
  "sex",
  "cp",
  "trestbps",
  "chol",
  "fbs",
  "restecg",
  "thalach",
  "exang",
  "oldpeak",
  "slope",
  "ca",
  "thal",
  "hd"
)


head(data)

str(data)

data[data =="?"] <- NA
data <- na.omit(data)
data[data$sex == 0,]$sex <-"F"
data[data$sex == 1,]$sex <-"M"

data$sex <- as.factor(data$sex)
data$cp <- as.factor(data$cp)
data$fbs <- as.factor(data$fbs)
data$restecg <- as.factor(data$restecg)
data$exang <- as.factor(data$exang)
data$slope <- as.factor(data$slope)

data$ca <- as.integer(data$ca)
data&ca <- as.factor(data$ca)

data$thal <- as.integer(data$thal)
data$thal <- as.factor(data$thal)

data$hd <- ifelse(test=data$hd == 0, yes="Healthy", no="Unhealthy")
data$hd <- as.factor(data$hd)

str(data)


#Data Partition

set.seed(123)
ind <-sample(2, nrow(data), replace = TRUE, prob = c(0.7, 0.3))
train <- data[ind==1,]
test <-data[ind==2,]

# Logistic Regression Model

logistic_model <- glm(hd ~., data=train, family='binomial')
summary(logistic_model)
```

```r
# Variable Selection

variable_selection <- step(logistic_model)

logistic_model_reduced <- glm(hd ~ sex + cp + trestbps + chol + exang + slope + ca + thal, data=train, family='binomial')
summary(logistic_model_reduced)

#Prediction and accuracy

predicted <- predict(logistic_model_reduced, test, type='response')
predicted
conf_matrix <- table(test$hd, predicted > 0.5)
conf_matrix

accuracy_logistic <- (conf_matrix[[1,1]] + conf_matrix[[2,2]]) / sum(conf_matrix)
accuracy_logistic

# Random Forests

set.seed(42)
rf <- randomForest(hd ~ ., data=train, na.action=na.roughfix)
print(rf)
attributes(rf)
```

```r
# Predictions and accuracy

pred_rf <- predict(rf, test)

confusion_matrix_rf <- table(test$hd, pred_rf)
confusion_matrix_rf

accuracy_rf <- (confusion_matrix_rf[[1,1]] + confusion_matrix_rf[[2,2]]) / sum(confusion_matrix_rf)
accuracy_rf

# Random Forest Variable Selection

selection <- VSURF(hd ~ ., data=train)
summary(selection)

selected_regressors <- selection$varselect.pred
selected_regressors <-as.integer(selected_regressors)

plot(selection, step = "pred", imp.sd = FALSE, var.names = TRUE)

plot(selection, step = "thres", imp.sd = FALSE, var.names = TRUE)

# Random Forest: New Data Set Consisting Only with Selected Variables

new_train_data <- select(train, selected_regressors,14)
new_test_data <- select(test, selected_regressors,14)
```

```r
# Reduced Ranodm Forests Model

rf_new <- randomForest(hd ~ ., data=new_train_data, na.action=na.roughfix)
print(rf_new)

# Reduced Ranodm Forests Model: Predictions and accuracy

pred_rf_new <- predict(rf_new, new_test_data)

confusion_matrix <- table(new_test_data$hd, pred_rf_new)
confusion_matrix

accuracy <- (confusion_matrix[[1,1]] + confusion_matrix[[2,2]]) / sum(confusion_matrix)
accuracy
```

```
# SVM

tunes_parameters <- tune.svm(hd ~ ., data=test, gamma=10^(-5:-1), cost=10^(-3:1))
summary(tunes_parameters)

svm_model <- svm(hd ~., data=test,type="C-classification", kernel="linear", cost=10, gamma=0.001)

# SVM prediction

pred_svm <- predict(svm_model, newdata=test)

confusion_matrix_svm <- table(test$hd, pred_svm)
confusion_matrix_svm

accuracy <- (confusion_matrix_svm[[1,1]] + confusion_matrix_svm[[2,2]]) / sum(confusion_matrix_svm)
accuracy
```