Final Project DSE6111

Adeline Casali

2023-12-05

# Section 1: What subset of predictors can be used for preventative health screening for heart disease?

**Loading data and packages**

heart\_df <- read.csv("Data/heart\_disease\_health\_indicators\_BRFSS2015.csv")  
library(corrplot)

## corrplot 0.92 loaded

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(e1071)  
library(class)  
library(tree)  
library(randomForest)

## randomForest 4.7-1.1  
## 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

library(gbm)

## Loaded gbm 2.1.8.1

library(kableExtra)

##   
## Attaching package: 'kableExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## group\_rows

**Data exploration and tidying**  
The only predictors that have a correlation stronger than 0.5 are physical health and general health with a correlation of 0.52.

# View the dataset and summary statistics  
head(heart\_df)

## HeartDiseaseorAttack HighBP HighChol CholCheck BMI Smoker Stroke Diabetes  
## 1 0 1 1 1 40 1 0 0  
## 2 0 0 0 0 25 1 0 0  
## 3 0 1 1 1 28 0 0 0  
## 4 0 1 0 1 27 0 0 0  
## 5 0 1 1 1 24 0 0 0  
## 6 0 1 1 1 25 1 0 0  
## PhysActivity Fruits Veggies HvyAlcoholConsump AnyHealthcare NoDocbcCost  
## 1 0 0 1 0 1 0  
## 2 1 0 0 0 0 1  
## 3 0 1 0 0 1 1  
## 4 1 1 1 0 1 0  
## 5 1 1 1 0 1 0  
## 6 1 1 1 0 1 0  
## GenHlth MentHlth PhysHlth DiffWalk Sex Age Education Income  
## 1 5 18 15 1 0 9 4 3  
## 2 3 0 0 0 0 7 6 1  
## 3 5 30 30 1 0 9 4 8  
## 4 2 0 0 0 0 11 3 6  
## 5 2 3 0 0 0 11 5 4  
## 6 2 0 2 0 1 10 6 8

summary(heart\_df)

## HeartDiseaseorAttack HighBP HighChol CholCheck   
## Min. :0.00000 Min. :0.000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:1.0000   
## Median :0.00000 Median :0.000 Median :0.0000 Median :1.0000   
## Mean :0.09419 Mean :0.429 Mean :0.4241 Mean :0.9627   
## 3rd Qu.:0.00000 3rd Qu.:1.000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.00000 Max. :1.000 Max. :1.0000 Max. :1.0000   
## BMI Smoker Stroke Diabetes   
## Min. :12.00 Min. :0.0000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:24.00 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000   
## Median :27.00 Median :0.0000 Median :0.00000 Median :0.0000   
## Mean :28.38 Mean :0.4432 Mean :0.04057 Mean :0.2969   
## 3rd Qu.:31.00 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:0.0000   
## Max. :98.00 Max. :1.0000 Max. :1.00000 Max. :2.0000   
## PhysActivity Fruits Veggies HvyAlcoholConsump  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.:0.0000   
## Median :1.0000 Median :1.0000 Median :1.0000 Median :0.0000   
## Mean :0.7565 Mean :0.6343 Mean :0.8114 Mean :0.0562   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## AnyHealthcare NoDocbcCost GenHlth MentHlth   
## Min. :0.0000 Min. :0.00000 Min. :1.000 Min. : 0.000   
## 1st Qu.:1.0000 1st Qu.:0.00000 1st Qu.:2.000 1st Qu.: 0.000   
## Median :1.0000 Median :0.00000 Median :2.000 Median : 0.000   
## Mean :0.9511 Mean :0.08418 Mean :2.511 Mean : 3.185   
## 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:3.000 3rd Qu.: 2.000   
## Max. :1.0000 Max. :1.00000 Max. :5.000 Max. :30.000   
## PhysHlth DiffWalk Sex Age   
## Min. : 0.000 Min. :0.0000 Min. :0.0000 Min. : 1.000   
## 1st Qu.: 0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 6.000   
## Median : 0.000 Median :0.0000 Median :0.0000 Median : 8.000   
## Mean : 4.242 Mean :0.1682 Mean :0.4403 Mean : 8.032   
## 3rd Qu.: 3.000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:10.000   
## Max. :30.000 Max. :1.0000 Max. :1.0000 Max. :13.000   
## Education Income   
## Min. :1.00 Min. :1.000   
## 1st Qu.:4.00 1st Qu.:5.000   
## Median :5.00 Median :7.000   
## Mean :5.05 Mean :6.054   
## 3rd Qu.:6.00 3rd Qu.:8.000   
## Max. :6.00 Max. :8.000

str(heart\_df)

## 'data.frame': 253680 obs. of 22 variables:  
## $ HeartDiseaseorAttack: num 0 0 0 0 0 0 0 0 1 0 ...  
## $ HighBP : num 1 0 1 1 1 1 1 1 1 0 ...  
## $ HighChol : num 1 0 1 0 1 1 0 1 1 0 ...  
## $ CholCheck : num 1 0 1 1 1 1 1 1 1 1 ...  
## $ BMI : num 40 25 28 27 24 25 30 25 30 24 ...  
## $ Smoker : num 1 1 0 0 0 1 1 1 1 0 ...  
## $ Stroke : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Diabetes : num 0 0 0 0 0 0 0 0 2 0 ...  
## $ PhysActivity : num 0 1 0 1 1 1 0 1 0 0 ...  
## $ Fruits : num 0 0 1 1 1 1 0 0 1 0 ...  
## $ Veggies : num 1 0 0 1 1 1 0 1 1 1 ...  
## $ HvyAlcoholConsump : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ AnyHealthcare : num 1 0 1 1 1 1 1 1 1 1 ...  
## $ NoDocbcCost : num 0 1 1 0 0 0 0 0 0 0 ...  
## $ GenHlth : num 5 3 5 2 2 2 3 3 5 2 ...  
## $ MentHlth : num 18 0 30 0 3 0 0 0 30 0 ...  
## $ PhysHlth : num 15 0 30 0 0 2 14 0 30 0 ...  
## $ DiffWalk : num 1 0 1 0 0 0 0 1 1 0 ...  
## $ Sex : num 0 0 0 0 0 1 0 0 0 1 ...  
## $ Age : num 9 7 9 11 11 10 9 11 9 8 ...  
## $ Education : num 4 6 4 3 5 6 6 4 5 4 ...  
## $ Income : num 3 1 8 6 4 8 7 4 1 3 ...

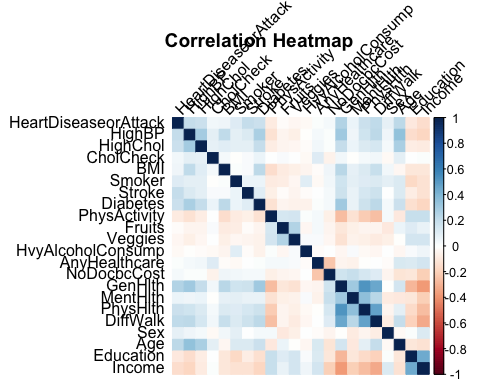
dim(heart\_df)

## [1] 253680 22

names(heart\_df)

## [1] "HeartDiseaseorAttack" "HighBP" "HighChol"   
## [4] "CholCheck" "BMI" "Smoker"   
## [7] "Stroke" "Diabetes" "PhysActivity"   
## [10] "Fruits" "Veggies" "HvyAlcoholConsump"   
## [13] "AnyHealthcare" "NoDocbcCost" "GenHlth"   
## [16] "MentHlth" "PhysHlth" "DiffWalk"   
## [19] "Sex" "Age" "Education"   
## [22] "Income"

# Check for correlated variables  
cor\_matrix <- cor(heart\_df)  
corrplot(cor\_matrix, method = "color", tl.col = "black", tl.srt = 45)  
title("Correlation Heatmap")



cor\_matrix <- cor(heart\_df)  
cor\_matrix\_filtered <- cor\_matrix  
cor\_matrix\_filtered[abs(cor\_matrix) <= 0.5] <- NA  
print(cor\_matrix\_filtered)

## HeartDiseaseorAttack HighBP HighChol CholCheck BMI Smoker  
## HeartDiseaseorAttack 1 NA NA NA NA NA  
## HighBP NA 1 NA NA NA NA  
## HighChol NA NA 1 NA NA NA  
## CholCheck NA NA NA 1 NA NA  
## BMI NA NA NA NA 1 NA  
## Smoker NA NA NA NA NA 1  
## Stroke NA NA NA NA NA NA  
## Diabetes NA NA NA NA NA NA  
## PhysActivity NA NA NA NA NA NA  
## Fruits NA NA NA NA NA NA  
## Veggies NA NA NA NA NA NA  
## HvyAlcoholConsump NA NA NA NA NA NA  
## AnyHealthcare NA NA NA NA NA NA  
## NoDocbcCost NA NA NA NA NA NA  
## GenHlth NA NA NA NA NA NA  
## MentHlth NA NA NA NA NA NA  
## PhysHlth NA NA NA NA NA NA  
## DiffWalk NA NA NA NA NA NA  
## Sex NA NA NA NA NA NA  
## Age NA NA NA NA NA NA  
## Education NA NA NA NA NA NA  
## Income NA NA NA NA NA NA  
## Stroke Diabetes PhysActivity Fruits Veggies  
## HeartDiseaseorAttack NA NA NA NA NA  
## HighBP NA NA NA NA NA  
## HighChol NA NA NA NA NA  
## CholCheck NA NA NA NA NA  
## BMI NA NA NA NA NA  
## Smoker NA NA NA NA NA  
## Stroke 1 NA NA NA NA  
## Diabetes NA 1 NA NA NA  
## PhysActivity NA NA 1 NA NA  
## Fruits NA NA NA 1 NA  
## Veggies NA NA NA NA 1  
## HvyAlcoholConsump NA NA NA NA NA  
## AnyHealthcare NA NA NA NA NA  
## NoDocbcCost NA NA NA NA NA  
## GenHlth NA NA NA NA NA  
## MentHlth NA NA NA NA NA  
## PhysHlth NA NA NA NA NA  
## DiffWalk NA NA NA NA NA  
## Sex NA NA NA NA NA  
## Age NA NA NA NA NA  
## Education NA NA NA NA NA  
## Income NA NA NA NA NA  
## HvyAlcoholConsump AnyHealthcare NoDocbcCost GenHlth  
## HeartDiseaseorAttack NA NA NA NA  
## HighBP NA NA NA NA  
## HighChol NA NA NA NA  
## CholCheck NA NA NA NA  
## BMI NA NA NA NA  
## Smoker NA NA NA NA  
## Stroke NA NA NA NA  
## Diabetes NA NA NA NA  
## PhysActivity NA NA NA NA  
## Fruits NA NA NA NA  
## Veggies NA NA NA NA  
## HvyAlcoholConsump 1 NA NA NA  
## AnyHealthcare NA 1 NA NA  
## NoDocbcCost NA NA 1 NA  
## GenHlth NA NA NA 1.0000000  
## MentHlth NA NA NA NA  
## PhysHlth NA NA NA 0.5243636  
## DiffWalk NA NA NA NA  
## Sex NA NA NA NA  
## Age NA NA NA NA  
## Education NA NA NA NA  
## Income NA NA NA NA  
## MentHlth PhysHlth DiffWalk Sex Age Education Income  
## HeartDiseaseorAttack NA NA NA NA NA NA NA  
## HighBP NA NA NA NA NA NA NA  
## HighChol NA NA NA NA NA NA NA  
## CholCheck NA NA NA NA NA NA NA  
## BMI NA NA NA NA NA NA NA  
## Smoker NA NA NA NA NA NA NA  
## Stroke NA NA NA NA NA NA NA  
## Diabetes NA NA NA NA NA NA NA  
## PhysActivity NA NA NA NA NA NA NA  
## Fruits NA NA NA NA NA NA NA  
## Veggies NA NA NA NA NA NA NA  
## HvyAlcoholConsump NA NA NA NA NA NA NA  
## AnyHealthcare NA NA NA NA NA NA NA  
## NoDocbcCost NA NA NA NA NA NA NA  
## GenHlth NA 0.5243636 NA NA NA NA NA  
## MentHlth 1 NA NA NA NA NA NA  
## PhysHlth NA 1.0000000 NA NA NA NA NA  
## DiffWalk NA NA 1 NA NA NA NA  
## Sex NA NA NA 1 NA NA NA  
## Age NA NA NA NA 1 NA NA  
## Education NA NA NA NA NA 1 NA  
## Income NA NA NA NA NA NA 1

# Convert HeartDiseaseorAttack to factor  
heart\_df$HeartDiseaseorAttack <- as.factor(heart\_df$HeartDiseaseorAttack)

**Logistic Regression**  
The linear model with all predictors shows that the variables HighBP, HighChol, CholCheck, Smoker, Stroke, Diabetes, HvyAlcoholConsump, NoDocbcCost, GenHlth, DiffWalk, Sex, Age, and Income were the most significant predictors, followed by PhysActivity, Veggies, and MentHlth. The model created with all of the significant predictors has an accuracy of 89.78%, and the model created with only the most significant predictors has the same accuracy of 89.78%.

# Null model  
heart\_lm\_null <- glm(HeartDiseaseorAttack ~ HeartDiseaseorAttack, data = heart\_df, family = binomial)

## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on  
## the right-hand side and was dropped

## Warning in model.matrix.default(mt, mf, contrasts): problem with term 1 in  
## model.matrix: no columns are assigned

summary(heart\_lm\_null)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ HeartDiseaseorAttack, family = binomial,   
## data = heart\_df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.263567 0.006797 -333 <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: 158355 on 253679 degrees of freedom  
## Residual deviance: 158355 on 253679 degrees of freedom  
## AIC: 158357  
##   
## Number of Fisher Scoring iterations: 5

# Model with all predictors  
heart\_lm <- glm(HeartDiseaseorAttack ~ ., data = heart\_df, family = binomial)  
summary(heart\_lm)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ ., family = binomial, data = heart\_df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.9124089 0.1028054 -76.965 < 2e-16 \*\*\*  
## HighBP 0.5245081 0.0177520 29.546 < 2e-16 \*\*\*  
## HighChol 0.6112677 0.0164496 37.160 < 2e-16 \*\*\*  
## CholCheck 0.5248111 0.0662510 7.922 2.35e-15 \*\*\*  
## BMI 0.0009744 0.0012122 0.804 0.4215   
## Smoker 0.3629524 0.0157323 23.071 < 2e-16 \*\*\*  
## Stroke 0.9783443 0.0244338 40.041 < 2e-16 \*\*\*  
## Diabetes 0.1465123 0.0089720 16.330 < 2e-16 \*\*\*  
## PhysActivity 0.0398975 0.0171991 2.320 0.0204 \*   
## Fruits 0.0060203 0.0163274 0.369 0.7123   
## Veggies 0.0426327 0.0189359 2.251 0.0244 \*   
## HvyAlcoholConsump -0.2940714 0.0392877 -7.485 7.15e-14 \*\*\*  
## AnyHealthcare -0.0070009 0.0412820 -0.170 0.8653   
## NoDocbcCost 0.2528463 0.0268896 9.403 < 2e-16 \*\*\*  
## GenHlth 0.4907058 0.0095105 51.596 < 2e-16 \*\*\*  
## MentHlth 0.0024628 0.0009778 2.519 0.0118 \*   
## PhysHlth 0.0010542 0.0008766 1.202 0.2292   
## DiffWalk 0.2947780 0.0193855 15.206 < 2e-16 \*\*\*  
## Sex 0.7611811 0.0160326 47.477 < 2e-16 \*\*\*  
## Age 0.2556493 0.0036439 70.158 < 2e-16 \*\*\*  
## Education 0.0110314 0.0081696 1.350 0.1769   
## Income -0.0431551 0.0042476 -10.160 < 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: 158355 on 253679 degrees of freedom  
## Residual deviance: 120933 on 253658 degrees of freedom  
## AIC: 120977  
##   
## Number of Fisher Scoring iterations: 6

# Create a training and testing set  
set.seed(123)  
heart\_train <- heart\_df %>%   
 sample\_frac(0.7)  
heart\_test <- anti\_join(heart\_df, heart\_train)

## Joining with `by = join\_by(HeartDiseaseorAttack, HighBP, HighChol, CholCheck,  
## BMI, Smoker, Stroke, Diabetes, PhysActivity, Fruits, Veggies,  
## HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, GenHlth, MentHlth, PhysHlth,  
## DiffWalk, Sex, Age, Education, Income)`

# Model with significant predictors  
heart\_lm\_sig <- glm(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train, family = binomial)  
summary(heart\_lm\_sig)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck +   
## Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost +   
## GenHlth + DiffWalk + Sex + Age + Income + PhysActivity +   
## Veggies + MentHlth, family = binomial, data = heart\_train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.835845 0.101326 -77.333 < 2e-16 \*\*\*  
## HighBP 0.526391 0.020942 25.136 < 2e-16 \*\*\*  
## HighChol 0.623145 0.019618 31.764 < 2e-16 \*\*\*  
## CholCheck 0.543923 0.079393 6.851 7.33e-12 \*\*\*  
## Smoker 0.354107 0.018672 18.965 < 2e-16 \*\*\*  
## Stroke 0.946965 0.029093 32.549 < 2e-16 \*\*\*  
## Diabetes 0.142707 0.010519 13.567 < 2e-16 \*\*\*  
## HvyAlcoholConsump -0.284262 0.046573 -6.104 1.04e-09 \*\*\*  
## NoDocbcCost 0.234995 0.031695 7.414 1.22e-13 \*\*\*  
## GenHlth 0.495146 0.010365 47.773 < 2e-16 \*\*\*  
## DiffWalk 0.317315 0.022180 14.306 < 2e-16 \*\*\*  
## Sex 0.762415 0.019058 40.006 < 2e-16 \*\*\*  
## Age 0.252476 0.004166 60.610 < 2e-16 \*\*\*  
## Income -0.041475 0.004690 -8.843 < 2e-16 \*\*\*  
## PhysActivity 0.038610 0.020277 1.904 0.0569 .   
## Veggies 0.038037 0.021958 1.732 0.0832 .   
## MentHlth 0.003395 0.001132 3.000 0.0027 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 111242 on 177575 degrees of freedom  
## Residual deviance: 84942 on 177559 degrees of freedom  
## AIC: 84976  
##   
## Number of Fisher Scoring iterations: 6

# Test the model and calculate accuracy  
predict\_lm <- predict(heart\_lm\_sig, newdata = heart\_test)  
binary\_predict\_lm <- ifelse(predict\_lm > 0.5, 1, 0)  
results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,   
 Predicted = binary\_predict\_lm  
)  
results$Correct <- results$Actual == results$Predicted  
confusion\_matrix\_lm <- table(Predicted = results$Predicted, Actual = results$Actual)  
print(confusion\_matrix\_lm)

## Actual  
## Predicted 0 1  
## 0 59853 6623  
## 1 234 391

accuracy\_lm <- (59853 + 391) / (59853 + 6623 + 234 + 391)  
error\_lm <- 1 - accuracy\_lm  
cat("Accuracy:", accuracy\_lm, "\n")

## Accuracy: 0.8978108

cat("Error Rate:", error\_lm, "\n")

## Error Rate: 0.1021892

# Model with only the most significant predictors  
heart\_lm\_sig2 <- glm(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income, data = heart\_train, family = binomial)  
summary(heart\_lm\_sig2)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck +   
## Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost +   
## GenHlth + DiffWalk + Sex + Age + Income, family = binomial,   
## data = heart\_train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.758065 0.098684 -78.615 < 2e-16 \*\*\*  
## HighBP 0.524694 0.020937 25.060 < 2e-16 \*\*\*  
## HighChol 0.625225 0.019603 31.894 < 2e-16 \*\*\*  
## CholCheck 0.546769 0.079366 6.889 5.61e-12 \*\*\*  
## Smoker 0.355084 0.018655 19.034 < 2e-16 \*\*\*  
## Stroke 0.947827 0.029083 32.590 < 2e-16 \*\*\*  
## Diabetes 0.141732 0.010513 13.482 < 2e-16 \*\*\*  
## HvyAlcoholConsump -0.280701 0.046548 -6.030 1.64e-09 \*\*\*  
## NoDocbcCost 0.244870 0.031503 7.773 7.67e-15 \*\*\*  
## GenHlth 0.498179 0.010072 49.462 < 2e-16 \*\*\*  
## DiffWalk 0.317142 0.021823 14.532 < 2e-16 \*\*\*  
## Sex 0.758174 0.018949 40.012 < 2e-16 \*\*\*  
## Age 0.250268 0.004088 61.223 < 2e-16 \*\*\*  
## Income -0.041128 0.004639 -8.867 < 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: 111242 on 177575 degrees of freedom  
## Residual deviance: 84958 on 177562 degrees of freedom  
## AIC: 84986  
##   
## Number of Fisher Scoring iterations: 6

# Test the second model and calculate accuracy  
predict\_lm2 <- predict(heart\_lm\_sig2, newdata = heart\_test)  
binary\_predict\_lm2 <- ifelse(predict\_lm2 > 0.5, 1, 0)  
results2 <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,   
 Predicted = binary\_predict\_lm2  
)  
results2$Correct <- results2$Actual == results2$Predicted  
confusion\_matrix\_lm2 <- table(Predicted = results2$Predicted, Actual = results2$Actual)  
print(confusion\_matrix\_lm2)

## Actual  
## Predicted 0 1  
## 0 59847 6618  
## 1 240 396

accuracy\_lm2 <- (59847 + 396) / (59847 + 6618 + 240 + 396)  
error\_lm2 <- 1 - accuracy\_lm2  
cat("Accuracy:", accuracy\_lm2, "\n")

## Accuracy: 0.8977959

cat("Error Rate:", error\_lm2, "\n")

## Error Rate: 0.1022041

**LDA**  
The LDA model containing all significant predictors has an accuracy of 89.14%.

# Model with significant predictors  
heart\_lda <- lda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train)  
heart\_lda

## Call:  
## lda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker +   
## Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth +   
## DiffWalk + Sex + Age + Income + PhysActivity + Veggies +   
## MentHlth, data = heart\_train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.90532504 0.09467496   
##   
## Group means:  
## HighBP HighChol CholCheck Smoker Stroke Diabetes  
## 0 0.3949205 0.3955799 0.9595059 0.4254684 0.02793536 0.2554614  
## 1 0.7501784 0.7031882 0.9888175 0.6190221 0.16404949 0.6846895  
## HvyAlcoholConsump NoDocbcCost GenHlth DiffWalk Sex Age Income  
## 0 0.05785499 0.08118733 2.420691 0.1427496 0.4269177 7.813808 6.147091  
## 1 0.03598620 0.10908875 3.372175 0.4182132 0.5739353 10.118011 5.143053  
## PhysActivity Veggies MentHlth  
## 0 0.7689532 0.8162400 3.038952  
## 1 0.6411492 0.7635023 4.732512  
##   
## Coefficients of linear discriminants:  
## LD1  
## HighBP 0.2934865419  
## HighChol 0.3740542652  
## CholCheck 0.1465449354  
## Smoker 0.2194045131  
## Stroke 1.7900323795  
## Diabetes 0.2102071299  
## HvyAlcoholConsump -0.1675486525  
## NoDocbcCost 0.0514054892  
## GenHlth 0.3356617796  
## DiffWalk 0.5344940710  
## Sex 0.5037227569  
## Age 0.1109019681  
## Income -0.0317635706  
## PhysActivity 0.0456651232  
## Veggies 0.0465340124  
## MentHlth 0.0009576617

# Test the model and calculate accuracy  
lda\_predictions <- predict(heart\_lda, newdata = heart\_test)$class  
lda\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = lda\_predictions  
)  
lda\_results$Correct <- lda\_results$Actual == lda\_results$Predicted  
lda\_confusion\_matrix <- table(Predicted = lda\_results$Predicted, Actual = lda\_results$Actual)  
print(lda\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 58497 5694  
## 1 1590 1320

accuracy\_lda <- (58497 + 1320) / (58497 + 5694 + 1590 + 1320)  
error\_lda <- 1 - accuracy\_lda  
cat("Accuracy:", accuracy\_lda, "\n")

## Accuracy: 0.8914472

cat("Error Rate:", error\_lda, "\n")

## Error Rate: 0.1085528

**QDA**  
The QDA model containing all significant predictors has an accuracy of 83.25%.

# Model with significant predictors  
heart\_qda <- qda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train)  
heart\_qda

## Call:  
## qda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker +   
## Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth +   
## DiffWalk + Sex + Age + Income + PhysActivity + Veggies +   
## MentHlth, data = heart\_train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.90532504 0.09467496   
##   
## Group means:  
## HighBP HighChol CholCheck Smoker Stroke Diabetes  
## 0 0.3949205 0.3955799 0.9595059 0.4254684 0.02793536 0.2554614  
## 1 0.7501784 0.7031882 0.9888175 0.6190221 0.16404949 0.6846895  
## HvyAlcoholConsump NoDocbcCost GenHlth DiffWalk Sex Age Income  
## 0 0.05785499 0.08118733 2.420691 0.1427496 0.4269177 7.813808 6.147091  
## 1 0.03598620 0.10908875 3.372175 0.4182132 0.5739353 10.118011 5.143053  
## PhysActivity Veggies MentHlth  
## 0 0.7689532 0.8162400 3.038952  
## 1 0.6411492 0.7635023 4.732512

# Test the model and calculate accuracy  
qda\_predictions <- predict(heart\_qda, newdata = heart\_test)$class  
qda\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = qda\_predictions  
)  
qda\_results$Correct <- qda\_results$Actual == qda\_results$Predicted  
qda\_confusion\_matrix <- table(Predicted = qda\_results$Predicted, Actual = qda\_results$Actual)  
print(qda\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 52651 3802  
## 1 7436 3212

accuracy\_qda <- (52651 + 3212) / (52651 + 3802 + 7436 + 3212)  
error\_qda <- 1 - accuracy\_qda  
cat("Accuracy:", accuracy\_qda, "\n")

## Accuracy: 0.8325211

cat("Error Rate:", error\_qda, "\n")

## Error Rate: 0.1674789

**Naive Bayes**  
The Naive Bayes model containing all significant predictors has an accuracy of 80.39%.

# Model with significant predictors  
heart\_nb <- naiveBayes(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train)  
heart\_nb

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.90532504 0.09467496   
##   
## Conditional probabilities:  
## HighBP  
## Y [,1] [,2]  
## 0 0.3949205 0.4888351  
## 1 0.7501784 0.4329225  
##   
## HighChol  
## Y [,1] [,2]  
## 0 0.3955799 0.4889764  
## 1 0.7031882 0.4568665  
##   
## CholCheck  
## Y [,1] [,2]  
## 0 0.9595059 0.1971157  
## 1 0.9888175 0.1051575  
##   
## Smoker  
## Y [,1] [,2]  
## 0 0.4254684 0.4944154  
## 1 0.6190221 0.4856416  
##   
## Stroke  
## Y [,1] [,2]  
## 0 0.02793536 0.1647882  
## 1 0.16404949 0.3703315  
##   
## Diabetes  
## Y [,1] [,2]  
## 0 0.2554614 0.6548011  
## 1 0.6846895 0.9338205  
##   
## HvyAlcoholConsump  
## Y [,1] [,2]  
## 0 0.05785499 0.2334698  
## 1 0.03598620 0.1862613  
##   
## NoDocbcCost  
## Y [,1] [,2]  
## 0 0.08118733 0.2731234  
## 1 0.10908875 0.3117598  
##   
## GenHlth  
## Y [,1] [,2]  
## 0 2.420691 1.025722  
## 1 3.372175 1.085493  
##   
## DiffWalk  
## Y [,1] [,2]  
## 0 0.1427496 0.3498184  
## 1 0.4182132 0.4932802  
##   
## Sex  
## Y [,1] [,2]  
## 0 0.4269177 0.4946317  
## 1 0.5739353 0.4945181  
##   
## Age  
## Y [,1] [,2]  
## 0 7.813808 3.046982  
## 1 10.118011 2.237037  
##   
## Income  
## Y [,1] [,2]  
## 0 6.147091 2.034289  
## 1 5.143053 2.207500  
##   
## PhysActivity  
## Y [,1] [,2]  
## 0 0.7689532 0.4215036  
## 1 0.6411492 0.4796776  
##   
## Veggies  
## Y [,1] [,2]  
## 0 0.8162400 0.3872896  
## 1 0.7635023 0.4249439  
##   
## MentHlth  
## Y [,1] [,2]  
## 0 3.038952 7.196951  
## 1 4.732512 9.251938

# Test the model and calculate accuracy  
nb\_predictions <- predict(heart\_nb, newdata = heart\_test)  
nb\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = nb\_predictions  
)  
nb\_results$Correct <- nb\_results$Actual == nb\_results$Predicted  
nb\_confusion\_matrix <- table(Predicted = nb\_results$Predicted, Actual = nb\_results$Actual)  
print(nb\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 50161 3234  
## 1 9926 3780

accuracy\_nb <- (50161 + 3780) / (50161 + 3234 + 9926 + 3780)  
error\_nb <- 1 - accuracy\_nb  
cat("Accuracy:", accuracy\_nb, "\n")

## Accuracy: 0.8038777

cat("Error Rate:", error\_nb, "\n")

## Error Rate: 0.1961223

**KNN (K-Nearest Neighbors)**  
The KNN model containing all significant predictors and K=3 has an accuracy of 98.13%.

# Model with all significant predictors and K = 3  
predictors <- c(  
 "HighBP", "HighChol", "CholCheck", "Smoker", "Stroke", "Diabetes",  
 "HvyAlcoholConsump", "NoDocbcCost", "GenHlth", "DiffWalk",  
 "Sex", "Age", "Income", "PhysActivity", "Veggies", "MentHlth"  
)  
train\_data <- heart\_train[, c(predictors, "HeartDiseaseorAttack")]  
test\_data <- heart\_test[, c(predictors, "HeartDiseaseorAttack")]  
knn\_predictions <- knn(  
 train = train\_data[, -length(predictors)],  
 test = test\_data[, -length(predictors)],  
 cl = train\_data$HeartDiseaseorAttack,  
 k = 3  
)  
knn\_results <- data.frame(  
 Actual = test\_data$HeartDiseaseorAttack,  
 Predicted = knn\_predictions  
)  
knn\_results$Correct <- knn\_results$Actual == knn\_results$Predicted  
knn\_confusion\_matrix <- table(Predicted = knn\_results$Predicted, Actual = knn\_results$Actual)  
print(knn\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 60081 1245  
## 1 6 5769

accuracy\_knn <- (60082 + 5763) / (60082 + 1251 + 5 + 5763)  
error\_knn <- 1 - accuracy\_knn  
cat("Accuracy:", accuracy\_knn, "\n")

## Accuracy: 0.9812819

cat("Error Rate:", error\_knn, "\n")

## Error Rate: 0.01871805

# Model with all significant predictors and K = 5  
knn\_predictions2 <- knn(  
 train = train\_data[, -length(predictors)],  
 test = test\_data[, -length(predictors)],  
 cl = train\_data$HeartDiseaseorAttack,  
 k = 5  
)  
knn\_results2 <- data.frame(  
 Actual = test\_data$HeartDiseaseorAttack,  
 Predicted = knn\_predictions2  
)  
knn\_results2$Correct <- knn\_results2$Actual == knn\_results2$Predicted  
knn\_confusion\_matrix2 <- table(Predicted = knn\_results2$Predicted, Actual = knn\_results2$Actual)  
print(knn\_confusion\_matrix2)

## Actual  
## Predicted 0 1  
## 0 60086 1419  
## 1 1 5595

accuracy\_knn2 <- (60085 + 5606) / (60085 + 1408 + 2 + 5606)  
error\_knn2 <- 1 - accuracy\_knn2  
cat("Accuracy:", accuracy\_knn2, "\n")

## Accuracy: 0.9789869

cat("Error Rate:", error\_knn2, "\n")

## Error Rate: 0.0210131

# Model with all significant predictors and K = 10  
knn\_predictions3 <- knn(  
 train = train\_data[, -length(predictors)],  
 test = test\_data[, -length(predictors)],  
 cl = train\_data$HeartDiseaseorAttack,  
 k = 10  
)  
knn\_results3 <- data.frame(  
 Actual = test\_data$HeartDiseaseorAttack,  
 Predicted = knn\_predictions3  
)  
knn\_results3$Correct <- knn\_results3$Actual == knn\_results3$Predicted  
knn\_confusion\_matrix3 <- table(Predicted = knn\_results3$Predicted, Actual = knn\_results3$Actual)  
print(knn\_confusion\_matrix3)

## Actual  
## Predicted 0 1  
## 0 60087 1679  
## 1 0 5335

accuracy\_knn3 <- (60087 + 5340) / (60087 + 1674 + 0 + 5340)  
error\_knn3 <- 1 - accuracy\_knn3  
cat("Accuracy:", accuracy\_knn3, "\n")

## Accuracy: 0.9750525

cat("Error Rate:", error\_knn3, "\n")

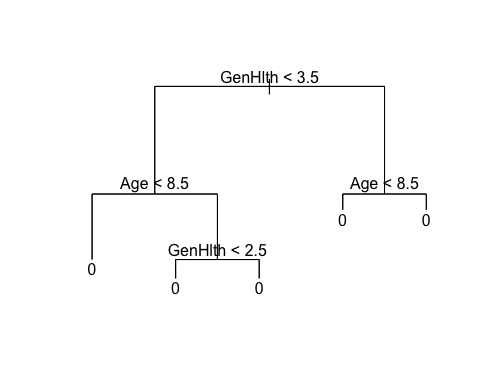
## Error Rate: 0.02494747

**Classification Trees**  
The classification tree containing all predictors and pruned with cross-validation has an accuracy of 89.55%.

# Model with all predictors  
heart\_tree <- tree(HeartDiseaseorAttack ~ ., heart\_train)  
summary(heart\_tree)

##   
## Classification tree:  
## tree(formula = HeartDiseaseorAttack ~ ., data = heart\_train)  
## Variables actually used in tree construction:  
## [1] "GenHlth" "Age"   
## Number of terminal nodes: 5   
## Residual mean deviance: 0.5353 = 95050 / 177600   
## Misclassification error rate: 0.09467 = 16812 / 177576

plot(heart\_tree)  
text(heart\_tree, pretty = 0)



# Test the model and calculate accuracy  
tree\_predict <- predict(heart\_tree, heart\_test, type = "class")  
tree\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict  
)  
tree\_results$Correct <- tree\_results$Actual == tree\_results$Predicted  
tree\_confusion\_matrix <- table(Predicted = tree\_results$Predicted, Actual = tree\_results$Actual)  
print(tree\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 60087 7014  
## 1 0 0

accuracy\_tree <- (60087 + 0) / (60087 + 7014 + 0 + 0)  
error\_tree <- 1 - accuracy\_tree  
cat("Accuracy:", accuracy\_tree, "\n")

## Accuracy: 0.895471

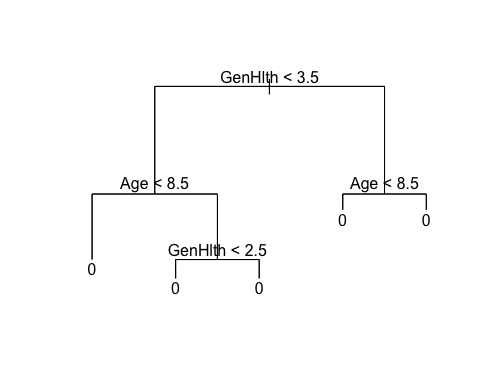
cat("Error Rate:", error\_tree, "\n")

## Error Rate: 0.104529

# Prune the tree with cross-validation  
set.seed(123)  
cv\_heart <- cv.tree(heart\_tree, FUN = prune.misclass)  
cv\_heart

## $size  
## [1] 5 1  
##   
## $dev  
## [1] 16812 16812  
##   
## $k  
## [1] -Inf 0  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

prune\_heart <- prune.misclass(heart\_tree, best = 5)  
plot(prune\_heart)  
text(prune\_heart, pretty = 0)



# Test the pruned tree and calculate accuracy  
tree\_predict2 <- predict(prune\_heart, heart\_test, type = "class")  
tree\_results2 <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict2  
)  
tree\_results2$Correct <- tree\_results2$Actual == tree\_results2$Predicted  
tree\_confusion\_matrix2 <- table(Predicted = tree\_results2$Predicted, Actual = tree\_results2$Actual)  
print(tree\_confusion\_matrix2)

## Actual  
## Predicted 0 1  
## 0 60087 7014  
## 1 0 0

accuracy\_tree2 <- (60087 + 0) / (60087 + 7014 + 0 + 0)  
error\_tree2 <- 1 - accuracy\_tree2  
cat("Accuracy:", accuracy\_tree2, "\n")

## Accuracy: 0.895471

cat("Error Rate:", error\_tree2, "\n")

## Error Rate: 0.104529

# The pruned and unpruned trees are the same

**Bagging**  
The bagged classification tree containing all predictors has an accuracy of 87.75%. The most important predictors in the tree are (1) Age, (2) BMI, and (3) Income.

# Model with all predictors  
set.seed(123)  
heart\_bag <- randomForest(HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 16, importance = TRUE, ntree = 25)  
heart\_bag

##   
## Call:  
## randomForest(formula = HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 16, importance = TRUE, ntree = 25)   
## Type of random forest: classification  
## Number of trees: 25  
## No. of variables tried at each split: 16  
##   
## OOB estimate of error rate: 10.92%  
## Confusion matrix:  
## 0 1 class.error  
## 0 155100 5661 0.03521376  
## 1 13724 3088 0.81632167

importance(heart\_bag)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## HighBP -9.453797 19.8925326 -0.3090534 631.2097  
## HighChol -12.549170 26.4123276 -3.2835389 564.9037  
## CholCheck 5.092407 -2.6031519 4.1494782 126.9510  
## BMI 7.533636 -3.8052444 4.9277909 5530.3945  
## Smoker -3.230946 7.6994319 0.4178417 869.6693  
## Stroke 10.338214 32.1654268 22.6743155 715.7023  
## Diabetes 3.808568 8.6099910 7.6360371 957.7022  
## PhysActivity 6.080891 -4.6828963 3.4617302 956.7861  
## Fruits 5.676729 -1.1941288 4.0209661 1082.7540  
## Veggies 2.003716 -0.1069718 1.8029961 846.1092  
## HvyAlcoholConsump 2.437638 -0.6126569 1.6702023 324.2037  
## AnyHealthcare 4.009127 -2.8926022 2.7306862 232.6367  
## NoDocbcCost 8.240178 -0.2863122 6.9116860 450.2840  
## GenHlth 16.273986 17.8019342 26.9336732 2279.5800  
## MentHlth 10.310916 -3.6418811 7.3903003 1950.9674  
## PhysHlth 14.775446 -2.4436951 14.2738733 2535.3968  
## DiffWalk 1.541262 12.2072850 9.2774139 860.0739  
## Sex 7.924247 11.9507307 13.6468996 645.5515  
## Age 11.125472 34.9689543 24.5523960 2777.3193  
## Education 19.012976 -6.2168920 14.6522430 2207.2180  
## Income 14.589597 -2.0897785 13.9938751 3157.4731

# Test the model and calculate accuracy  
tree\_predict\_bag <- predict(heart\_bag, heart\_test, type = "class")  
tree\_results\_bag <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict\_bag  
)  
tree\_results\_bag$Correct <- tree\_results\_bag$Actual == tree\_results\_bag$Predicted  
tree\_confusion\_matrix\_bag <- table(Predicted = tree\_results\_bag$Predicted, Actual = tree\_results\_bag$Actual)  
print(tree\_confusion\_matrix\_bag)

## Actual  
## Predicted 0 1  
## 0 58207 5894  
## 1 1880 1120

accuracy\_tree\_bag <- (57539 + 1341) / (57539 + 5673 + 2548 + 1341)  
error\_tree\_bag <- 1 - accuracy\_tree\_bag  
cat("Accuracy:", accuracy\_tree\_bag, "\n")

## Accuracy: 0.8774832

cat("Error Rate:", error\_tree\_bag, "\n")

## Error Rate: 0.1225168

**Random Forests**  
The random forest classification tree containing all predictors has an accuracy of 89.45%. The most important predictors in the tree are (1) Age, (2) BMI, and (3) GenHlth.

# Model with all predictors  
set.seed(123)  
heart\_rf <- randomForest(HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 4, importance = TRUE, ntree = 25)  
heart\_rf

##   
## Call:  
## randomForest(formula = HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 4, importance = TRUE, ntree = 25)   
## Type of random forest: classification  
## Number of trees: 25  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 9.84%  
## Confusion matrix:  
## 0 1 class.error  
## 0 158101 2661 0.01655242  
## 1 14810 2001 0.88097079

importance(heart\_rf)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## HighBP -9.6305526 12.5565119 0.6843860 809.81440  
## HighChol -10.3694707 14.8229002 0.9644538 679.93561  
## CholCheck 2.6103315 -0.4093466 2.5155214 97.11157  
## BMI 12.9636999 -2.9705478 9.8569128 3263.23924  
## Smoker -9.5435946 9.6846098 -0.3882864 502.72400  
## Stroke 5.6253149 39.7938864 18.8813333 763.19208  
## Diabetes -0.6267632 9.1181822 3.6660343 673.13682  
## PhysActivity 4.7218669 -2.5749532 2.2937435 620.59727  
## Fruits 3.8365262 -1.7856203 2.1357503 668.07655  
## Veggies 2.0719579 -2.0302961 1.0935410 591.07056  
## HvyAlcoholConsump 2.6230475 0.3989772 2.9316309 209.55416  
## AnyHealthcare 4.4841208 -0.5483266 4.0833164 182.03580  
## NoDocbcCost 5.4151921 -2.0114670 3.8731648 354.61536  
## GenHlth 7.0065827 26.9871170 16.3452153 1762.41789  
## MentHlth 12.8490718 -7.6722303 10.5400008 1425.18542  
## PhysHlth 10.0921953 -1.8453169 9.8522780 1872.28068  
## DiffWalk 1.4581647 12.2251124 9.6233747 682.55238  
## Sex 9.2676631 10.2017891 11.2873441 595.04416  
## Age 6.4873333 19.8988746 16.6811849 2487.90392  
## Education 11.2803123 -3.7764568 8.2815047 1442.37075  
## Income 6.3686110 2.1722698 7.6344434 1940.88147

# Test the model and calculate accuracy  
tree\_predict\_rf <- predict(heart\_rf, heart\_test, type = "class")  
tree\_results\_rf <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict\_rf  
)  
tree\_results\_rf$Correct <- tree\_results\_rf$Actual == tree\_results\_rf$Predicted  
tree\_confusion\_matrix\_rf <- table(Predicted = tree\_results\_rf$Predicted, Actual = tree\_results\_rf$Actual)  
print(tree\_confusion\_matrix\_rf)

## Actual  
## Predicted 0 1  
## 0 59362 6335  
## 1 725 679

accuracy\_tree\_rf <- (59362 + 679) / (59362 + 6335 + 725 + 679)  
error\_tree\_rf <- 1 - accuracy\_tree\_rf  
cat("Accuracy:", accuracy\_tree\_rf, "\n")

## Accuracy: 0.8947855

cat("Error Rate:", error\_tree\_rf, "\n")

## Error Rate: 0.1052145

**Overview of models**  
Overall, the model with the highest accuracy is kNN with k=3. This model has an accuracy of 98.13%.

# Table with models and relative accuracies  
classification\_overview <- data.frame(  
 Method = c("Logistic Regression", "LDA", "QDA", "Naive Bayes", "kNN (k=3)", "CV Tree", "Bagging", "Random Forest"),  
 Accuracy = c("89.78%", "89.14%", "83.25%", "80.39%", "98.13%", "89.55%", "87.75%", "89.45%")  
)  
classification\_table <- kable(classification\_overview, "markdown") %>%  
 kable\_styling(full\_width = FALSE) %>%  
 column\_spec(1, bold = TRUE)

## Warning in kable\_styling(., full\_width = FALSE): Please specify format in  
## kable. kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

## Warning in column\_spec(., 1, bold = TRUE): Please specify format in kable.  
## kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

classification\_table

| Method | Accuracy |
| --- | --- |
| Logistic Regression | 89.78% |
| LDA | 89.14% |
| QDA | 83.25% |
| Naive Bayes | 80.39% |
| kNN (k=3) | 98.13% |
| CV Tree | 89.55% |
| Bagging | 87.75% |
| Random Forest | 89.45% |

# Section 2: What indicators can be used to predict HIV rates?

**Loading data and packages**

library(readxl)  
library(tidyverse)  
library(boot)  
library(pls)

##   
## Attaching package: 'pls'

## The following object is masked from 'package:corrplot':  
##   
## corrplot

## The following object is masked from 'package:stats':  
##   
## loadings

library(corrplot)  
library(leaps)  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1-8

library(tree)  
library(randomForest)  
library(gbm)  
library(kableExtra)  
library(tidyverse)  
hiv\_df <- read\_xlsx("Data/HIV\_Data.xlsx")

**Data Tidying**  
Cleaned and formatted data, and used bootstrapping to generate additional observations.

# Remove male and female (only interested in total)  
hiv\_df <- subset(hiv\_df, !(Sex %in% c("Male", "Female")))  
hiv\_df <- hiv\_df[ , -3]  
  
# Pivot from long to wide format  
hiv\_df <- pivot\_wider(  
 data = hiv\_df,  
 names\_from = c(Indicator), # Specify the columns to pivot  
 values\_from = c(OBS\_VALUE), # Specify the values column  
 names\_sep = "\_"  
)

## Warning: Values from `OBS\_VALUE` are not uniquely identified; output will contain  
## list-cols.  
## • Use `values\_fn = list` to suppress this warning.  
## • Use `values\_fn = {summary\_fun}` to summarise duplicates.  
## • Use the following dplyr code to identify duplicates.  
## {data} %>%  
## dplyr::group\_by(`Geographic area`, TIME\_PERIOD, Indicator) %>%  
## dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%  
## dplyr::filter(n > 1L)

hiv\_df <- hiv\_df[ , -13]  
  
# Change column names  
colnames(hiv\_df) <- c("country", "year", "infant\_mortality\_rate", "fertility\_rate", "life\_exp", "pop\_growth\_rate", "urban\_pop", "youth\_literacy\_rate", "aids\_death\_rate", "hiv\_infection\_rate", "mother\_child\_hiv\_transmis\_rate", "per\_child\_under\_poverty")  
  
# Interpolate missing values for years based on other years for that country  
hiv\_df <- hiv\_df %>%  
 group\_by(country) %>%  
 fill(c(pop\_growth\_rate, youth\_literacy\_rate, per\_child\_under\_poverty), .direction = "updown") %>%  
 ungroup()  
  
# Convert to numeric columns  
hiv\_df <- hiv\_df %>%  
 unnest\_wider(infant\_mortality\_rate, names\_sep = "\_") %>%  
 unnest\_wider(fertility\_rate, names\_sep = "\_") %>%  
 unnest\_wider(life\_exp, names\_sep = "\_") %>%  
 unnest\_wider(pop\_growth\_rate, names\_sep = "\_") %>%  
 unnest\_wider(urban\_pop, names\_sep = "\_") %>%  
 unnest\_wider(youth\_literacy\_rate, names\_sep = "\_") %>%  
 unnest\_wider(aids\_death\_rate, names\_sep = "\_") %>%  
 unnest\_wider(hiv\_infection\_rate, names\_sep = "\_") %>%  
 unnest\_wider(mother\_child\_hiv\_transmis\_rate, names\_sep = "\_") %>%  
 unnest\_wider(per\_child\_under\_poverty, names\_sep = "\_") %>%  
 mutate(across(-c(country, year), ~as.numeric(as.character(.))))

## Warning: There were 2 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `across(-c(country, year), ~as.numeric(as.character(.)))`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 1 remaining warning.

hiv\_df <- hiv\_df[ , -c(4, 5, 7, 9, 11)]  
colnames(hiv\_df) <- c("country", "year", "infant\_mortality\_rate", "fertility\_rate", "life\_exp", "pop\_growth\_rate", "urban\_pop", "youth\_literacy\_rate", "aids\_death\_rate", "hiv\_infection\_rate", "mother\_child\_hiv\_transmis\_rate", "per\_child\_under\_poverty")  
  
# Check for NA values  
colSums(is.na(hiv\_df))

## country year   
## 1 1   
## infant\_mortality\_rate fertility\_rate   
## 325 199   
## life\_exp pop\_growth\_rate   
## 199 199   
## urban\_pop youth\_literacy\_rate   
## 235 1504   
## aids\_death\_rate hiv\_infection\_rate   
## 1227 1304   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty   
## 1540 1324

hiv\_df <- na.omit(hiv\_df)  
  
# Use bootstrapping to expand the dataset to 1000 observations  
set.seed(123)  
generate\_boot\_dataset <- function(data) {  
 boot\_indices <- sample(nrow(data), replace = TRUE)  
 boot\_data <- data[boot\_indices, ]  
 return(boot\_data)  
}  
num\_bootstrap\_samples <- ceiling(1000 / nrow(hiv\_df))  
boot\_datasets <- lapply(1:num\_bootstrap\_samples, function(i) generate\_boot\_dataset(hiv\_df))  
hiv\_df\_boot <- do.call(rbind, boot\_datasets)

**Data Exploration**  
Colinearity is present between multiple variables (infant\_mortality\_rate, fertility\_rate, life\_exp, pop\_growth\_rate, and youth\_literacy\_rate all seem to be correlated). This is something to keep in mind during the evaluation of models.

# View the dataset and summary statistics  
head(hiv\_df\_boot)

## # A tibble: 6 × 12  
## country year infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Indonesia 2019 20.1 2.21 70.5 0.654  
## 2 Algeria 2017 21.0 3.05 75.7 1.24   
## 3 Kyrgyzstan 2017 17.8 3.09 70.6 1.33   
## 4 Sao Tome … 2015 18.6 4.39 67.2 1.75   
## 5 El Salvad… 2013 14.4 2.15 71.8 0.370  
## 6 Rwanda 2017 32.9 4.03 65.9 2.00   
## # ℹ 6 more variables: urban\_pop <dbl>, youth\_literacy\_rate <dbl>,  
## # aids\_death\_rate <dbl>, hiv\_infection\_rate <dbl>,  
## # mother\_child\_hiv\_transmis\_rate <dbl>, per\_child\_under\_poverty <dbl>

summary(hiv\_df\_boot)

## country year infant\_mortality\_rate fertility\_rate   
## Length:1233 Length:1233 Min. : 2.05 Min. :1.331   
## Class :character Class :character 1st Qu.:14.45 1st Qu.:2.194   
## Mode :character Mode :character Median :28.98 Median :3.050   
## Mean :32.44 Mean :3.389   
## 3rd Qu.:45.57 3rd Qu.:4.489   
## Max. :98.03 Max. :7.344   
## life\_exp pop\_growth\_rate urban\_pop youth\_literacy\_rate  
## Min. :51.49 Min. :-0.4127 Min. :16.21 Min. : 43.46   
## 1st Qu.:61.62 1st Qu.: 0.8632 1st Qu.:35.59 1st Qu.: 83.63   
## Median :66.44 Median : 1.3011 Median :50.65 Median : 95.42   
## Mean :67.27 Mean : 1.3996 Mean :51.78 Mean : 87.42   
## 3rd Qu.:73.13 3rd Qu.: 2.0922 3rd Qu.:68.87 3rd Qu.: 99.01   
## Max. :79.48 Max. : 3.3609 Max. :95.60 Max. :100.00   
## aids\_death\_rate hiv\_infection\_rate mother\_child\_hiv\_transmis\_rate  
## Min. : 0.030 Min. : 0.010 Min. : 1.61   
## 1st Qu.: 0.220 1st Qu.: 0.040 1st Qu.: 13.46   
## Median : 1.180 Median : 0.260 Median : 18.92   
## Mean : 9.953 Mean : 1.049 Mean : 21.96   
## 3rd Qu.:15.610 3rd Qu.: 0.810 3rd Qu.: 28.33   
## Max. :96.080 Max. :17.610 Max. :128.28   
## per\_child\_under\_poverty  
## Min. : 0.4306   
## 1st Qu.:25.1501   
## Median :34.9413   
## Mean :37.5100   
## 3rd Qu.:50.1748   
## Max. :84.6042

str(hiv\_df\_boot)

## tibble [1,233 × 12] (S3: tbl\_df/tbl/data.frame)  
## $ country : chr [1:1233] "Indonesia" "Algeria" "Kyrgyzstan" "Sao Tome and Principe" ...  
## $ year : chr [1:1233] "2019" "2017" "2017" "2015" ...  
## $ infant\_mortality\_rate : num [1:1233] 20.1 21 17.8 18.6 14.4 ...  
## $ fertility\_rate : num [1:1233] 2.21 3.05 3.09 4.39 2.15 ...  
## $ life\_exp : num [1:1233] 70.5 75.7 70.6 67.2 71.8 ...  
## $ pop\_growth\_rate : num [1:1233] 0.654 1.237 1.328 1.749 0.37 ...  
## $ urban\_pop : num [1:1233] 56 72.1 36.1 70.2 68 ...  
## $ youth\_literacy\_rate : num [1:1233] 99.8 97.4 99.8 97.8 98.5 ...  
## $ aids\_death\_rate : num [1:1233] 0.44 0.07 0.03 11.07 0.29 ...  
## $ hiv\_infection\_rate : num [1:1233] 0.16 0.03 0.03 0.04 0.21 0.42 0.36 0.48 0.03 1.68 ...  
## $ mother\_child\_hiv\_transmis\_rate: num [1:1233] 29.7 24.8 31.1 21.4 13.9 ...  
## $ per\_child\_under\_poverty : num [1:1233] 10.55 6.33 29.26 63.73 34.94 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:2011] 10 11 12 13 14 15 16 17 18 28 ...  
## ..- attr(\*, "names")= chr [1:2011] "10" "11" "12" "13" ...

dim(hiv\_df\_boot)

## [1] 1233 12

names(hiv\_df\_boot)

## [1] "country" "year"   
## [3] "infant\_mortality\_rate" "fertility\_rate"   
## [5] "life\_exp" "pop\_growth\_rate"   
## [7] "urban\_pop" "youth\_literacy\_rate"   
## [9] "aids\_death\_rate" "hiv\_infection\_rate"   
## [11] "mother\_child\_hiv\_transmis\_rate" "per\_child\_under\_poverty"

# Check for correlated variables  
cor\_matrix <- cor(hiv\_df\_boot[, c(3:12)])  
corrplot(cor\_matrix, method = "color", tl.col = "black", tl.srt = 45)

## Warning in plot.window(...): "method" is not a graphical parameter

## Warning in plot.window(...): "tl.col" is not a graphical parameter

## Warning in plot.window(...): "tl.srt" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "method" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "tl.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "tl.srt" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "method" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.col" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.srt" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "method" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.col" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.srt" is not a  
## graphical parameter

## Warning in box(...): "method" is not a graphical parameter

## Warning in box(...): "tl.col" is not a graphical parameter

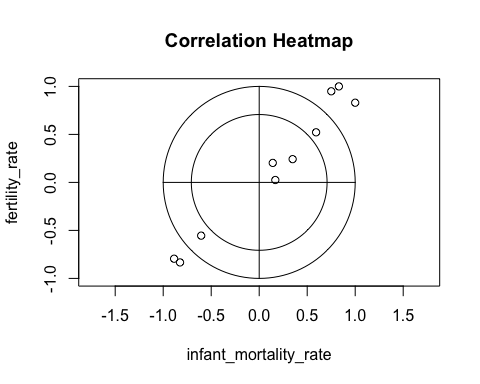
## Warning in box(...): "tl.srt" is not a graphical parameter

## Warning in title(...): "method" is not a graphical parameter

## Warning in title(...): "tl.col" is not a graphical parameter

## Warning in title(...): "tl.srt" is not a graphical parameter

title("Correlation Heatmap")



cor\_matrix\_filtered <- cor\_matrix  
cor\_matrix\_filtered[abs(cor\_matrix) <= 0.7] <- NA  
print(cor\_matrix\_filtered)

## infant\_mortality\_rate fertility\_rate life\_exp  
## infant\_mortality\_rate 1.0000000 0.8300778 -0.8865746  
## fertility\_rate 0.8300778 1.0000000 -0.7949911  
## life\_exp -0.8865746 -0.7949911 1.0000000  
## pop\_growth\_rate 0.7506318 0.9491751 -0.7188660  
## urban\_pop NA NA NA  
## youth\_literacy\_rate -0.8245610 -0.8335437 NA  
## aids\_death\_rate NA NA NA  
## hiv\_infection\_rate NA NA NA  
## mother\_child\_hiv\_transmis\_rate NA NA NA  
## per\_child\_under\_poverty NA NA NA  
## pop\_growth\_rate urban\_pop youth\_literacy\_rate  
## infant\_mortality\_rate 0.7506318 NA -0.8245610  
## fertility\_rate 0.9491751 NA -0.8335437  
## life\_exp -0.7188660 NA NA  
## pop\_growth\_rate 1.0000000 NA -0.7238268  
## urban\_pop NA 1 NA  
## youth\_literacy\_rate -0.7238268 NA 1.0000000  
## aids\_death\_rate NA NA NA  
## hiv\_infection\_rate NA NA NA  
## mother\_child\_hiv\_transmis\_rate NA NA NA  
## per\_child\_under\_poverty NA NA NA  
## aids\_death\_rate hiv\_infection\_rate  
## infant\_mortality\_rate NA NA  
## fertility\_rate NA NA  
## life\_exp NA NA  
## pop\_growth\_rate NA NA  
## urban\_pop NA NA  
## youth\_literacy\_rate NA NA  
## aids\_death\_rate 1.0000000 0.8657092  
## hiv\_infection\_rate 0.8657092 1.0000000  
## mother\_child\_hiv\_transmis\_rate NA NA  
## per\_child\_under\_poverty NA NA  
## mother\_child\_hiv\_transmis\_rate  
## infant\_mortality\_rate NA  
## fertility\_rate NA  
## life\_exp NA  
## pop\_growth\_rate NA  
## urban\_pop NA  
## youth\_literacy\_rate NA  
## aids\_death\_rate NA  
## hiv\_infection\_rate NA  
## mother\_child\_hiv\_transmis\_rate 1  
## per\_child\_under\_poverty NA  
## per\_child\_under\_poverty  
## infant\_mortality\_rate NA  
## fertility\_rate NA  
## life\_exp NA  
## pop\_growth\_rate NA  
## urban\_pop NA  
## youth\_literacy\_rate NA  
## aids\_death\_rate NA  
## hiv\_infection\_rate NA  
## mother\_child\_hiv\_transmis\_rate NA  
## per\_child\_under\_poverty 1

**Best Subset Selection**  
The model created using best subset selection determined the model with the lowest MSE is one that contains 6 predictors (infant\_mortality\_rate, fertility\_rate, youth\_literacy\_rate, aids\_death\_rate, mother\_child\_hiv\_transmis\_rate, and per\_child\_under\_poverty) with an MSE of 1.116785 and an adjusted R-squared value of 0.778.

# Creating a training and testing set  
set.seed(123)   
train\_hiv <- sample(c(TRUE, FALSE), nrow(hiv\_df\_boot), replace = TRUE)  
test\_hiv <- (!train\_hiv)  
  
# Creating the model  
best\_subset\_hiv <- regsubsets(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ], nvmax = 9)  
best\_summary <- summary(best\_subset\_hiv)  
best\_summary

## Subset selection object  
## Call: regsubsets.formula(hiv\_infection\_rate ~ infant\_mortality\_rate +   
## fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop +   
## youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate +   
## per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ],   
## nvmax = 9)  
## 9 Variables (and intercept)  
## Forced in Forced out  
## infant\_mortality\_rate FALSE FALSE  
## fertility\_rate FALSE FALSE  
## life\_exp FALSE FALSE  
## pop\_growth\_rate FALSE FALSE  
## urban\_pop FALSE FALSE  
## youth\_literacy\_rate FALSE FALSE  
## aids\_death\_rate FALSE FALSE  
## mother\_child\_hiv\_transmis\_rate FALSE FALSE  
## per\_child\_under\_poverty FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: exhaustive  
## infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " "   
## 3 ( 1 ) " " "\*" " " " "   
## 4 ( 1 ) " " "\*" " " " "   
## 5 ( 1 ) " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## urban\_pop youth\_literacy\_rate aids\_death\_rate  
## 1 ( 1 ) " " " " "\*"   
## 2 ( 1 ) " " " " "\*"   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"

# Testing and determining the best model  
test\_mat\_best <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[test\_hiv, ])  
val.errors <- rep(NA, 9)  
for (i in 1:9) {  
 coefi <- coef(best\_subset\_hiv, id = i)  
 pred <- test\_mat\_best[, names(coefi)] %\*% coefi  
 val.errors[i] <- mean((hiv\_df\_boot$hiv\_infection\_rate[test\_hiv] - pred)^2)  
}  
val.errors

## [1] 1.393887 1.192356 1.159057 1.144428 1.123322 1.116785 1.117192 1.118514  
## [9] 1.118408

which.min(val.errors)

## [1] 6

coef(best\_subset\_hiv, 6)

## (Intercept) infant\_mortality\_rate   
## 2.912988940 -0.008395259   
## fertility\_rate youth\_literacy\_rate   
## -0.555738245 -0.019081947   
## aids\_death\_rate mother\_child\_hiv\_transmis\_rate   
## 0.139927726 0.008088391   
## per\_child\_under\_poverty   
## 0.010227776

cat("Adjusted RSq:", best\_summary$rsq[6], "\n")

## Adjusted RSq: 0.778206

cat("MSE:", val.errors[6], "\n")

## MSE: 1.116785

**Forward Stepwise Selection**  
The best model created with forward stepwise selection contains the same 6 predictors as the best subset selection model, with the same MSE of 1.116785 and adjusted R-squared value of 0.778.

forward\_subset\_hiv <- regsubsets(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ], nvmax = 9, method = "forward")  
forward\_summary <- summary(forward\_subset\_hiv)  
forward\_summary

## Subset selection object  
## Call: regsubsets.formula(hiv\_infection\_rate ~ infant\_mortality\_rate +   
## fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop +   
## youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate +   
## per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ],   
## nvmax = 9, method = "forward")  
## 9 Variables (and intercept)  
## Forced in Forced out  
## infant\_mortality\_rate FALSE FALSE  
## fertility\_rate FALSE FALSE  
## life\_exp FALSE FALSE  
## pop\_growth\_rate FALSE FALSE  
## urban\_pop FALSE FALSE  
## youth\_literacy\_rate FALSE FALSE  
## aids\_death\_rate FALSE FALSE  
## mother\_child\_hiv\_transmis\_rate FALSE FALSE  
## per\_child\_under\_poverty FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: forward  
## infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " "   
## 3 ( 1 ) " " "\*" " " " "   
## 4 ( 1 ) " " "\*" " " " "   
## 5 ( 1 ) " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## urban\_pop youth\_literacy\_rate aids\_death\_rate  
## 1 ( 1 ) " " " " "\*"   
## 2 ( 1 ) " " " " "\*"   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"

# Testing and determining the best model  
test\_mat\_forward <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[test\_hiv, ], method = "forward")  
val.errors2 <- rep(NA, 9)  
for (i in 1:9) {  
 coefi <- coef(forward\_subset\_hiv, id = i)  
 pred <- test\_mat\_forward[, names(coefi)] %\*% coefi  
 val.errors2[i] <- mean((hiv\_df\_boot$hiv\_infection\_rate[test\_hiv] - pred)^2)  
}  
val.errors2

## [1] 1.393887 1.192356 1.159057 1.144428 1.123322 1.116785 1.117192 1.118514  
## [9] 1.118408

which.min(val.errors2)

## [1] 6

coef(forward\_subset\_hiv, 6)

## (Intercept) infant\_mortality\_rate   
## 2.912988940 -0.008395259   
## fertility\_rate youth\_literacy\_rate   
## -0.555738245 -0.019081947   
## aids\_death\_rate mother\_child\_hiv\_transmis\_rate   
## 0.139927726 0.008088391   
## per\_child\_under\_poverty   
## 0.010227776

cat("Adjusted RSq:", forward\_summary$rsq[6], "\n")

## Adjusted RSq: 0.778206

cat("MSE:", val.errors2[6], "\n")

## MSE: 1.116785

**Backward Stepwise Selection**  
The best model created with backward stepwise selection contains the same 6 predictors as the best subset selection model, with the same MSE of 1.116785 and adjusted R-squared value of 0.778.

backward\_subset\_hiv <- regsubsets(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ], nvmax = 9, method = "backward")  
backward\_summary <- summary(backward\_subset\_hiv)  
backward\_summary

## Subset selection object  
## Call: regsubsets.formula(hiv\_infection\_rate ~ infant\_mortality\_rate +   
## fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop +   
## youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate +   
## per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ],   
## nvmax = 9, method = "backward")  
## 9 Variables (and intercept)  
## Forced in Forced out  
## infant\_mortality\_rate FALSE FALSE  
## fertility\_rate FALSE FALSE  
## life\_exp FALSE FALSE  
## pop\_growth\_rate FALSE FALSE  
## urban\_pop FALSE FALSE  
## youth\_literacy\_rate FALSE FALSE  
## aids\_death\_rate FALSE FALSE  
## mother\_child\_hiv\_transmis\_rate FALSE FALSE  
## per\_child\_under\_poverty FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: backward  
## infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " "   
## 3 ( 1 ) " " "\*" " " " "   
## 4 ( 1 ) " " "\*" " " " "   
## 5 ( 1 ) " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## urban\_pop youth\_literacy\_rate aids\_death\_rate  
## 1 ( 1 ) " " " " "\*"   
## 2 ( 1 ) " " " " "\*"   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"

# Testing and determining the best model  
test\_mat\_backward <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[test\_hiv, ], method = "backward")  
val.errors3 <- rep(NA, 9)  
for (i in 1:9) {  
 coefi <- coef(forward\_subset\_hiv, id = i)  
 pred <- test\_mat\_backward[, names(coefi)] %\*% coefi  
 val.errors3[i] <- mean((hiv\_df\_boot$hiv\_infection\_rate[test\_hiv] - pred)^2)  
}  
val.errors3

## [1] 1.393887 1.192356 1.159057 1.144428 1.123322 1.116785 1.117192 1.118514  
## [9] 1.118408

which.min(val.errors3)

## [1] 6

coef(forward\_subset\_hiv, 6)

## (Intercept) infant\_mortality\_rate   
## 2.912988940 -0.008395259   
## fertility\_rate youth\_literacy\_rate   
## -0.555738245 -0.019081947   
## aids\_death\_rate mother\_child\_hiv\_transmis\_rate   
## 0.139927726 0.008088391   
## per\_child\_under\_poverty   
## 0.010227776

cat("Adjusted RSq:", backward\_summary$rsq[6], "\n")

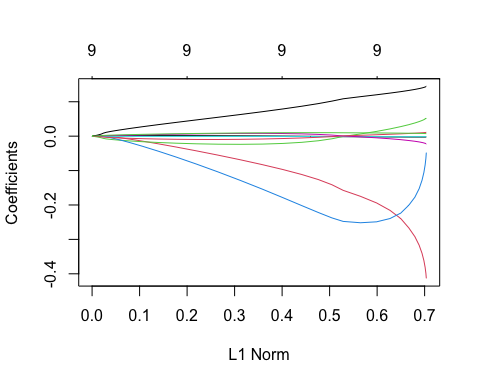
## Adjusted RSq: 0.778206

cat("MSE:", val.errors3[6], "\n")

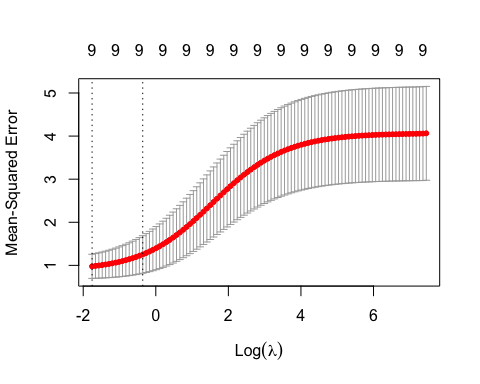
## MSE: 1.116785

**Ridge Regression**  
The model created with ridge regression has an MSE of 1.655353.

# Creating x and y vectors  
x <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, hiv\_df\_boot)[, -1]  
y <- hiv\_df\_boot$hiv\_infection\_rate  
  
# Creating test and training sets  
set.seed(123)  
train <- sample(1:nrow(x), nrow(x) / 2)  
test <- (-train)  
y.test <- y[test]  
  
# Creating the model  
grid <- 10^seq(10, -2, length = 100)  
ridge\_hiv <- glmnet(x[train, ], y[train], alpha = 0, lambda = grid)  
plot(ridge\_hiv)



# Determing the best lambda value  
set.seed(123)  
cv\_out <- cv.glmnet(x[train, ], y[train], alpha = 0)  
plot(cv\_out)



bestlam\_ridge <- cv\_out$lambda.min  
bestlam\_ridge

## [1] 0.1728292

# Testing the model and calculating MSE  
ridge\_pred\_hiv <- predict(ridge\_hiv, s = bestlam\_ridge, newx = x[test, ])  
out <- glmnet(x, y, alpha = 0)  
predict(out, type = "coefficients", s = bestlam\_ridge)[1:10, ]

## (Intercept) infant\_mortality\_rate   
## 1.9107533692 -0.0035154918   
## fertility\_rate life\_exp   
## -0.2198640258 -0.0186218317   
## pop\_growth\_rate urban\_pop   
## -0.3426151115 -0.0011982380   
## youth\_literacy\_rate aids\_death\_rate   
## 0.0009869136 0.1169710784   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty   
## 0.0026597813 0.0128554347

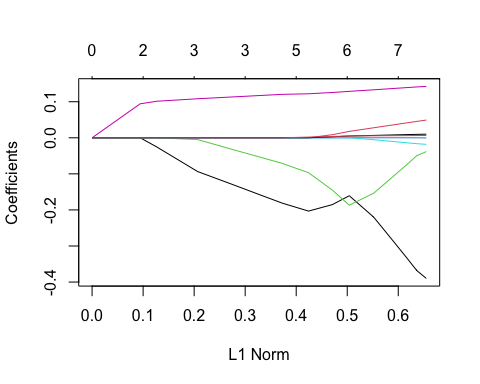
cat("MSE:", mean((ridge\_pred\_hiv - y.test)^2), "\n")

## MSE: 1.655353

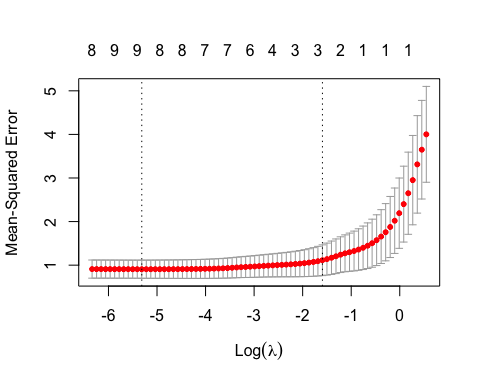
**Lasso**  
The model created with the lasso method has an MSE of 1.586618.

# Creating the model  
lasso\_hiv <- glmnet(x[train, ], y[train], alpha = 1, lambda = grid)  
plot(lasso\_hiv)

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):  
## collapsing to unique 'x' values



# Determining the best lambda value  
set.seed(123)  
cv\_out2 <- cv.glmnet(x[train, ], y[train], alpha = 1)  
plot(cv\_out2)



bestlam\_lasso <- cv\_out2$lambda.min  
bestlam\_lasso

## [1] 0.004922238

# Testing the model and calculating MSE  
lasso\_pred\_hiv <- predict(lasso\_hiv, s = bestlam\_lasso, newx = x[test, ])  
out2 <- glmnet(x, y, alpha = 1)  
predict(out2, type = "coefficients", s = bestlam\_lasso)[1:10, ]

## (Intercept) infant\_mortality\_rate   
## 1.898350049 -0.005324432   
## fertility\_rate life\_exp   
## -0.445373207 0.006512949   
## pop\_growth\_rate urban\_pop   
## -0.123410016 0.000000000   
## youth\_literacy\_rate aids\_death\_rate   
## -0.015631559 0.138645679   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty   
## 0.008562343 0.009742525

cat("MSE:", mean((lasso\_pred\_hiv - y.test)^2), "\n")

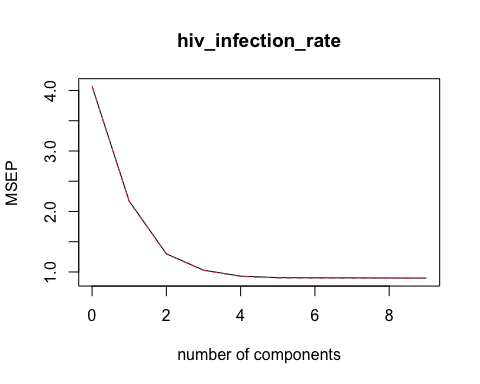
## MSE: 1.586618

**Partial Least Squares (PLS)**  
The model created with the partial least squares method has an MSE of 1.587266.

# Creating the model  
set.seed(123)  
pls\_hiv <- plsr(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, scale = TRUE, validation = "CV")  
summary(pls\_hiv)

## Data: X dimension: 616 9   
## Y dimension: 616 1  
## Fit method: kernelpls  
## Number of components considered: 9  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 2.017 1.473 1.140 1.015 0.9648 0.9519 0.9502  
## adjCV 2.017 1.471 1.138 1.012 0.9619 0.9492 0.9474  
## 7 comps 8 comps 9 comps  
## CV 0.9498 0.9486 0.9481  
## adjCV 0.9470 0.9458 0.9453  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## X 38.49 73.96 81.31 84.97 89.99 94.80  
## hiv\_infection\_rate 49.04 70.36 76.84 79.49 80.04 80.13  
## 7 comps 8 comps 9 comps  
## X 97.48 99.55 100.00  
## hiv\_infection\_rate 80.20 80.25 80.27

# Determining the best M value and calculating MSE  
validationplot(pls\_hiv, val.type = "MSEP")



# The lowest cross-validation error occurs when M = 5 partial least squares directions are used.   
pls\_pred\_hiv <- predict(pls\_hiv, x[test, ], ncomp = 5)  
pls\_fit\_hiv <- plsr(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, scale = TRUE, ncomp = 5)  
summary(pls\_fit\_hiv)

## Data: X dimension: 616 9   
## Y dimension: 616 1  
## Fit method: kernelpls  
## Number of components considered: 5  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps  
## X 38.49 73.96 81.31 84.97 89.99  
## hiv\_infection\_rate 49.04 70.36 76.84 79.49 80.04

coef(pls\_fit\_hiv)

## , , 5 comps  
##   
## hiv\_infection\_rate  
## infant\_mortality\_rate -0.10814010  
## fertility\_rate -0.28171048  
## life\_exp 0.33081303  
## pop\_growth\_rate -0.29326391  
## urban\_pop 0.00452169  
## youth\_literacy\_rate -0.35790432  
## aids\_death\_rate 2.08548682  
## mother\_child\_hiv\_transmis\_rate 0.16845244  
## per\_child\_under\_poverty 0.15944112

cat("MSE:", mean((pls\_pred\_hiv - y.test)^2), "\n")

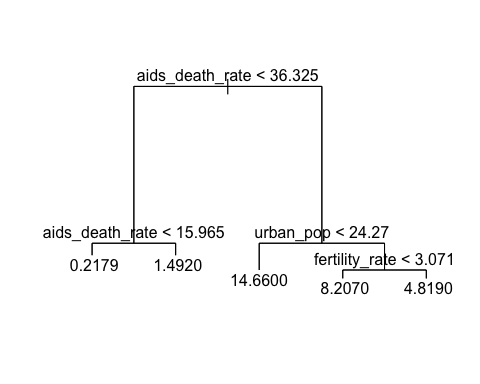
## MSE: 1.587266

**Regression Tree**  
The regression tree created with cross validation has an MSE of 0.8744749.

# Creating a training data set  
set.seed(123)  
train <- sample(1:nrow(hiv\_df\_boot), nrow(hiv\_df\_boot) / 2)  
  
# Creating the model  
tree\_hiv <- tree(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, hiv\_df\_boot, subset = train)  
summary(tree\_hiv)

##   
## Regression tree:  
## tree(formula = hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate +   
## life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate +   
## aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty,   
## data = hiv\_df\_boot, subset = train)  
## Variables actually used in tree construction:  
## [1] "aids\_death\_rate" "urban\_pop" "fertility\_rate"   
## Number of terminal nodes: 5   
## Residual mean deviance: 0.3543 = 216.5 / 611   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -6.8420 -0.1879 -0.1099 0.0000 0.1082 4.1530

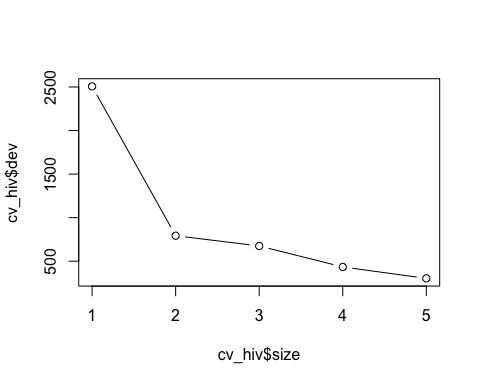
plot(tree\_hiv)  
text(tree\_hiv, pretty = 0)



# Prune the tree with cross validation  
cv\_hiv <- cv.tree(tree\_hiv)  
cv\_hiv

## $size  
## [1] 5 4 3 2 1  
##   
## $dev  
## [1] 302.8638 434.2763 675.0429 792.6930 2506.5022  
##   
## $k  
## [1] -Inf 93.90304 134.19955 299.35920 1753.62875  
##   
## $method  
## [1] "deviance"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

plot(cv\_hiv$size, cv\_hiv$dev, type = "b")



# The best number of terminal nodes was already selected by the unpruned tree, at 5.  
yhat <- predict(tree\_hiv, newdata = hiv\_df\_boot[-train, ])  
hiv\_test <- unlist(hiv\_df\_boot[-train, "hiv\_infection\_rate"])  
cat("MSE:", mean((yhat - hiv\_test)^2), "\n")

## MSE: 0.8744749

**Bagging**  
The regression tree created with bagging has an MSE of 0.07755276.

# Creating the model  
set.seed(123)  
bag\_hiv <- randomForest(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, mtry = 9, importance = TRUE)  
bag\_hiv

##   
## Call:  
## randomForest(formula = hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, mtry = 9, importance = TRUE, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 9  
##   
## Mean of squared residuals: 0.09357584  
## % Var explained: 97.69

importance(bag\_hiv)

## %IncMSE IncNodePurity  
## infant\_mortality\_rate 13.616166 124.74003  
## fertility\_rate 10.180432 23.54548  
## life\_exp 9.992152 108.80376  
## pop\_growth\_rate 11.986066 33.54232  
## urban\_pop 14.014906 119.03309  
## youth\_literacy\_rate 16.104698 44.77552  
## aids\_death\_rate 74.378480 1984.55314  
## mother\_child\_hiv\_transmis\_rate 23.805099 45.66542  
## per\_child\_under\_poverty 12.448492 32.58985

# Testing the model and calculating MSE  
yhat\_bag <- predict(bag\_hiv, newdata = hiv\_df\_boot[-train, ])  
cat("MSE:", mean((yhat\_bag - hiv\_test)^2), "\n")

## MSE: 0.07755176

**Random Forests**  
The regression tree created with random forests has an MSE of 0.118209.

# Creating the model  
set.seed(123)  
rf\_hiv <- randomForest(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, mtry = 3, importance = TRUE)  
rf\_hiv

##   
## Call:  
## randomForest(formula = hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, mtry = 3, importance = TRUE, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 0.1168581  
## % Var explained: 97.12

importance(rf\_hiv)

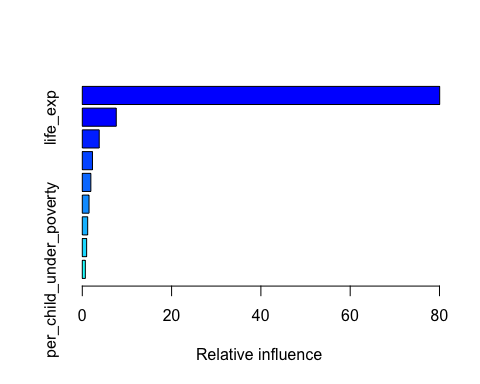
## %IncMSE IncNodePurity  
## infant\_mortality\_rate 7.218622 167.13185  
## fertility\_rate 6.259991 104.04900  
## life\_exp 8.611361 333.75934  
## pop\_growth\_rate 6.351998 127.73579  
## urban\_pop 13.701107 87.23218  
## youth\_literacy\_rate 7.348407 169.03632  
## aids\_death\_rate 28.872739 1053.24399  
## mother\_child\_hiv\_transmis\_rate 14.034151 233.26366  
## per\_child\_under\_poverty 11.909348 218.73305

# Testing the model and calculating MSE  
yhat\_rf <- predict(rf\_hiv, newdata = hiv\_df\_boot[-train, ])  
cat("MSE:", mean((yhat\_rf - hiv\_test)^2), "\n")

## MSE: 0.118209

**Boosting**  
The regression tree created with boosting has an MSE of 0.1268387.

# Creating the model  
set.seed(123)  
boost\_hiv <- gbm(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train, ], distribution = "gaussian", n.trees = 5000, interaction.depth = 4)  
summary(boost\_hiv)



## var rel.inf  
## aids\_death\_rate aids\_death\_rate 80.0433314  
## life\_exp life\_exp 7.6051348  
## youth\_literacy\_rate youth\_literacy\_rate 3.7877096  
## pop\_growth\_rate pop\_growth\_rate 2.2789846  
## urban\_pop urban\_pop 1.9165094  
## mother\_child\_hiv\_transmis\_rate mother\_child\_hiv\_transmis\_rate 1.5101233  
## fertility\_rate fertility\_rate 1.2074696  
## infant\_mortality\_rate infant\_mortality\_rate 0.9728071  
## per\_child\_under\_poverty per\_child\_under\_poverty 0.6779302

# Testing the model and calculating the MSE  
yhat\_boost <- predict(boost\_hiv, newdata = hiv\_df\_boot[-train, ], n.trees = 5000)  
cat("MSE:", mean((yhat\_boost - hiv\_test)^2), "\n")

## MSE: 0.1268387

**Overview of Models**

# Table with models and relative MSEs  
quantitative\_overview <- data.frame(  
 Method = c("Best Subset", "Forward Selection", "Backward Selection", "Ridge Regression", "Lasso", "PLS", "Regression Tree", "Bagging", "Random Forests", "Boosting"),   
 MSE = c(1.116786, 1.116785, 1.116785, 1.655353, 1.586618, 1.587266, 0.8744749, 0.07755276, 0.118209, 0.1268387)  
)  
quantitative\_table <- kable(quantitative\_overview, "markdown") %>%   
 kable\_styling(full\_width = FALSE) %>%  
 column\_spec(1, bold = TRUE)

## Warning in kable\_styling(., full\_width = FALSE): Please specify format in  
## kable. kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

## Warning in column\_spec(., 1, bold = TRUE): Please specify format in kable.  
## kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

quantitative\_table

| Method | MSE |
| --- | --- |
| Best Subset | 1.1167860 |
| Forward Selection | 1.1167850 |
| Backward Selection | 1.1167850 |
| Ridge Regression | 1.6553530 |
| Lasso | 1.5866180 |
| PLS | 1.5872660 |
| Regression Tree | 0.8744749 |
| Bagging | 0.0775528 |
| Random Forests | 0.1182090 |
| Boosting | 0.1268387 |

# Section 3: How can the indicators be used to predict AIDS deaths?

**Principal Components Regression**  
A PCR model on the data has an MSE of 52.88973, and explains 84.87% of the variance.

# Creating x and y vectors  
x <- model.matrix(aids\_death\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + hiv\_infection\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, hiv\_df\_boot)[, -1]  
y <- hiv\_df\_boot$aids\_death\_rate  
  
# Creating test and training sets  
set.seed(123)  
train <- sample(1:nrow(x), nrow(x) / 2)  
test <- (-train)  
y.test <- y[test]  
  
# Creating the model  
set.seed(123)  
pcr\_aids <- pcr(aids\_death\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + hiv\_infection\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, scale = TRUE, validation = "CV")  
summary(pcr\_aids)

## Data: X dimension: 1233 9   
## Y dimension: 1233 1  
## Fit method: svdpc  
## Number of components considered: 9  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 16.3 15.25 8.038 7.651 7.630 6.414 6.340  
## adjCV 16.3 15.25 8.034 7.647 7.628 6.409 6.336  
## 7 comps 8 comps 9 comps  
## CV 6.309 6.280 6.239  
## adjCV 6.305 6.276 6.235  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps  
## X 56.87 72.97 81.72 88.45 93.10 96.70 98.87  
## aids\_death\_rate 12.52 76.04 78.29 78.58 84.87 85.23 85.39  
## 8 comps 9 comps  
## X 99.72 100.00  
## aids\_death\_rate 85.53 85.74

# Determining the best M value and calculating MSE  
validationplot(pcr\_aids, val.type = "MSEP")



# The lowest cross-validation error occurs when M = 5 partial least squares directions are used.   
pcr\_pred\_aids <- predict(pcr\_aids, x[test, ], ncomp = 5)  
pcr\_fit\_aids <- pcr(y ~ x, scale = TRUE, ncomp = 5)  
summary(pcr\_fit\_aids)

## Data: X dimension: 1233 9   
## Y dimension: 1233 1  
## Fit method: svdpc  
## Number of components considered: 5  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps  
## X 56.87 72.97 81.72 88.45 93.10  
## y 12.52 76.04 78.29 78.58 84.87

coef(pcr\_fit\_aids)

## , , 5 comps  
##   
## y  
## infant\_mortality\_rate 1.4753177  
## fertility\_rate 0.7749042  
## life\_exp -4.4565081  
## pop\_growth\_rate 0.6232586  
## urban\_pop 0.3087891  
## youth\_literacy\_rate 2.3969200  
## hiv\_infection\_rate 11.7871813  
## mother\_child\_hiv\_transmis\_rate -1.9731541  
## per\_child\_under\_poverty -0.2372773

cat("MSE:", mean((pcr\_pred\_aids - y.test)^2), "\n")

## MSE: 52.88973