Cardiovascular Disease Analysis

Jaspreet Kang

11/6/2019

Introduction

The following analysis is performed on a cardiovascular disease dataset founded on Kaggle. The link to the dataset is below:

https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

There are 70,000 data points and 13 variables in total.

1. **id**
2. **age** (in days | Integer type)
3. **gender** (Numeric): 1 = Female; 2 = Male
4. **height** (in cm | Integer)
5. **weight** (in kg | Numeric)
6. **ap_hi** (Integer): Systolic Blood Pressure
7. **ap_lo** (Integer): Diastolic Blood Pressure
8. **cholesterol** (Integer): 1 = normal; 2 = above normal; 3 = well above normal
9. **gluc** (Integer): 1 = normal; 2 = above normal; 3 = well above normal
10. **smoke** (Integer): 0 = Non-smoker; 1 = Smoker
11. **alco** (Integer): 0 = Non-drinker; 1 = Drinker
12. **active** (Integer): 0 = Not Active; 1 = Active
13. **cardio** (Target Variable | Integer): 0 = No CVD; 1 = CVD

The following research questions and topics will be answered and looked at:

1. At what age does the event of CVD surpass not having CVD?
2. Comparing those with CVD and w/o CVD, which variables show greater risk/correlation? In other words, what variables are the most correlated with cardiovascular disease?
3. Taking a closer look at the relationship between gender and bmi with CVD.

Additionally, I will use and compare the performance of several classification modeling techniques in order to predict cardiovascular disease.

1. Logistic Regression
2. XG Boosting Classifier
3. Random Foresting Classifier

Data Cleaning and Feature Engineering

1. I removed all the data points where diastolic blood pressure was greater than or equal to systolic blood pressure
2. Upon further research, I addressed outliers from diastolic and systolic blood pressure variables by removing observations where

- Systolic blood pressure was greater than 300 or less than 70.
- Diastolic blood pressure was less than or equal to 20

3. I created two new variables:

- **age1** (in years) by converting **age** variable in days to years

- **bmi** by using the variable **height** and **weight** provided

4. There were obvious misentries for height and weight data. This is why **bmi** variable was created. I addressed the outliers and/or misentries by removing observations where bmi >= 50 and bmi <= 15.

Overall 1,234 of the 70,000 observations were removed due to suspicion of inaccurate information entry.

```r
rm(list=ls())
setwd("/home/jaspo/Documents/Cardiovascular-Disease-Analysis-master/CVD/")
cvd <- read.csv(file = "cardio_train.csv", header = T, sep = ";")
str(cvd)
```

```
## 'data.frame':    70000 obs. of  13 variables:
##  $ id         : int  0 1 2 3 4 8 9 12 13 14 ...
##  $ age        : int  18393 20228 18857 17623 17474 21914 22113 22584 17668 19834 ...
##  $ gender     : int  2 1 1 2 1 1 1 2 1 1 ...
##  $ height     : int  168 156 165 169 156 151 157 178 158 164 ...
##  $ weight     : num  62 85 64 82 56 67 93 95 71 68 ...
##  $ ap_hi      : int  110 140 130 150 100 120 130 130 110 110 ...
##  $ ap_lo      : int  80 90 70 100 60 80 80 90 70 60 ...
##  $ cholesterol: int  1 3 3 1 1 2 3 3 1 1 ...
##  $ gluc       : int  1 1 1 1 1 2 1 3 1 1 ...
##  $ smoke      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ alco       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ active     : int  1 1 0 1 0 0 1 1 1 0 ...
##  $ cardio     : int  0 1 1 1 0 0 0 1 0 0 ...
```

```r
# Checking for NA values
colSums(is.na(cvd)) # No NA values
```

```
##          id         age      gender      height      weight       ap_hi
##           0           0           0           0           0           0
##       ap_lo cholesterol        gluc       smoke        alco      active
##           0           0           0           0           0           0
##      cardio
##           0
```

```r
# Cleaning Data
index <- which(cvd$ap_lo >= cvd$ap_hi) # Checking and indexing rows with diastolic blood
# pressure being greater than systolic blood pressure
cvd <- cvd[-index, ] # Removing these rows as they are incorrect entries

# There is a systolic blood pressure value of 16,020. An obvious misentry.
# After further inspection, we remove observations with greater than 300 or lower than
# 70 systolic blood pressure
index <- which(cvd$ap_hi > 300 | cvd$ap_hi < 70)
cvd <- cvd[-index, ]

index <- which(cvd$ap_lo <= 25)
cvd <- cvd[-index, ]

# Creating two new variables: age1 and bmi.
# age1 is age in years
cvd$age1 <- round(cvd$age/365, 0)
cvd$bmi <- round(cvd$weight / ((cvd$height/100)^2), 2)

# We see that there are incorrect entries from looking at weights.
```

```
# For example, some people weigh 20 lbs or less despite being middle-aged

# We are going to address these incorrect entries by calculating bmi and eliminating all
# data entries with less than 15 bmi value and greater than 50. These account for
# only ~1% of the data.

index <- which(cvd$bmi <= 15 | cvd$bmi >= 50)
cvd <- cvd[-index, ]

summary(cvd)
```

```
##        id             age            gender          height
##  Min.   :    0   Min.   :10798   Min.   :1.000   Min.   :120.0
##  1st Qu.:25010   1st Qu.:17656   1st Qu.:1.000   1st Qu.:159.0
##  Median :50024   Median :19701   Median :1.000   Median :165.0
##  Mean   :49978   Mean   :19464   Mean   :1.349   Mean   :164.4
##  3rd Qu.:74870   3rd Qu.:21324   3rd Qu.:2.000   3rd Qu.:170.0
##  Max.   :99999   Max.   :23713   Max.   :2.000   Max.   :207.0
##      weight           ap_hi           ap_lo         cholesterol
##  Min.   : 28.00   Min.   : 70.0   Min.   : 30.00   Min.   :1.000
##  1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 80.00   1st Qu.:1.000
##  Median : 72.00   Median :120.0   Median : 80.00   Median :1.000
##  Mean   : 73.97   Mean   :126.6   Mean   : 81.29   Mean   :1.364
##  3rd Qu.: 82.00   3rd Qu.:140.0   3rd Qu.: 90.00   3rd Qu.:1.000
##  Max.   :180.00   Max.   :240.0   Max.   :182.00   Max.   :3.000
##       gluc           smoke              alco              active
##  Min.   :1.000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:1.0000
##  Median :1.000   Median :0.00000   Median :0.00000   Median :1.0000
##  Mean   :1.225   Mean   :0.08809   Mean   :0.05337   Mean   :0.8034
##  3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##      cardio            age1            bmi
##  Min.   :0.0000   Min.   :30.00   Min.   :15.01
##  1st Qu.:0.0000   1st Qu.:48.00   1st Qu.:23.88
##  Median :0.0000   Median :54.00   Median :26.30
##  Mean   :0.4942   Mean   :53.33   Mean   :27.38
##  3rd Qu.:1.0000   3rd Qu.:58.00   3rd Qu.:30.12
##  Max.   :1.0000   Max.   :65.00   Max.   :49.98
```

Drawback

Due to my limited knowledge of abnormal ranges of blood pressure and BMI, I relied on online research to determine blood pressure and BMI values that are impossible or highly improbable to achieve. However, getting an expert's consultation would have allowed me to address outliers more accurately.

Next, I converted all categorical variables (**gender, cholesterol, gluc, smoke, alco, active, cardio**) from integer types to factor variables.

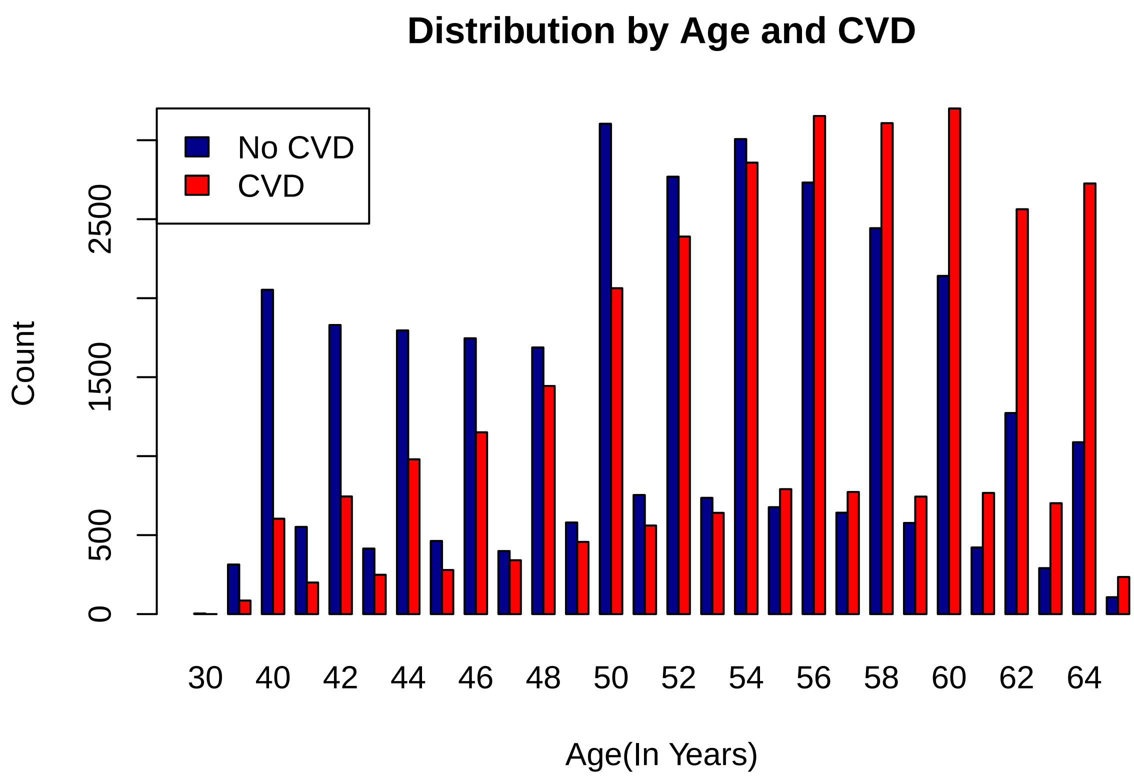
```
cvd_dup <- cvd

# Converting integer variables that are categorical into factor variables
cvd$gender <- as.factor(cvd$gender); levels(cvd$gender) <- c("female", "male")
```

```r
cvd$cholesterol <- as.factor(cvd$cholesterol)
levels(cvd$cholesterol) <- c("normal", "above normal", "well above normal")
cvd$gluc <- as.factor(cvd$gluc)
levels(cvd$gluc) <- c("normal", "above normal", "well above normal")
cvd$smoke <- as.factor(cvd$smoke); levels(cvd$smoke) <- c("Non-smoker", "Smoker")
cvd$alco <- as.factor(cvd$alco); levels(cvd$alco) <- c("Non-drinker", "Drinker")
cvd$active <- as.factor(cvd$active); levels(cvd$active) <- c("Not active", "Active")
cvd$cardio <- as.factor(cvd$cardio); levels(cvd$cardio) <- c("No CVD", "CVD")

str(cvd)
```

```
## 'data.frame':    68415 obs. of  15 variables:
##  $ id         : int  0 1 2 3 4 8 9 12 13 14 ...
##  $ age        : int  18393 20228 18857 17623 17474 21914 22113 22584 17668 19834 ...
##  $ gender     : Factor w/ 2 levels "female","male": 2 1 1 2 1 1 1 2 1 1 ...
##  $ height     : int  168 156 165 169 156 151 157 178 158 164 ...
##  $ weight     : num  62 85 64 82 56 67 93 95 71 68 ...
##  $ ap_hi      : int  110 140 130 150 100 120 130 130 110 110 ...
##  $ ap_lo      : int  80 90 70 100 60 80 80 90 70 60 ...
##  $ cholesterol: Factor w/ 3 levels "normal","above normal",..: 1 3 3 1 1 2 3 3 1 1 ...
##  $ gluc       : Factor w/ 3 levels "normal","above normal",..: 1 1 1 1 1 2 1 3 1 1 ...
##  $ smoke      : Factor w/ 2 levels "Non-smoker","Smoker": 1 1 1 1 1 1 1 1 1 1 ...
##  $ alco       : Factor w/ 2 levels "Non-drinker",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ active     : Factor w/ 2 levels "Not active","Active": 2 2 1 2 1 1 2 2 2 1 ...
##  $ cardio     : Factor w/ 2 levels "No CVD","CVD": 1 2 2 2 1 1 1 2 1 1 ...
##  $ age1       : num  50 55 52 48 48 60 61 62 48 54 ...
##  $ bmi        : num  22 34.9 23.5 28.7 23 ...
```

At what age does the event of CVD surpass not having CVD?

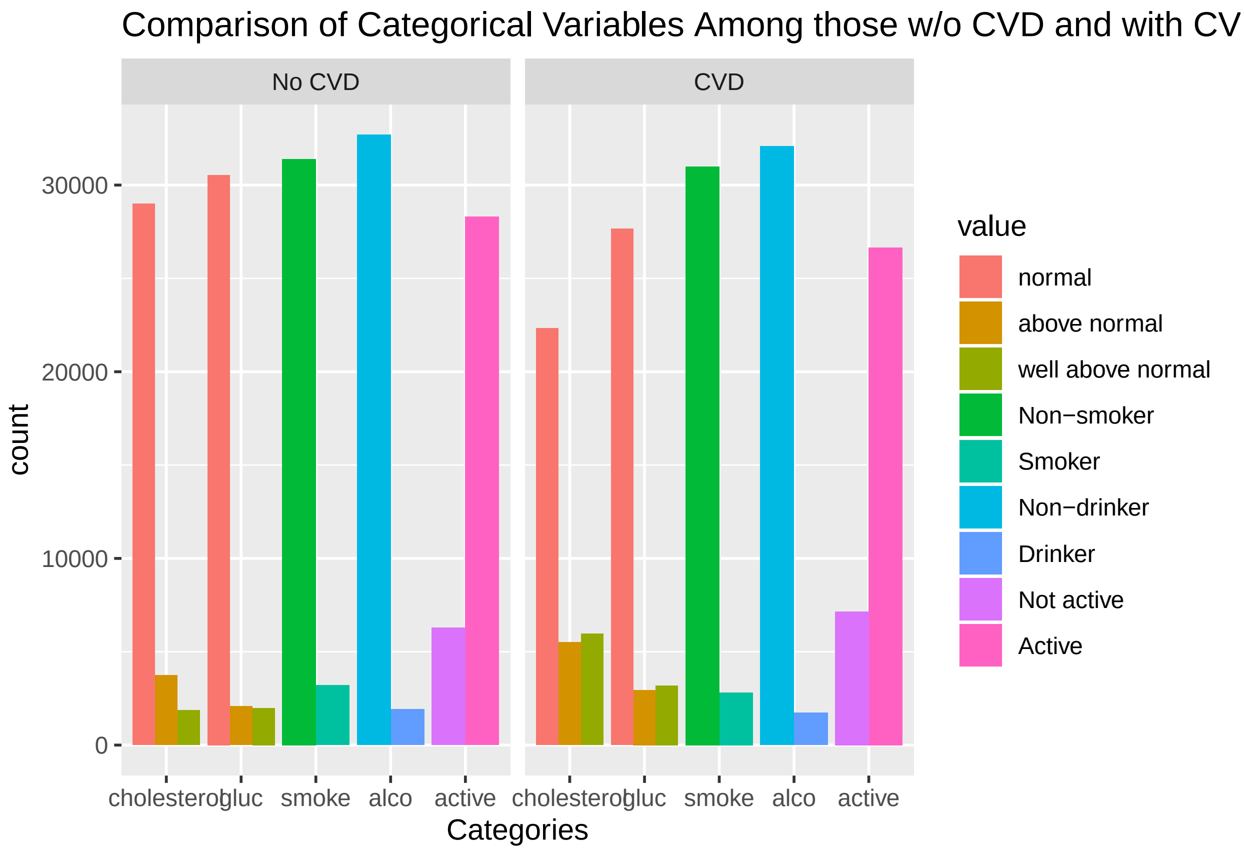
```r
count <- table(cvd$cardio, cvd$age1)
barplot(count, main="Distribution by Age and CVD",
        xlab="Age(In Years)", ylab="Count", col=c("darkblue","red"),
        legend = rownames(count), beside=TRUE, args.legend = list(x = "topleft"))
```



The presence of CVD surpasses the absence of CVD after the age of 54. Furthermore, the ratio of CVD:NoCVD quickly heads towards a 1:1 ratio in the mid 40s.

Comparing those with CVD and w/o CVD, which variables show greater risk/correlation? In other words, what variables are the most correlated with cardiovascular disease?

```
library(reshape)
library(ggplot2)
data_noCVD <- cvd[cvd$cardio == 'No CVD', ]; data_CVD <- cvd[cvd$cardio == 'CVD', ]
melt_noCVD <- melt(data_noCVD, id.vars = 'cardio', measure.vars = c('cholesterol', 'gluc', 'smoke', 'al
melt_CVD <- melt(data_CVD, id.vars = 'cardio', measure.vars = c('cholesterol', 'gluc', 'smoke', 'alco',
combine_melt <- rbind(melt_noCVD, melt_CVD)

ggplot(combine_melt,aes(factor(variable)))+geom_bar(aes(fill = value), position = "dodge")+
  ggtitle("Comparison of Categorical Variables Among those w/o CVD and with CVD")+labs(x = "Categories"
  facet_grid(. ~ cardio)
```

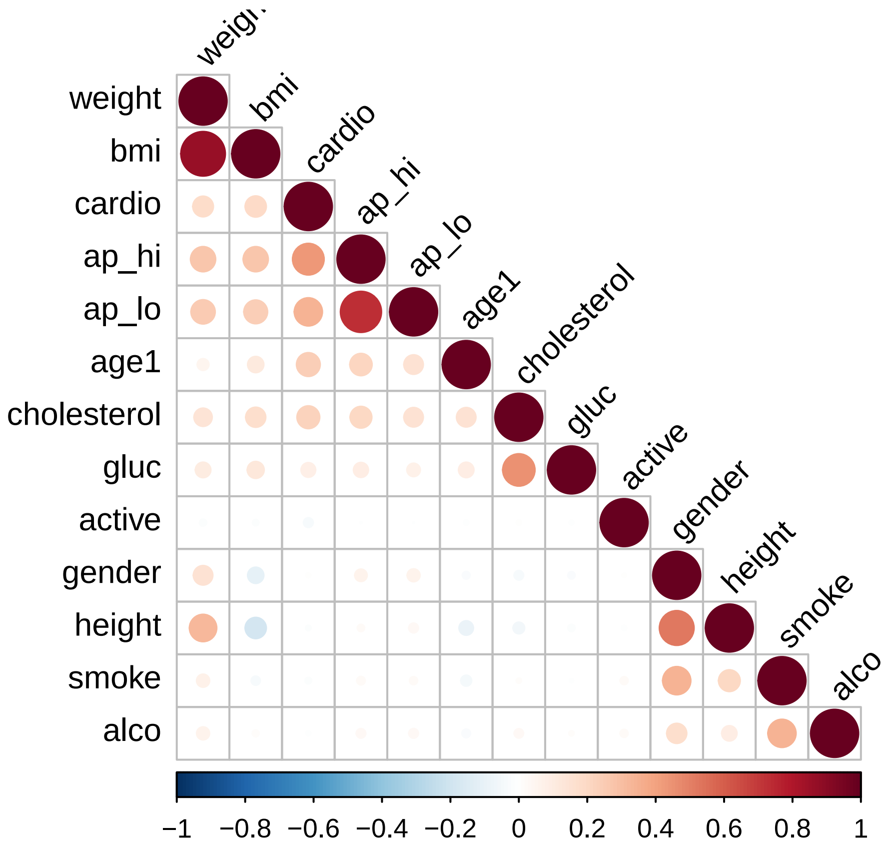


The ratio of (normal:above normal+well above normal) with regards to cholesterol levels drastically reduces from No CVD group to CVD group. The same can be said for glucose levels to a smaller extent. Smoking and alchol consumption seem to show no significant changes. Those with CVD seem to be less active on average.

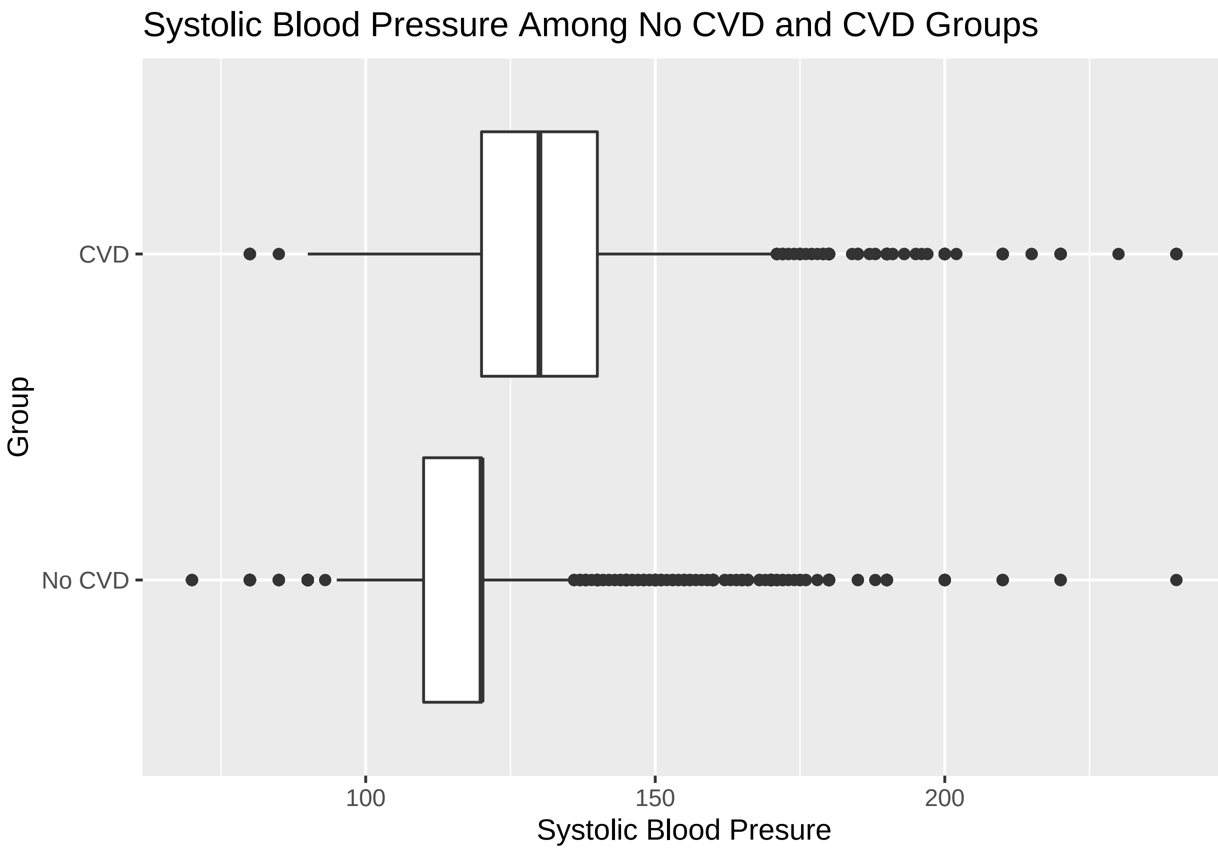
```
# Correlation matrix
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
source("http://www.sthda.com/upload/rquery_cormat.r")

corr_matrix <- rquery.cormat(cvd_dup[, c(-1, -2)])
```



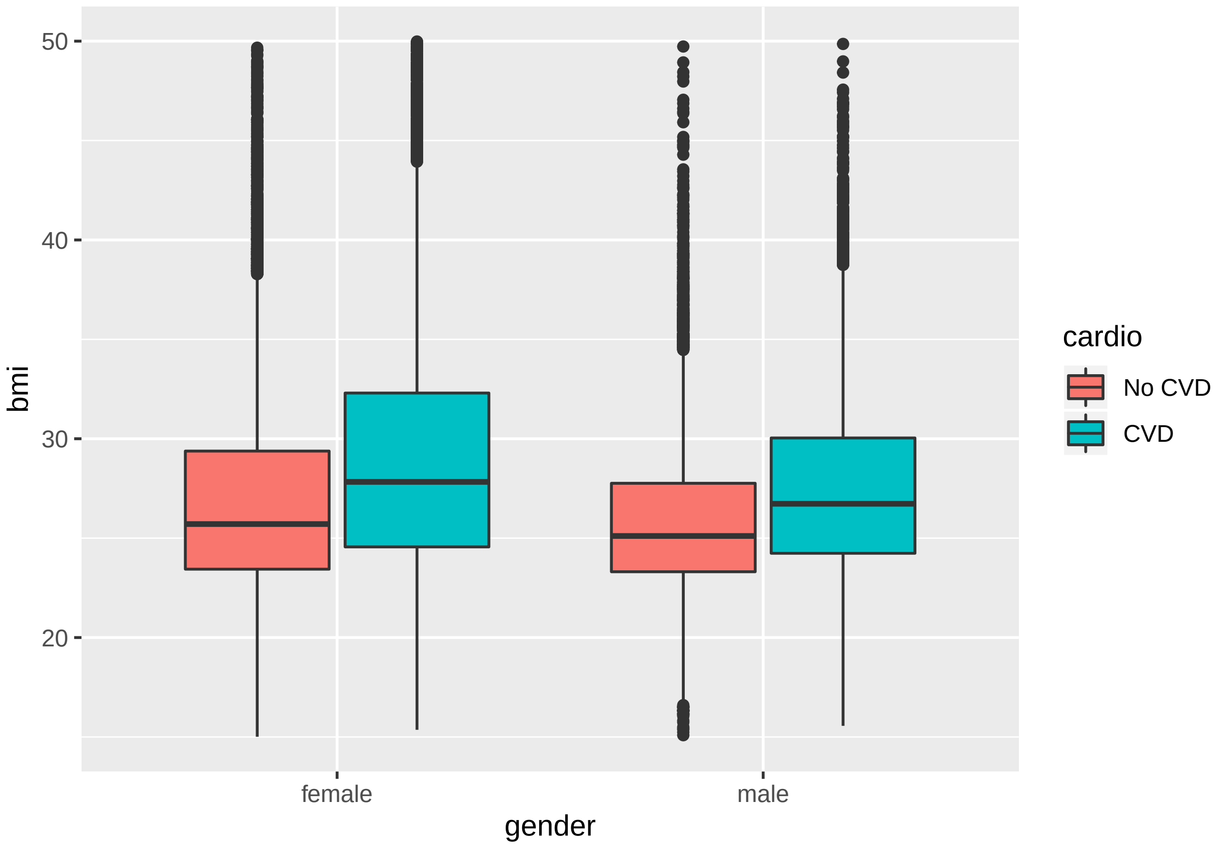

```r
ggplot(cvd, aes(x=cardio, y=ap_hi)) +
  geom_boxplot() + coord_flip()+
  ggtitle("Systolic Blood Pressure Among No CVD and CVD Groups")+
  labs(x="Group", y="Systolic Blood Presure")
```



The correlation heat matrix shows the strength of correlation or relationship between each variable. **ap__hi**, **ap__lo**, **age1**, **cholesterol**, **bmi**, and **weight** are the most correlated with CVD with correlation values of 0.43, 0.34, 0.24, 0.22, 0.19, and 0.18 respectively. However, no variables are strongly correlated with CVD.

Taking a closer look at the relationship between gender and bmi with CVD.

```
ggplot(cvd, aes(x=gender, y=bmi, fill=cardio)) +
  geom_boxplot()
```



Women tend to have on average a higher BMI value than men. Furthermore, the median of BMI for both genders is higher among the CVD group.

Classification

Three classification methods are used to predict cardiovascular disease.

1. Logistic Regression
2. XG Boosting
3. Random Foresting

I split up the dataset into a training set (80% of the data) and testing set (20% of the data). The models are trained on the training set and tested on the testing set.

Classification: Logistic Regression Model

```
# First off, we look at tables displaying CVD with all of the categorical variables
# We do this to see if there are a sufficient amount of reported data across all variables
# of each data. If there are an insufficient amount of reported data, that could cause
# an issue with finding a  model/line that best fits the data
xtabs(~ cardio + gender, data = cvd)
```

```
##          gender
## cardio    female  male
##   No CVD   22647 11956
##   CVD      21887 11925
```

```r
xtabs(~ cardio + cholesterol, data = cvd)
```

```
##         cholesterol
## cardio   normal above normal well above normal
##   No CVD  29006         3738              1859
##   CVD     22331         5516              5965
```

```r
xtabs(~ cardio + gluc, data = cvd)
```

```
##         gluc
## cardio   normal above normal well above normal
##   No CVD  30541         2077              1985
##   CVD     27667         2953              3192
```

```r
xtabs(~ cardio + smoke, data = cvd)
```

```
##         smoke
## cardio   Non-smoker Smoker
##   No CVD      31397   3206
##   CVD         30991   2821
```

```r
xtabs(~ cardio + alco, data = cvd)
```

```
##         alco
## cardio   Non-drinker Drinker
##   No CVD       32692    1911
##   CVD          32072    1740
```

```r
xtabs(~ cardio + active, data = cvd)
```

```
##         active
## cardio   Not active Active
##   No CVD       6291  28312
##   CVD          7160  26652
```

```r
# There are sufficient amount of reported data across all levels of each categorical variable

# We make a new data frame
cvd_new = cvd[, c(-1, -2, -4, -5)]

set.seed(1234)
ind <- sample(2, nrow(cvd_new), replace = T, prob = c(0.8, 0.2))
train <- cvd_new[ind==1, ]
test <- cvd_new[ind==2, ]

# Logistic Regression Classifier Model
logistic <- glm(cardio ~ ., data = train, family = "binomial")
# step(logistic, direction = "both") # Performing stepwise selection
summary(logistic)
```

```
##
## Call:
## glm(formula = cardio ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7970  -0.9203  -0.3338   0.9318   2.5400
##
```

```
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -11.428933   0.137907 -82.874  < 2e-16 ***
## gendermale                     0.027873   0.021994   1.267  0.20505
## ap_hi                          0.056583   0.001043  54.257  < 2e-16 ***
## ap_lo                          0.010642   0.001639   6.493 8.43e-11 ***
## cholesterolabove normal        0.369685   0.030562  12.096  < 2e-16 ***
## cholesterolwell above normal   1.112809   0.040063  27.776  < 2e-16 ***
## glucabove normal               0.038261   0.040650   0.941  0.34658
## glucwell above normal         -0.335338   0.044151  -7.595 3.07e-14 ***
## smokeSmoker                   -0.119121   0.038863  -3.065  0.00218 **
## alcoDrinker                   -0.237309   0.047698  -4.975 6.52e-07 ***
## activeActive                  -0.232421   0.024574  -9.458  < 2e-16 ***
## age1                           0.050786   0.001512  33.585  < 2e-16 ***
## bmi                            0.028109   0.002104  13.362  < 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: 75864  on 54727  degrees of freedom
## Residual deviance: 61303  on 54715  degrees of freedom
## AIC: 61329
##
## Number of Fisher Scoring iterations: 4
```

```
table(Predicted = ifelse(logistic$fitted.values < 0.50, "No CVD", "CVD"), Actual = train$cardio)
```

```
##          Actual
## Predicted No CVD   CVD
##    CVD      5825 18057
##    No CVD  21801  9045
```

```
(21805+18072)/nrow(train) # 0.7286265 - Training Classification Rate
```

```
## [1] 0.7286398
```

```
p <- predict(logistic, newdata = test, type="response")
table_class <- table(Predicted = ifelse(p < 0.50, "No CVD", "CVD"), Actual = test$cardio)
table_class
```

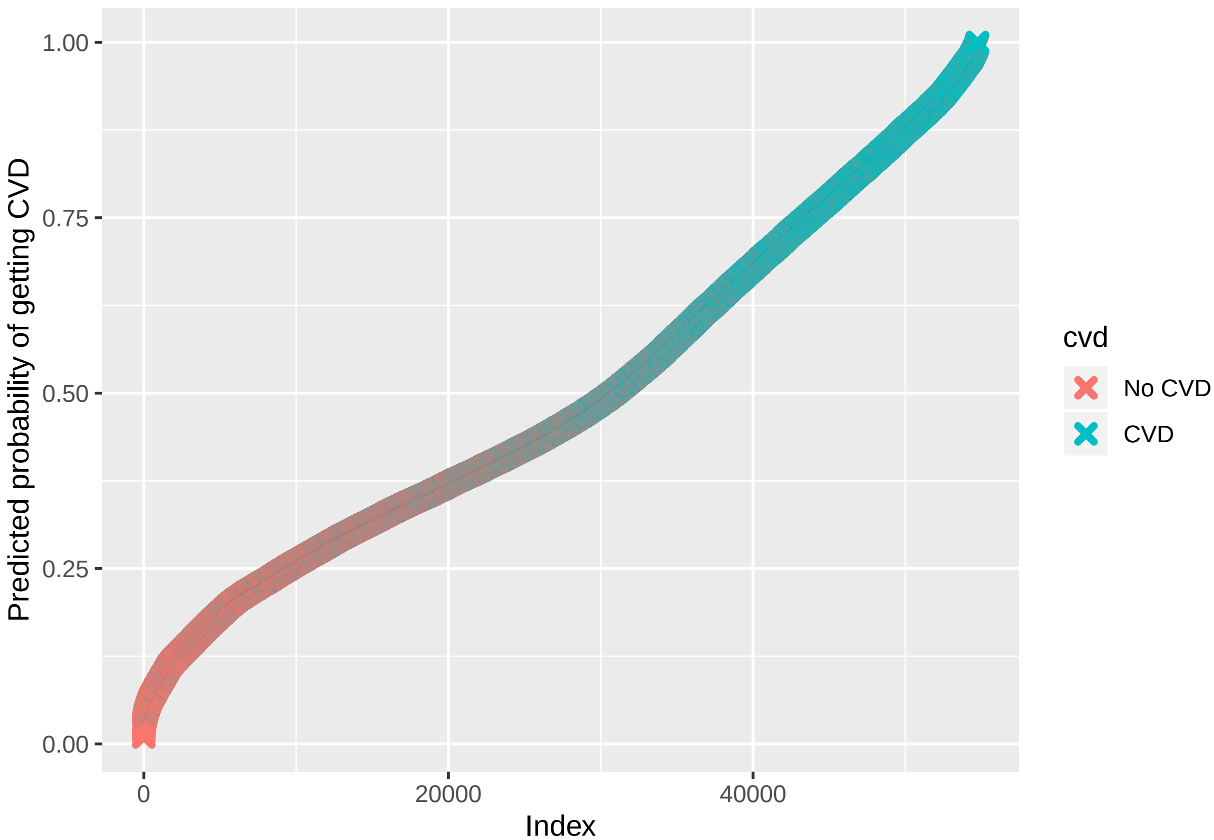
```
##          Actual
## Predicted No CVD  CVD
##    CVD      1490 4488
##    No CVD   5487 2222
```

```
correct <- (table_class[1,2] + table_class[2,1])/nrow(test)
cat("")
cat("Logistic Regression Model with one-hot encoding and stepwise feature selection yields
a testing successful classification rate of",correct)
```

```
## Logistic Regression Model with one-hot encoding and stepwise feature selection yields
## a testing successful classification rate of 0.7287937
```

```
predicted.data <- data.frame(
  probability.of.cvd = logistic$fitted.values, cvd = train$cardio)
```

```r
predicted.data <- predicted.data[
  order(predicted.data$probability.of.cvd, decreasing = FALSE),]

predicted.data$rank <- 1:nrow(predicted.data)


library(ggplot2)
#library(cowplot)
#theme_set(theme_cowplot())
ggplot(data = predicted.data, aes(x=rank, y=probability.of.cvd)) +
  geom_point(aes(color=cvd), alpha=1, shape=4, stroke=2) +
  xlab("Index") +
  ylab("Predicted probability of getting CVD")
```



Classification: XG Boosting

```r
# XG Boosting Classifier Algorithm with One-Hot Encoding
library(xgboost)
library(magrittr)
library(dplyr)
```

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

## The following object is masked from 'package:xgboost':
##
```

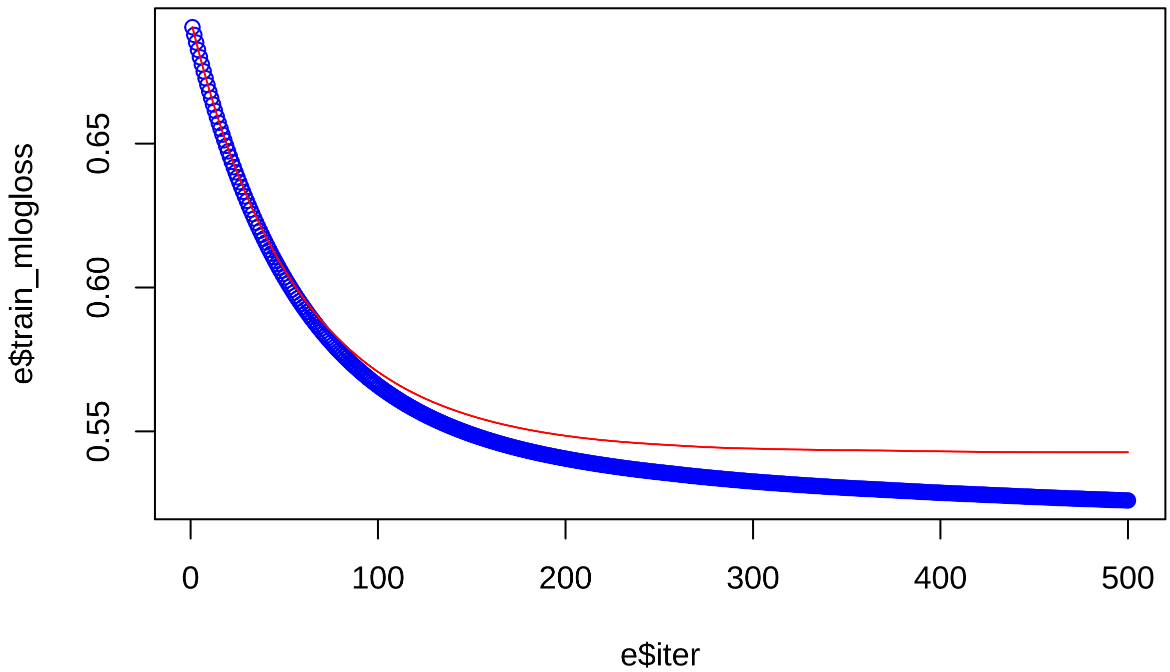
```
##     slice
## The following object is masked from 'package:reshape':
##
##     rename
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:reshape':
##
##     expand
```

```
train_xgb <- train; test_xgb <- test
train_xgb$cardio <- as.integer(train$cardio) - 1; test_xgb$cardio <- as.integer(test$cardio) - 1

# Create matrix - One-Hot Encoding for Factor variables
trainm <- sparse.model.matrix(cardio ~ .-1, data = train_xgb)
train_label <- train_xgb[, "cardio"]
train_matrix <- xgb.DMatrix(data = as.matrix(trainm), label = train_label)

testm <- sparse.model.matrix(cardio ~.-1, data = test_xgb)
test_label <- test_xgb[,"cardio"]
test_matrix <- xgb.DMatrix(data = as.matrix(testm), label = test_label)

# Parameters
nc <- length(unique(train_label))
xgb_params <- list("objective" = "multi:softprob",
                   "eval_metric" = "mlogloss",
                   "num_class" = nc)
watchlist <- list(train = train_matrix, test = test_matrix)

# XGB Model
bst_model <- xgb.train(params = xgb_params,
                       data = train_matrix,
                       nrounds = 500,
                       watchlist = watchlist,
                       eta = 0.01,
                       max.depth = 6,
                       seed = 333)
```

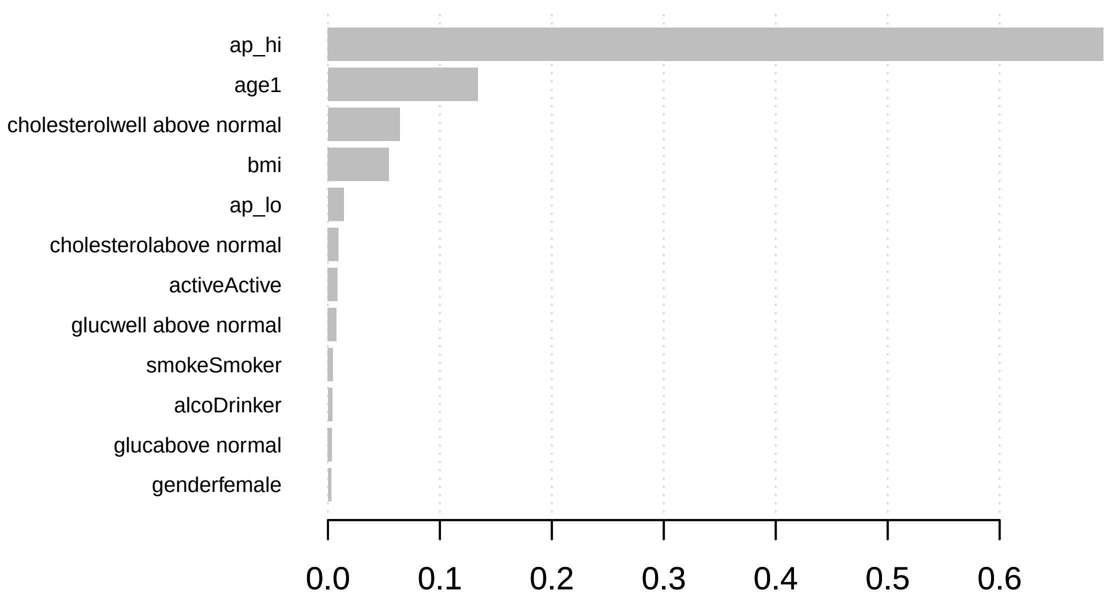
```
# Training & test error plot
e <- data.frame(bst_model$evaluation_log)
plot(e$iter, e$train_mlogloss, col = 'blue')
lines(e$iter, e$test_mlogloss, col = 'red')
```




```r
# Some overfitting is taking place

min(e$test_mlogloss)
```

```
## [1] 0.542762
```

```r
# e[e$test_mlogloss == 0.543199, ]

# Feature Importance
imp <- xgb.importance(colnames(train_matrix), model = bst_model)
xgb.plot.importance(imp)
```




```r
# Prediction and confusion matrix
p <- predict(bst_model, newdata = test_matrix)
pred <- matrix(p, nrow = nc, ncol = length(p)/nc) %>%
```

```
  t() %>%
  data.frame() %>%
  mutate(label = test_label, max_prob = max.col(., "last")-1)
table_class <- table(Prediction = pred$max_prob, Actual = pred$label)
table_class
```

```
##           Actual
## Prediction    0    1
##          0 5456 2132
##          1 1521 4578
```

```
correct <- (table_class[1,1] + table_class[2,2])/nrow(test)
cat("XG Boosting Classifier model yields a testing successful classification rate of",correct)
```

```
## XG Boosting Classifier model yields a testing successful classification rate of 0.7331044
```

Classification: Random Foresting

```
# Random Foresting
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(1234)
model <- randomForest(cardio ~ ., data = train, type="classification", ntree=300, proximity = FALSE, im
model
```

```
##
## Call:
##  randomForest(formula = cardio ~ ., data = train, type = "classification",      ntree = 300, proximi
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 26.99%
## Confusion matrix:
##        No CVD   CVD class.error
## No CVD  21133  6493   0.2350322
## CVD      8280 18822   0.3055125
```

```
model$importance
```

```
##                  No CVD           CVD MeanDecreaseAccuracy
## gender      0.004950906  0.0001628356         0.0025789812
```

```
## ap_hi       0.145981013  0.0828245386         0.1146910027
## ap_lo       0.028329075  0.0093676014         0.0189291769
## cholesterol 0.049530875  0.0053484362         0.0276456967
## gluc        0.011911138 -0.0048323436         0.0036175407
## smoke       0.002547974  0.0010124207         0.0017873791
## alco        0.002082770 -0.0004004061         0.0008526822
## active      0.001490671  0.0031445434         0.0023085173
## age1        0.035015923  0.0148949959         0.0250487679
## bmi         0.014730463  0.0006746863         0.0077678974
##             MeanDecreaseGini
## gender              324.6931
## ap_hi              4531.4686
## ap_lo              2154.4689
## cholesterol        1007.7366
## gluc                413.3847
## smoke               209.9023
## alco                173.1899
## active              275.8945
## age1               2451.6516
## bmi                3857.1128
```

```r
p = predict(model, newdata=test[,-9])
table_class <- table(Predicted = p, Actual = test$cardio)
table_class
```

```
##          Actual
## Predicted No CVD  CVD
##    No CVD   5304 2059
##    CVD      1673 4651
```

```r
correct <- (table_class[1,1] + table_class[2,2])/nrow(test)
cat("Random Forest model with 300 trees yields a testing successful classification rate of",correct)
```

```
## Random Forest model with 300 trees yields a testing successful classification rate of 0.7273325
```

The performance of Logistic Regression, XG Boosting, and Random Foresting is very similar with all 3 models yielding successful classification rates of around 73% on the testing set.