STAT 451 Project: “Personal Key Indicators of Heart Disease”

Charles Hwang

4/29/2022

Charles Hwang

Dr. Perry

STAT 451-001

29 April 2022

Introduction

The dataset for this project is the “Personal Key Indicators of Heart Disease” dataset on Kaggle (https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease). The goal of this project is to fit the best possible tree/random forest to the dataset to predict whether or not a respondent has heart disease based on the given variables. The nature of machine learning makes the selection of the “best” possible tree/random forest subjective, but several variables like cross-validation error, number of splits/terminal nodes, etc. can help guide this process. Unlike the Titanic dataset on Kaggle, this dataset is not part of a competition and there is no process for uploading submissions. There are also no defined training and test partitions of the dataset and users wishing to perform any machine learning must create them on their own. However, there are subpages where it appears code “notebooks” or attachments can be uploaded (by clicking the “New Notebook” button or by navigating to the “Discussion” subpage and clicking the “New Topic” button), so I also uploaded my project to Kaggle as a notebook (https://www.kaggle.com/code/charleshwang/stat-451-project).

Dataset

The dataset was compiled from annual survey data of 401,958 adults with 279 variables taken by the Centers for Disease Control and Prevention (CDC) in 2020. The dataset contains 319,795 observations and 18 variables: (1) heart disease (binary)[1], (2) body mass index (BMI), (3) smoking (binary), (4) alcohol consumption (binary), (5) stroke (binary), (6) numbers of days of poor physical and (7) mental health in the last 30 days (discrete), (8) difficulty walking (binary), (9) sex (binary), (10) age category (18-24, 25-29, 30-34, . . . , 75-79, 80 or older), (11) race (white, black, Hispanic, Asian, American Indian/Alaska native, other), (12) diabetes (no, borderline, during pregnancy, yes), (13) physical activity in the last 30 days (binary), (14) general health (Likert), (15) average hours of sleep (discrete), (16) asthma (binary), (17) kidney disease (binary), and (18) skin cancer (binary). According to the webpage on Kaggle, there are no mismatched or missing data out of all 5,756,310 cells. It is worth briefly mentioning that since the data are from 2020, there are very little to no instances of myocardial infarction caused by side effects from any vaccine or immunization against COVID-19 among the respondents in the dataset (as has been discussed in the news/media) and there are no variables for any vaccine or injection in the dataset.

In visualizing the data, Kaggle is helpful in this regard as the webpage provides pie charts for Boolean (yes/no) variables (variables 1, 3-5, 8, 13, and 16-18), proportions of the two most common levels for non-Boolean factor variables (variables 9-12, and 14), and histograms for numeric variables (variables 2, 6-7, and 15). Kaggle also provides the number of levels for non-Boolean factor variables and the mean, standard deviation,

[1] The official definition for the HeartDisease variable as listed on the Kaggle webpage states an affirmative response is recorded for “Respondents that have ever reported having coronary heart disease (CHD) or myocardial infarction (MI)”.

and interquartile range (IQR) for numeric variables. The pie chart and a table I created show approximately 8.6 percent of, or 27,373, respondents in the data were reported as being diagnosed with heart disease. With there being 14 factor variables, I did not find it practical or meaningful for analysis to compare all $\binom{14}{2} = 91$ pairs of variables, so I did not print any frequency tables.

I created a boxplot for (2) BMI and saw it had several unreasonable outliers with a maximum of 94.85 (which is not possible as 40+ is defined by the CDC as "class III obesity"). I inferred this may be due to inadvertently entering weight, either in pounds or kilograms, instead of BMI ($\frac{kg}{m^2}$). I also created boxplots for the other three numeric variables and saw there were several values considered statistical outliers. It is possible to have 6+ days of poor physical health and/or 8+ for days of poor mental health in the last 30 days, but it seemed unreasonable to sleep for 11+ hours *on average*. I inferred this could be from misunderstanding or misinterpreting the question (hours of sleep last night instead of on average, inverse question (hours awake), unserious responses, etc.). The histograms on Kaggle also showed a local mode with 19,505 responses for (6-7) days of poor mental and physical health in the last 30 days at 30, which could similarly be from misunderstanding the question as the inverse or a reactionary/knee-jerk pessimistic response.

I initially produced a plot matrix of the numeric variables, but the large sample size made it impractical to interpret as almost all points overlapped with one another making it difficult to visualize any trends. In place of this, I produced the correlation and covariance matrices to see the relationships between the numeric variables. The greatest linear correlation was between (6) days of poor physical and (7) mental health ($r = 0.2879867$), which indicated there is little to no linear correlation between any two of the numeric variables. This could be due to the large sample size, but this made sense because the domain and range of the plots (ranges of the variables) meant a linear model was likely not the best fit. Either way, the $\binom{4}{2} = 6$ values of r in relation to each other also made sense intuitively. The covariance matrix similarly showed low levels of covariance between any pair of numeric variables except days of poor physical and mental health ($cov(PH, MH) = 18.21541$). All univariate and bivariate analyses can be found in the (1) Data Cleaning and Univariate and Bivariate Analyses section of the Appendix.

Methods

I first read the data into RStudio and changed variables 1, 3-5, 8-14, and 16-18 to factors. For the (14) general health variable, the levels were assigned in alphabetical order and I recoded them to match the standard Likert scale. After conducting the above univariate and bivariate analyses, I split the data into training and test datasets with the `sample()` function and an 80/20 split. Since there was no missing data, there was no need to impute the data with the `mice` package or any other method.

I grew a classification tree using the `tree` package, available in the (2) Classification Tree section of the Appendix. The tree showed (10) age and (14) general health to be the only variables contributing to splits. There was a clear split in age with the tree splitting 18-54 from 55+. The general health variable then split the 18-54 branch at responses of 3 through 5 (good, very good, and excellent) and 1 and 2 (poor and fair) and the 55+ branch at responses of 4 and 5 and 1 through 3. The final split was among this branch of 1 through 3, further splitting general health by separating 3 from 1 and 2. These splits appear to make sense intuitively, with younger Americans less likely than older Americans to be diagnosed with heart disease. The difference in secondary splits indicates a higher threshold of personal general health rating for older Americans for predicting a lower probability of heart disease than for younger Americans, and the tertiary branch similarly predicts older Americans with "good" general health have a lower probability than those with "poor" or "fair" general health. However, it is interesting that this tree did not consider any other variables to be significant in predicting heart disease.

We can also see that the tree predicted a probability lower than 0.5 for all of the five terminal nodes. This would mean the prediction for every respondent in the test dataset would be classified as 0 ("No") regardless of any variables. Only approximately 8.4 percent of respondents in the test dataset have been diagnosed with heart disease, so although this would result in an accuracy rate of approximately $1 - 8.4\% = 91.6$ percent and a 0 percent false positive rate, it would also result in a 100 percent false negative rate which would likely not be the "best" model for predictions in practice. If I wanted this result I would simply code a string of zeroes and submit it as my predictions.

Thus, I grew a random forest with the default of 500 trees on the data, available in the (3) Random Forest section of the Appendix. The out-of-bag cross-validation error was approximately 8.57 percent. After making predictions on the test dataset, the accuracy rate was approximately 91.7 percent and the false positive rate was approximately 0.6 percent, but the false negative rate was still high at approximately 91.4 percent. However, it appeared to perform better than a trivial string of zeroes overall with the accuracy rate being slightly higher and the false negative rate being lower. I also plotted the error as the number of trees increased and created a variable importance plot to visualize the influential variables in the random forest. We can see the error stops decreasing after about 100 trees. (2) BMI is the most important variable in the random forest, followed by (10) age and (15) sleep time. (14) General health and (6-7) days of poor physical and mental health appear to round out the influential variables.

Results

The accuracy rate of the final "best" model (what the Kaggle score would have been if this dataset were part of a competition), the random forest, is approximately 91.7 percent. The false positive rate is approximately 0.6 percent and the false negative rate is approximately 91.4 percent. We can see when the proportion of affirmative responses is low for binary response variables that a high false negative rate seems to be unavoidable even in robust models. This makes sense intuitively when looking at how each are calculated.

Conclusions/Future Work

Since this dataset is not part of a competition, there is no official score displayed in Kaggle for my predictions or any leaderboard to compare them with other users. However, it appears a 91.7 percent accuracy rate is relatively good in practice. Of course, the purpose of prediction would have influence on the model. As previously mentioned in the "Methods" section, if we wanted to minimize the false positive rate, we would simply say no respondent has heart disease, and conversely, if we wanted to minimize the false negative rate, we would simply say every respondent has heart disease. The tradeoff between the two is similar to the bias-variance tradeoff (https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff) encountered when fitting models.

Future work if given additional time could include fitting additional trees from other packages like `rpart` or using different seed values and reviewing the dataset to see which observations were predicted incorrectly. Since the dataset is large, some observations may have been outliers or incorrectly classified in one or more of the splits in the tree for various reasons. Another analysis could be experimenting with other types of prediction models like gradient boosted models (GBM).

In conclusion, it appears this project was a success in using machine learning to predict the presence of heart disease in Americans. The data are relatively recent which means there may not be a lot of prior statistical analysis to this degree.

Appendix

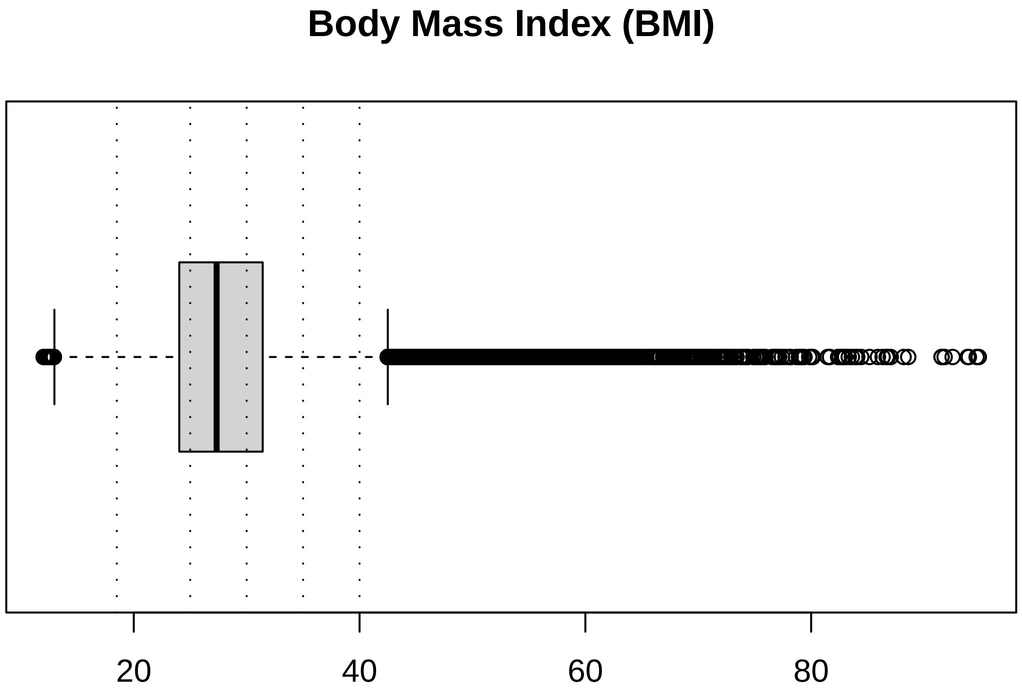
(1) Data Cleaning and Univariate and Bivariate Analyses

```
rm(list=ls())
heart<-read.csv("/Users/newuser/Desktop/Notes/Graduate/STAT 451 - Nonparametric Statistical Methods/hea
heart$HeartDisease<-as.factor(heart$HeartDisease)
heart$Smoking<-as.factor(heart$Smoking)
heart$AlcoholDrinking<-as.factor(heart$AlcoholDrinking)
heart$Stroke<-as.factor(heart$Stroke)
heart$DiffWalking<-as.factor(heart$DiffWalking)
heart$Sex<-as.factor(heart$Sex)
heart$AgeCategory<-as.factor(heart$AgeCategory)
heart$Race<-as.factor(heart$Race)
heart$Diabetic<-factor(heart$Diabetic,labels=c("No","BL","Yes","YesPreg")) # Renaming levels
```

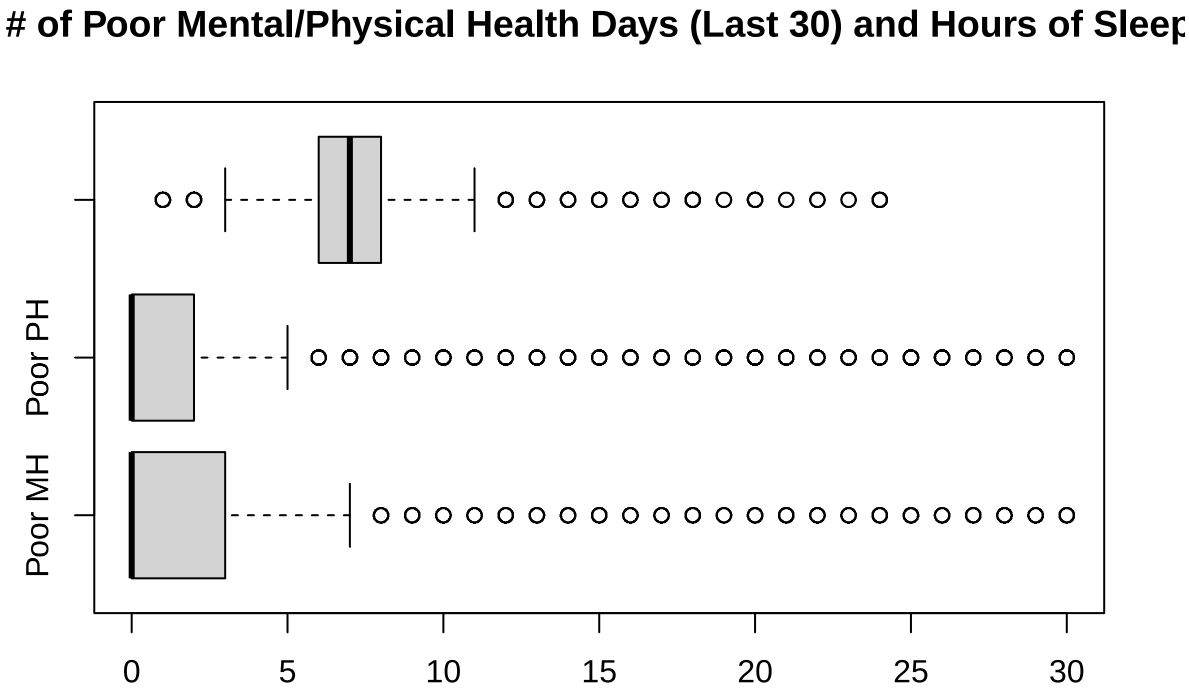
```
heart$PhysicalActivity<-as.factor(heart$PhysicalActivity)
heart$GenHealth<-factor(heart$GenHealth,labels=c(5,2,3,1,4)) # Recoding to Likert scale (1-5)
heart$Asthma<-as.factor(heart$Asthma)
heart$KidneyDisease<-as.factor(heart$KidneyDisease)
heart$SkinCancer<-as.factor(heart$SkinCancer)
table(heart$HeartDisease)
```

```
##
##     No    Yes
## 292422  27373
```

```
boxplot(heart$BMI,main="Body Mass Index (BMI)",horizontal=TRUE)
abline(v=c(18.5,25,30,35,40),lty="17")
```



```
boxplot(heart$MentalHealth,heart$PhysicalHealth,heart$SleepTime,names=c("Poor MH","Poor PH","Sleep (hrs
```



```
cor(heart[,c("BMI","PhysicalHealth","MentalHealth","SleepTime")])
```

```
##                        BMI PhysicalHealth MentalHealth   SleepTime
## BMI             1.00000000     0.10978754   0.06413057 -0.05182225
## PhysicalHealth  0.10978754     1.00000000   0.28798667 -0.06138663
## MentalHealth    0.06413057     0.28798667   1.00000000 -0.11971679
## SleepTime      -0.05182225    -0.06138663  -0.11971679  1.00000000
```

```
cov(heart[,c("BMI","PhysicalHealth","MentalHealth","SleepTime")])
```

```
##                       BMI PhysicalHealth MentalHealth  SleepTime
## BMI            40.4000098      5.5482673     3.242716 -0.4730027
## PhysicalHealth  5.5482673     63.2160186    18.215412 -0.7008805
## MentalHealth    3.2427156     18.2154115    63.285767 -1.3676175
## SleepTime      -0.4730027     -0.7008805    -1.367618  2.0621163
```

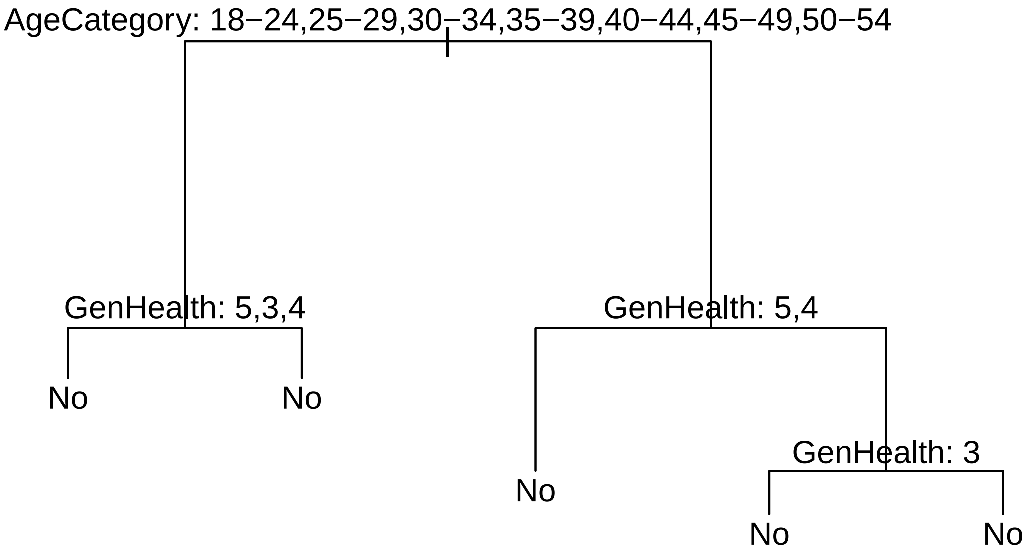
```
set.seed(2904)
s<-sort(sample(nrow(heart),nrow(heart)*.8))
train<-heart[s,]
test<-heart[-s,]
rm(s)
```

(2) Classification Tree

```
library(tree)
set.seed(2904)
tree2<-tree(HeartDisease~.,method="class",data=train)
tree2
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
## 
##  1) root 255836 149800 No ( 0.91416 0.08584 )
##    2) AgeCategory: 18-24,25-29,30-34,35-39,40-44,45-49,50-54 116539  25740 No ( 0.97675 0.02325 )
##      4) GenHealth: 5,3,4 105152  16080 No ( 0.98534 0.01466 ) *
##      5) GenHealth: 2,1 11387   7527 No ( 0.89751 0.10249 ) *
##    3) AgeCategory: 55-59,60-64,65-69,70-74,75-79,80 or older 139297 111900 No ( 0.86178 0.13822 )
##      6) GenHealth: 5,4 70501  35190 No ( 0.93158 0.06842 ) *
##      7) GenHealth: 2,3,1 68796  70670 No ( 0.79026 0.20974 )
##       14) GenHealth: 3 43295  37770 No ( 0.84211 0.15789 ) *
##       15) GenHealth: 2,1 25501  31060 No ( 0.70225 0.29775 ) *
```

```
plot(tree2)
text(tree2,pretty=0)
```



```
tree2$frame$yprob
```

```
##              No        Yes
##  [1,] 0.9141559 0.08584406
##  [2,] 0.9767546 0.02324544
##  [3,] 0.9853355 0.01466449
##  [4,] 0.8975147 0.10248529
##  [5,] 0.8617845 0.13821547
##  [6,] 0.9315754 0.06842456
##  [7,] 0.7902640 0.20973603
##  [8,] 0.8421065 0.15789352
##  [9,] 0.7022470 0.29775303
```

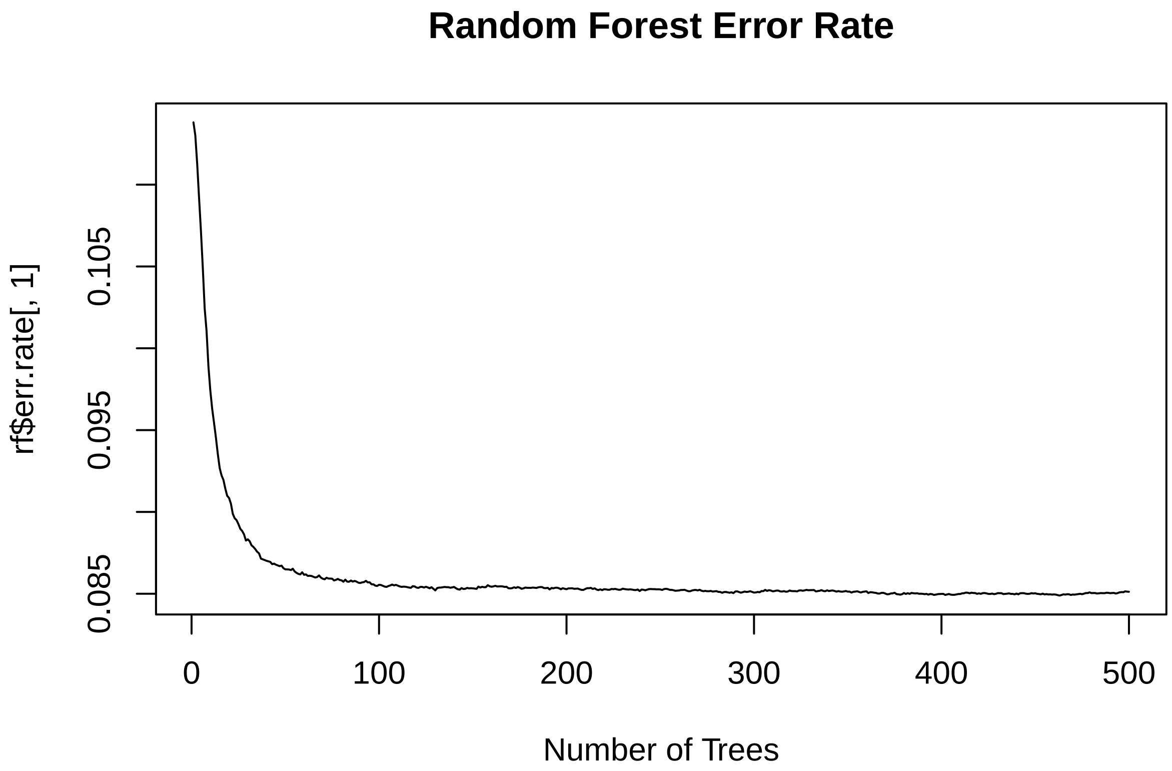
(3) Random Forest

```
library(randomForest)
set.seed(2904)
rf<-randomForest(HeartDisease~.,ntree=500,data=train)
rf
```

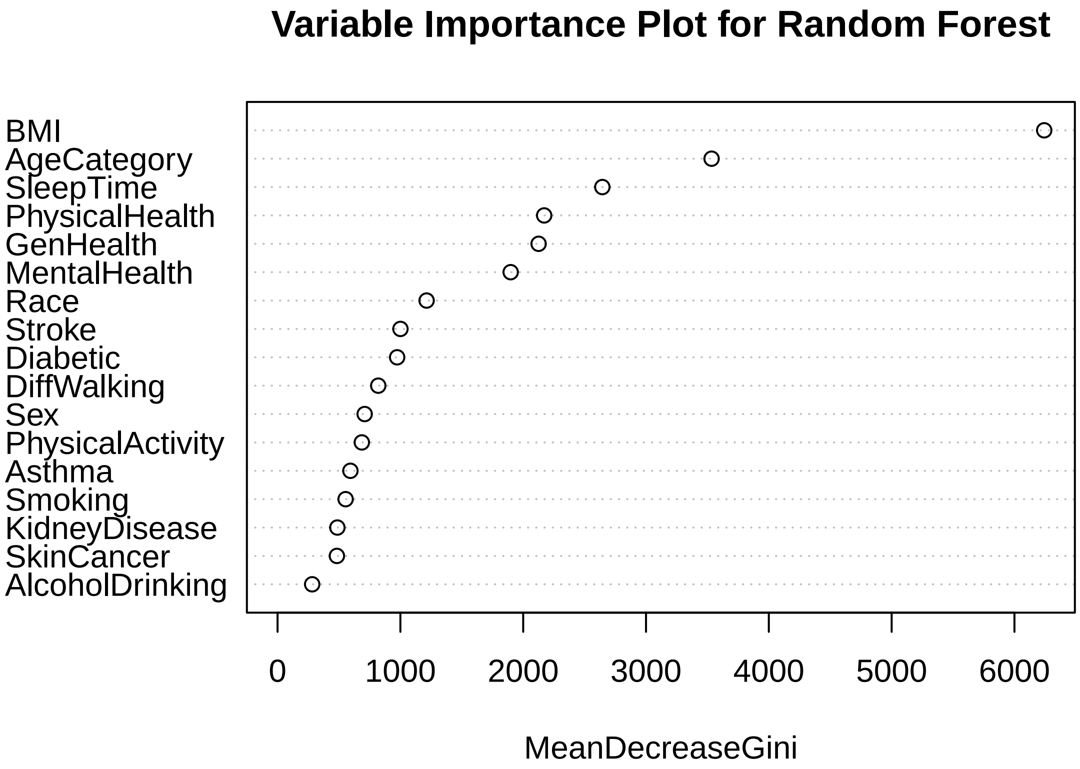
```
##
## Call:
##  randomForest(formula = HeartDisease ~ ., data = train, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
```

```
##         OOB estimate of  error rate: 8.51%
## Confusion matrix:
##         No  Yes class.error
## No  232228 1646 0.007037978
## Yes  20131 1831 0.916628722
```

```
plot(rf$err.rate[,1],type="l",main="Random Forest Error Rate",xlab="Number of Trees")
```



```
varImpPlot(rf,main="Variable Importance Plot for Random Forest")
```



```
rfpred<-predict(rf,test,type="class")
rft<-table(test$HeartDisease,rfpred)
rft
```

```
##      rfpred
##           No   Yes
##   No   58119   429
##   Yes   4955   456
```

```
sum(diag(rft))/nrow(test)
```

```
## [1] 0.9158211
```

```
rft["No","Yes"]/sum(rft["No",])
```

```
## [1] 0.007327321
```

```
rft["Yes","No"]/sum(rft["Yes",])
```

```
## [1] 0.9157272
```

```
test$HeartDiseaseRF<-rfpred
write.csv(test[,c("HeartDisease","HeartDiseaseRF")],"Heart Disease Predictions") # Export CSV
```