Predicting Risk Factors for Cardiovascular Disease

Group 19

Jason Mullen, Chuong Tran, Vinh Nguyen, Trang Hoang, Nathan Lilly

28 April 2023

Introduction

While modern medicine has made strides in many aspects of human health, cardiovascular heart disease remains one of the leading causes of death today. Statistically speaking, finding ways to combat these diseases can be the difference between an early death and a long and prosperous life. Because of this, our group decided to analyze research done by Kuzak Dempsy, who compiled the data that we are using, taking over 70 thousand accounts of various variables that could help better determine the risk of cardiovascular heart disease.

Our group settled on the question "What factors contribute to cardiovascular heart disease?" to better educate ourselves on the parameters of good health in addition to providing this research and information to our peers and educators. To answer this question, our team decided to utilize logistic regression and decision tree as well as the k-fold Cross Validation method.

Variables

Input:

- Age: Age of individuals (integer).
- Gender: Gender of individuals (string).
- Height: height of individuals in centimeter (integer).
- Weight: weight of individuals in kilograms (integer).
- Ap_hi: Systolic blood pressure reading. (Integer).
- Ap_lo: Diastolic blood pressure reading. (Integer).
- Cholesterol: Cholesterol level of the individual. (Integer).
- Gluc: Glucose level of the individual. (Integer).
- Smoke: Smoking status of the individual. (Boolean).
- Alco: Alcohol consumption status of the individual. (Boolean).
- Active: Physical activity level of the individual. (Boolean).

Output:

- Cardio: Presence (1) or absence (0) of cardiovascular disease. (Boolean).

Question for the Data

- We want to understand the key findings and insights which can affect the results and find which parameters have the greatest impact on cardiovascular health.

Methods and Results

Logistic Regression (Chuong, Trang, Vinh)

Advantage: Logistic regression is designed to predict binary outcomes, provides interpretable results, has low computational requirements, and is robust to noise in the data (overfitting).

Disadvantage: Non-linear problems can't be solved because logistic regression has a linear decision surface.

Formula:

$$\widehat{y(x)} = \frac{\exp(-8.49 + 1.49e\text{-}04*age + 1.53e\text{-}02*gender - 5.73e\text{-}03*height + 1.54e\text{-}02*weight + 3.95e\text{-}02*ap\_hi + 3e\text{-}04*ap\_lo + 5.23e\text{-}01*cholesterol - 1.18e\text{-}01*gluc - 1.32e\text{-}01*smoke - 1.69e\text{-}01*alco - 2.1e\text{-}01*active)}{1 + \exp(-8.49 + 1.49e\text{-}04*age + 1.53e\text{-}02*gender - 5.73e\text{-}03*height + 1.54e\text{-}02*weight + 3.95e\text{-}02*ap\_hi + 3e\text{-}04*ap\_lo + 5.23e\text{-}01*cholesterol - 1.18e\text{-}01*gluc - 1.32e\text{-}01*smoke - 1.69e\text{-}01*alco - 2.1e\text{-}01*active)}$$

Thought process when considering fitting the model

After reviewing our data and seeing that it is clean to start building the models, we began to build a logistic regression model. Prior to that, the "as.factor" command was used to ensure that our Boolean variables were converted to factor variables. We also assume that the data follows LINE when building logistic regression.

Upon completion of the model, we observed that the p-value for *Gender* was greater than 0.05. Therefore, we concluded that *Gender* was not a significant variable for the model. *Age, Height, Weight, ap_hi, ap_lo, Cholesterol, Gluc, Smoke, Alco*, and *Active* were all significant variables for the model. We then removed the *Gender* variable from our future calculations and continued building our models.

Subdivide your full data set in 80% for training, and 20% for testing

```
set.seed(101)
sample <- sample.int(n = nrow(heart_data), size = round(.8*nrow(heart_data),0
), replace=F)
train <- heart_data[sample,]
test <- heart_data[-sample,]

data.train2 = glm(cardio ~ age+height+weight+ap_hi+ap_lo+cholesterol+gluc+smo
ke+alco+active, family = "binomial", data = train) ; summary(data.train2)


## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.9563   0.1751   0.9764   4.8660
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.045e+00  2.290e-01 -39.493  < 2e-16 ***
## age          1.487e-04  4.002e-06  37.145  < 2e-16 ***
## height      -5.124e-03  1.235e-03  -4.148 3.36e-05 ***
## weight       1.510e-02  7.409e-04  20.384  < 2e-16 ***
## ap_hi        4.339e-02  6.947e-04  62.463  < 2e-16 ***
## ap_lo        2.442e-04  6.987e-05   3.495 0.000473 ***
## cholesterol  5.115e-01  1.685e-02  30.357  < 2e-16 ***
## gluc        -1.015e-01  1.916e-02  -5.300 1.16e-07 ***
## smoke1      -1.092e-01  3.600e-02  -3.033 0.002418 **
## alco1       -1.714e-01  4.546e-02  -3.771 0.000162 ***
## active1     -2.117e-01  2.362e-02  -8.960  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 77632  on 55999  degrees of freedom
## Residual deviance: 64415  on 55989  degrees of freedom
## AIC: 64437
##
## Number of Fisher Scoring iterations: 6
```

```
glm.pred2 = predict.glm(data.train2, newdata = test, type = "response")
yHat2 <- glm.pred2 > 0.5
table(test$cardio, yHat2)
```

```
##    yHat2
##     FALSE TRUE
##   0  5427 1674
##   1  2189 4710
```

Updated formula without *Gender* variable:

$$\widehat{y(x)} = \frac{\exp(-8.49 + 1.49e\text{-}04*age - 5.73e\text{-}03*height + 1.54e\text{-}02*weight + 3.95e\text{-}02*ap\_hi + 3e\text{-}04*ap\_lo + 5.23e\text{-}01*cholesterol - 1.18e\text{-}01*gluc - 1.32e\text{-}01*smoke - 1.69e\text{-}01*alco - 2.1e\text{-}01*active)}{1 + \exp(-8.49 + 1.49e\text{-}04*age - 5.73e\text{-}03*height + 1.54e\text{-}02*weight + 3.95e\text{-}02*ap\_hi + 3e\text{-}04*ap\_lo + 5.23e\text{-}01*cholesterol - 1.18e\text{-}01*gluc - 1.32e\text{-}01*smoke - 1.69e\text{-}01*alco - 2.1e\text{-}01*active)}$$

- From this output, we were able to conduct the R^2 calculation to see how close is the fit to being pefect or the worst:
  - $R^2 = \frac{residual\ deviance}{null\ deviance} = \frac{64415}{77631} = 0.83$, this indicates that about 83% variance can be explained by this model.
  - Error rate = $\frac{1674 + 2189}{5427 + 1674 + 2189 + 4710}$ = 27.6%.

Repeating the previous process 10 times

```
set.seed(10)
error <- c()
for (i in 1:10) {sampleLoop <- sample.int(n = nrow(heart_data), size = round(
.8*nrow(heart_data),0), replace=F)
  trainLoop <- heart_data[sampleLoop,]
  testLoop <- heart_data[-sampleLoop,]
  data.trainLoop = glm(cardio ~ . -id -index -gender,
family = "binomial", data = trainLoop)
  glm.predLoop = predict.glm(data.trainLoop, newdata = testLoop, type = "resp
onse")
  yHatLoop <- glm.predLoop > 0.5
  tab = table(testLoop$cardio, yHatLoop)

  error[i] = (tab[1,2] + tab[2,1]) / sum(tab)}
mean(error)

## [1] 0.2766429 ~ 28%
```

After repeating the model 10 times, the test error rate remains 28% which shows the

consistency of Logistic Regression Model compared with the tests above.

Cross-Validation for the Logistic Regression Model

```
library(boot)
data.glm = glm(cardio ~. -id -index -gender, family = "binomial", data = hear
t_data)
cost = function(cardio, pi = 0) mean(abs(cardio - pi) > 0.5) #IMPORTANT

set.seed(5)
(cv.15.err = cv.glm(heart_data, data.glm,cost,K = 15)$delta)

## [1] 0.2786571 0.2786848
```

This code does K-fold cross-validation and calculates the test error rate for the logistic regression model. In this case, we chose 15 as our K for this model. This resampling method results in a test error rate of 27.87%, or roughly 28%.

Decision Tree (Trang, Vinh, Chuong, Nathan, Jason)

Advantage: It is easy to interpret and will result in a good prediction on training data.

Disadvantage: Overfits and performs poorly on test data, as trees generally do not have the same level of predictive accuracy.

Formula:

$$cardio \sim age + gender + height + weight + ap_{hi} + ap_{lo} + cholesterol + gluc + smoke + alco + active$$

Thought process when approaching fitting the model

After observing the high error rates in our previous models, we opted to utilize a classification tree in the hopes of achieving a better error rate. As part of the data preparation, we introduced a new binary variable called "*high*" which indicates whether the quality is considered high, with a value of "Yes" and "No" if it is not.

We constructed a single classification tree using all variables except id and index to determine the key variables for the prediction.

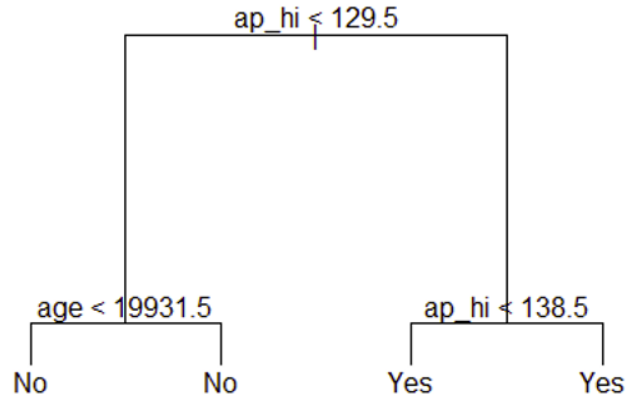
```
High = ifelse(heart_data$cardio == 0, "No","Yes")
High = as.factor(High)
heart_data_2 = data.frame(heart_data,High)

library(tree)

tree.heart = tree(High ~ .-id -index -cardio, data=heart_data_2)
summary(tree.heart)

##
## Classification tree:
## tree(formula = High ~ . - id - index - cardio, data = heart_data_2)
## Variables actually used in tree construction:
## [1] "ap_hi" "age"
## Number of terminal nodes:  4
## Residual mean deviance:  1.135 = 79480 / 70000
## Misclassification error rate: 0.2861 = 20027 / 70000

plot(tree.heart)
text(tree.heart, pretty = 0)
```



This resulted in a 4-node tree, with the variables "*ap_hi*" and "*age*" identified as the most important with the misclassification rate of 29%.

The tree indicates that an individual who has a *Systolic blood pressure (ap_hi)* below 129.5 and is younger or older than 19931.5 days (55 years old) will not have cardiovascular

disease. Conversely, if an individual has an *ap_hi* greater than 129.5, it is likely that they will have cardiovascular disease.

It is necessary to divide the observations into a training and test set as seen below to adequately assess the accuracy of a classification tree model on this dataset.

The data set has been subdivided into 80% for training and 20% for testing

```
set.seed(2)
train.tree = sample(1:round(0.8*nrow(heart_data_2)),)
heart.test = heart_data_2[-train.tree,]
High.test = High[-train.tree]
tree.heart = tree(High ~ . -id -index -cardio, heart_data_2,subset =
train.tree)
tree.pred = predict(tree.heart,heart.test,type = "class")
(tab.seats = table(tree.pred,High.test))
```

```
##          High.test
## tree.pred   No  Yes
##       No  5590 2673
##       Yes 1402 4335
```

$$\text{Test error rate} = \frac{2673 + 1402}{5590 + 2673 + 1402 + 4335} = 29.1\%.$$

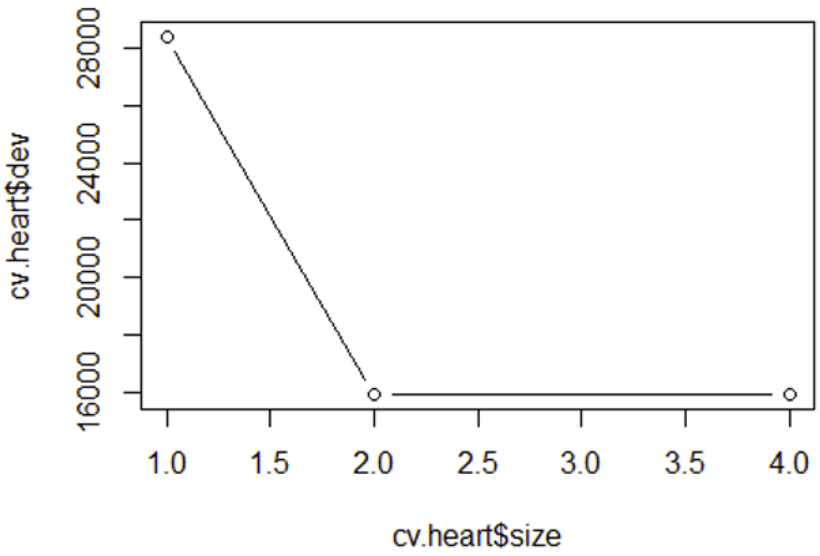
Despite our efforts, the misclassification error rate remained at approximately 29.1% which is similar to the previous models. We then proceeded to prune the tree to enhance its predictive accuracy and hopefully reduce the error rate.

Cross-Validation for Pruning

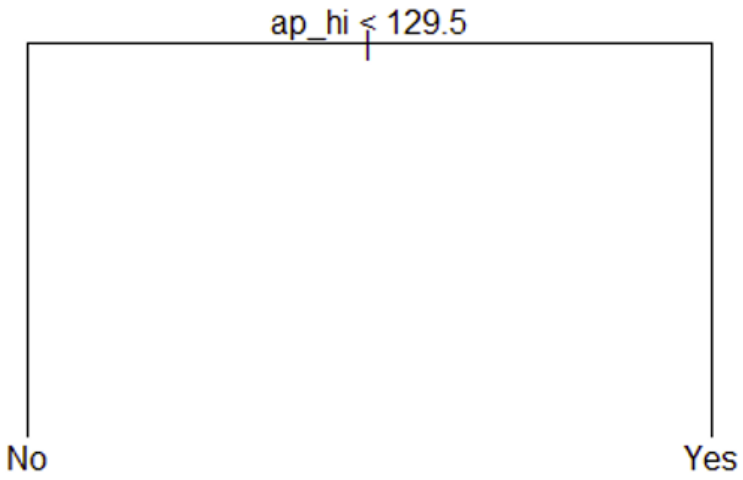
```
set.seed(3)
cv.heart = cv.tree(tree.heart,FUN = prune.misclass)
cv.heart
```

```
## $size
## [1] 4 2 1
##
## $dev
## [1] 15952 15952 28384
##
## $k
## [1]  -Inf     0 12019
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"   "tree.sequence"
```

```
plot(cv.heart$size,cv.heart$dev,type = "b")
```




```
prune.heart = prune.misclass(tree.heart,best = 2)
plot(prune.heart)
text(prune.heart,pretty = 0)
```




```
tree.pred = predict(prune.heart,heart.test,type = "class")
(tab.prune = table(tree.pred,High.test))
```

```
##          High.test
## tree.pred   No  Yes
##       No  5590 2673
##       Yes 1402 4335
```

Test error rate = $\frac{2673 + 1402}{5590 + 2673 + 1402 + 4335}$ = 29.1%.

Subsequently, we opted to perform tree pruning through cross-validation. Although the tree is pruned, it results in the same error rate of 29.1%.

Repeat 10 times

```
set.seed(10)
error.tree <- c()
for (i in 1:10) {
   trainLoop.tree = sample(1:round(0.8*nrow(heart_data_2)),)
   heartLoop.test = heart_data_2[-trainLoop.tree,]
   High.testLoop = High[-train.tree]

  tree.heartLoop = tree(High ~ . -id -index -cardio, heart_data_2,subset = tr
ainLoop.tree)
   tree.predLoop = predict(tree.heartLoop, heartLoop.test, type = "class")
   tabheart = table(tree.predLoop, High.testLoop)

  error.tree[i] = (tabheart[1,2] + tabheart[2,1]) / sum(tabheart)
}
mean(error.tree)
```

Despite our efforts, after repeating the model 10 times the test error rate remains 29.1%. It appears there is only so much that can be done in this case, and the error rate holds steady.

Conclusion

Uncovering the underlying factors behind cardiovascular heart disease risk could one day allow for improved health across the world. Our research has suggested that, while not perfect, we can gain insight into the factors behind the risk. When comparing the test error rates of logistic regression and decision tree models, we found that logistic regression had a slightly lower error rate (28% < 29.1%). Therefore, we determined that logistic regression performed better. Additionally, our analysis revealed that the most significant variables for predicting the disease were *ap_hi* and *age*. While we now have a greater understanding of factors that impact cardiovascular disease risk, we acknowledge that we cannot yield 100% accuracy in scientific research. This is primarily due to the multitude of factors and studies involved, and the variance within those factors. Regardless, we hope that our analysis provides valuable insights into cardiovascular disease and the risks associated with it.

Bibliography

Devastator, The. “Risk Factors for Cardiovascular Heart Disease.” Kaggle, 12 Jan. 2023, https://www.kaggle.com/datasets/thedevastator/exploring-risk-factors-for-cardiovascular-diseas.

Dempsy, Kuzak. “Kuzak Dempsy's Datasets.” *Data.world*, 4 May 2021, https://data.world/kudem.

Appendix: R Source Code

```
## Logistic Regression

# heart_data <-
read.csv("\\Users\\nguye\\Downloads\\archive\\heart_data.csv")
# heart_data$gender = as.factor(heart_data$gender)
# heart_data$smoke = as.factor(heart_data$smoke)
# heart_data$alco = as.factor(heart_data$alco)
# heart_data$active = as.factor(heart_data$active)
#
#
# data.glm = glm(cardio ~ . -id -index, family = "binomial", data =
heart_data)
# summary(data.glm)
#
# set.seed(101)
# sample <- sample.int(n = nrow(heart_data), size =
round(.8*nrow(heart_data),0), replace=F)
# train <- heart_data[sample,]
# test <- heart_data[-sample,]


### Without Gender


# data.train2 = glm(cardio ~
age+height+weight+ap_hi+ap_lo+cholesterol+gluc+smoke+alco+active, family =
"binomial", data = train)
# summary(data.train2)
# glm.pred2 = predict.glm(data.train2, newdata = test, type = "response")
# yHat2 <- glm.pred2 > 0.5
# table(test$cardio, yHat2)

# do for loop 10 times
# set.seed(10)
# error <- c()
#
# for (i in 1:10) {
#   sampleLoop <- sample.int(n = nrow(heart_data), size =
round(.8*nrow(heart_data),0), replace=F)
#   trainLoop <- heart_data[sampleLoop,]
#   testLoop <- heart_data[-sampleLoop,]
#   data.trainLoop = glm(cardio ~ . -id -index -gender,
# family = "binomial", data = trainLoop)
#   glm.predLoop = predict.glm(data.trainLoop, newdata = testLoop, type =
"response")
#   yHatLoop <- glm.predLoop > 0.5
#   tab = table(testLoop$cardio, yHatLoop)
#
#   error[i] = (tab[1,2] + tab[2,1]) / sum(tab)
# }
```

```
#
# mean(error)


### Cross-Validation for the Logistic Regression Model


# library(boot)
# data.glm = glm(cardio ~. -id -index -gender, family = "binomial", data =
heart_data)
# cost = function(cardio, pi = 0) mean(abs(cardio - pi) > 0.5) #IMPORTANT
#
# set.seed(5)
# (cv.15.err = cv.glm(heart_data, data.glm,cost,K = 15)$delta)


## Decision Tree


# High = ifelse(heart_data$cardio == 0, "No","Yes")
# High = as.factor(High)
# heart_data_2 = data.frame(heart_data,High)
#
# library(tree)
# tree.heart = tree(High ~ .-id -index -cardio, data=heart_data)
# summary(tree.heart)
#
#
# plot(tree.heart)
# text(tree.heart, pretty = 0)
#
#
# set.seed(2)
# train.tree = sample(1:round(0.8*nrow(heart_data)),)
# heart.test = heart_data[-train.tree,]
# High.test = High[-train.tree]
# tree.heart = tree(High ~ . -id -index -cardio, heart_data,subset =
train.tree)
# tree.pred = predict(tree.heart,heart.test,type = "class")
# (tab.seats = table(tree.pred,High.test))

### Cross-Validation for Prunning

# set.seed(3)
# cv.heart = cv.tree(tree.heart,FUN = prune.misclass)
# cv.heart
# plot(cv.heart$size,cv.heart$dev,type = "b")


# prune.heart = prune.misclass(tree.heart,best = 2)
# plot(prune.heart)
```

```
# text(prune.heart,pretty = 0)
#
#
# tree.pred = predict(prune.heart,heart.test,type = "class")
# (tab.prune = table(tree.pred,High.test))


# do for loop 10 times
# set.seed(10)
# error.tree <- c()
#
# for (i in 1:10) {
#   trainLoop.tree = sample(1:round(0.8*nrow(heart_data_2)),)
#   heartLoop.test = heart_data_2[-trainLoop.tree,]
#   High.testLoop = High[-train.tree]
#
#   tree.heartLoop = tree(High ~ . -id -index -cardio, heart_data_2,subset =
trainLoop.tree)
#   tree.predLoop = predict(tree.heartLoop, heartLoop.test, type = "class")
#   tabheart = table(tree.predLoop, High.testLoop)
#
#   error.tree[i] = (tabheart[1,2] + tabheart[2,1]) / sum(tabheart)
# }
#
# mean(error.tree)
```