# Heart Disease Prediction

Lam Vu

5/28/2019

## INTRODUCTION

This dataset contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

**Attribute Information:**
1. **age** in year
2. **sex** (F=0, M=1)
3. **cp** - chest pain type (4 values)
4. **trestbps** - resting blood pressure
5. **chol** - serum cholestoral in mg/dl
6. **fbs** - fasting blood sugar (value 0: <= 120 mg/dl, value 1: > 120 mg/dl)
7. **restecg** - resting electrocardiographic results (values 0,1,2)
8. **thalach** - maximum heart rate achieved
9. **exang** - exercise induced angina (value 1: yes; value 0: no)
10. **oldpeak** - ST depression induced by exercise relative to rest
11. **slope** of the peak exercise ST segment
12. **ca** - number of major vessels (0-3) colored by flourosopy
13. **thal** Thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)
14. **target** heart disease present = 1, healthy = 0

The **objectives** of this project are to gain insights with the **heart** dataset through exploration, and visualization, and using different modeling approach to predict present of heart disease.

### Dataset

The **heart** dataset can be view or download at https://www.kaggle.com/ronitf/heart-disease-uci. After downloaded the dataset, it can be examine in the following codes. File's name call is depending on its local location.

```
library(lattice)
library(ggplot2)
library(readr)
library(caret)
library(tidyr)
library(dplyr)
library(corrplot)
```

```
###file's name call is depending on its location
heart <- read_csv("…/heart.csv")

str(heart)

## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 303 obs. of  14 v
ariables:
##  $ age     : num  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : num  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : num  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: num  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : num  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : num  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : num  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : num  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : num  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : num  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : num  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   sex = col_double(),
##   ..   cp = col_double(),
##   ..   trestbps = col_double(),
##   ..   chol = col_double(),
##   ..   fbs = col_double(),
##   ..   restecg = col_double(),
##   ..   thalach = col_double(),
##   ..   exang = col_double(),
##   ..   oldpeak = col_double(),
##   ..   slope = col_double(),
##   ..   ca = col_double(),
##   ..   thal = col_double(),
##   ..   target = col_double()
##   .. )
```

Data contains *no* missing values

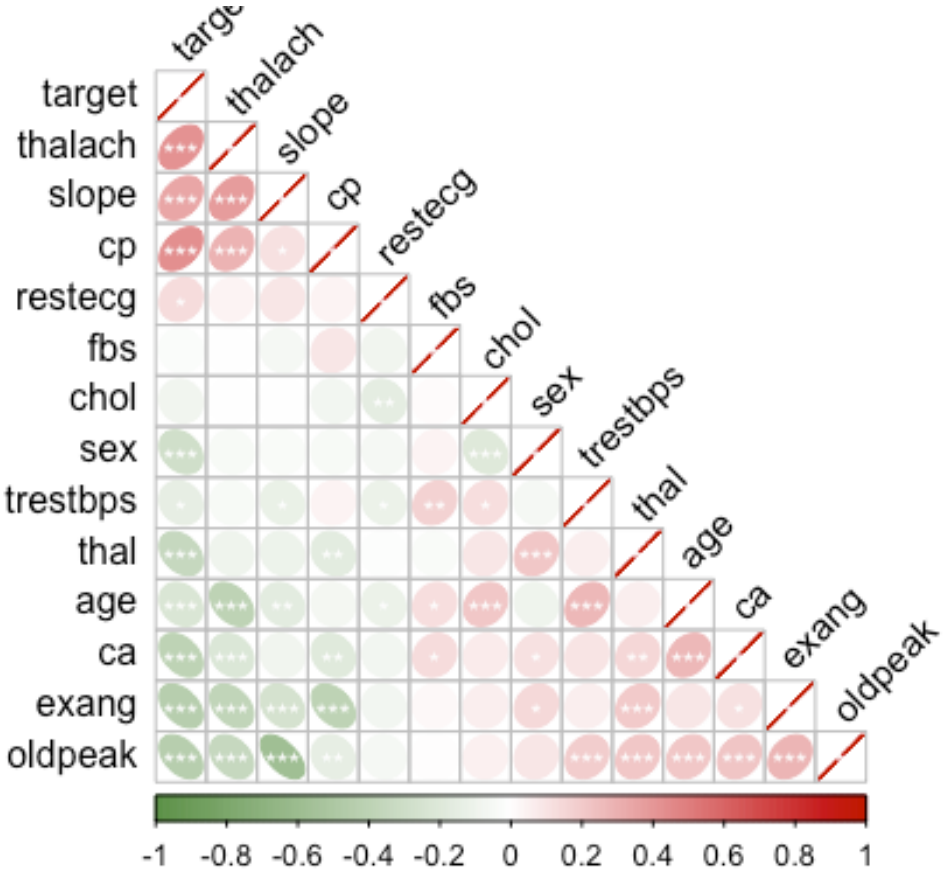
```
#does data contain any missing values
sum(is.na(heart))

## [1] 0
```

## DATA ANALYSIS

### Correlation matrix

This is a graphical display of a correlation matrix and confidence interval of 13 predictors to target (aka heart condition). Positive correlations are displayed in red and negative correlations in green color. Color intensity and the size of the ellipse are proportional to the correlation coefficients and confidence interval. It is reordered by the first principal component "FPC".
Also, level of significance is denoted as stars: *** 0.001, ** 0.01, * 0.05.




### Preprocessing

Converting type to factor and columns from numeric to categorical is necessary to illustrate relationship between different predictors to heart condition well. Preprocessed data as followed:

```
###converting type to factor
heart$target<-as.factor(heart$target)
heart$sex<-as.factor(heart$sex)
heart$cp<-as.factor(heart$cp)
heart$fbs<-as.factor(heart$fbs)
heart$exang<-as.factor(heart$exang)
```

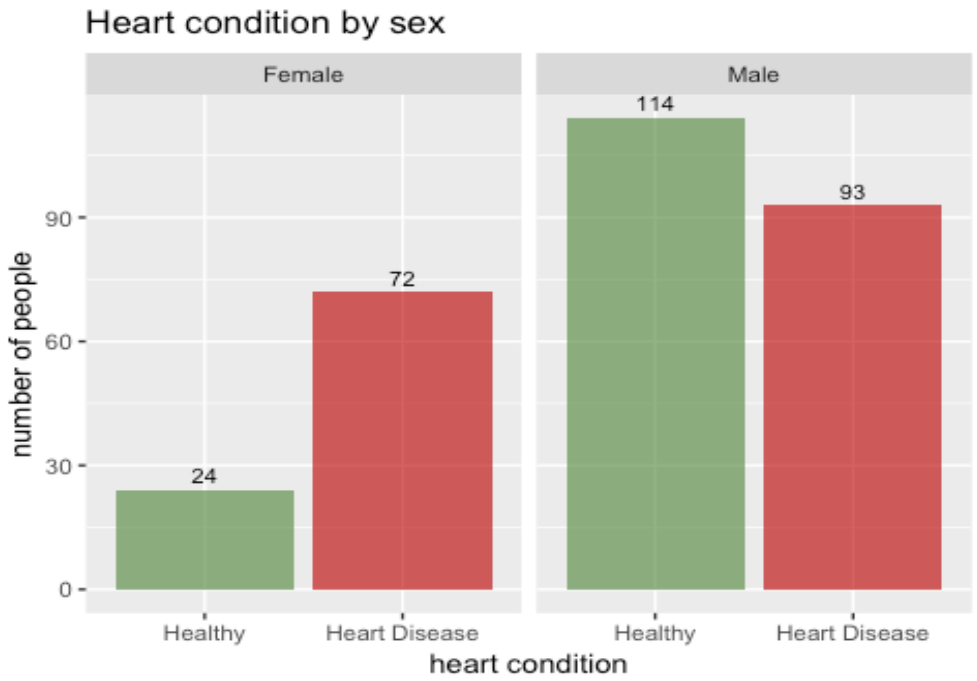
```r
heart$restecg<-as.factor(heart$restecg)
heart$slope<-as.factor(heart$slope)
heart$thal<-as.factor(heart$thal)

##converting columns from numeric to categorical
levels(heart$sex)[levels(heart$sex)==0] <- "Female"
levels(heart$sex)[levels(heart$sex)==1] <- "Male"
levels(heart$fbs)[levels(heart$fbs)==0] <- "Fasting Blood Sugar <= 120"
levels(heart$fbs)[levels(heart$fbs)==1] <- "Fasting Blood Sugar > 120"
levels(heart$thal)[levels(heart$thal)==0] <- "No Thalassemia"
levels(heart$thal)[levels(heart$thal)==1] <- "Normal Thalassemia"
levels(heart$thal)[levels(heart$thal)==2] <- "Fixed Defect Thalassemia"
levels(heart$thal)[levels(heart$thal)==3] <- "Reversible Defect Thalassemia"
levels(heart$target)[levels(heart$target)==0] <- "Healthy"
levels(heart$target)[levels(heart$target)==1] <- "Heart Disease"
levels(heart$exang)[levels(heart$exang)==1] <- "Exercise Induced Angina"
levels(heart$exang)[levels(heart$exang)==0] <- "No Exercise Induced Angina"
levels(heart$restecg)[levels(heart$restecg)==0] <- "Rest ECG 0"
levels(heart$restecg)[levels(heart$restecg)==1] <- "Rest ECG 1"
levels(heart$restecg)[levels(heart$restecg)==2] <- "Rest ECG 2"
levels(heart$slope)[levels(heart$slope)==0] <- "Peak Excercise ST Slope 0"
levels(heart$slope)[levels(heart$slope)==1] <- "Peak Excercise ST Slope 1"
levels(heart$slope)[levels(heart$slope)==2] <- "Peak Excercise ST Slope 2"

summary(heart)
```

## Visualization

**Table of number of male and female**




| sex | n |
|---|---|
| Female | 96 |
| Male | 207 |

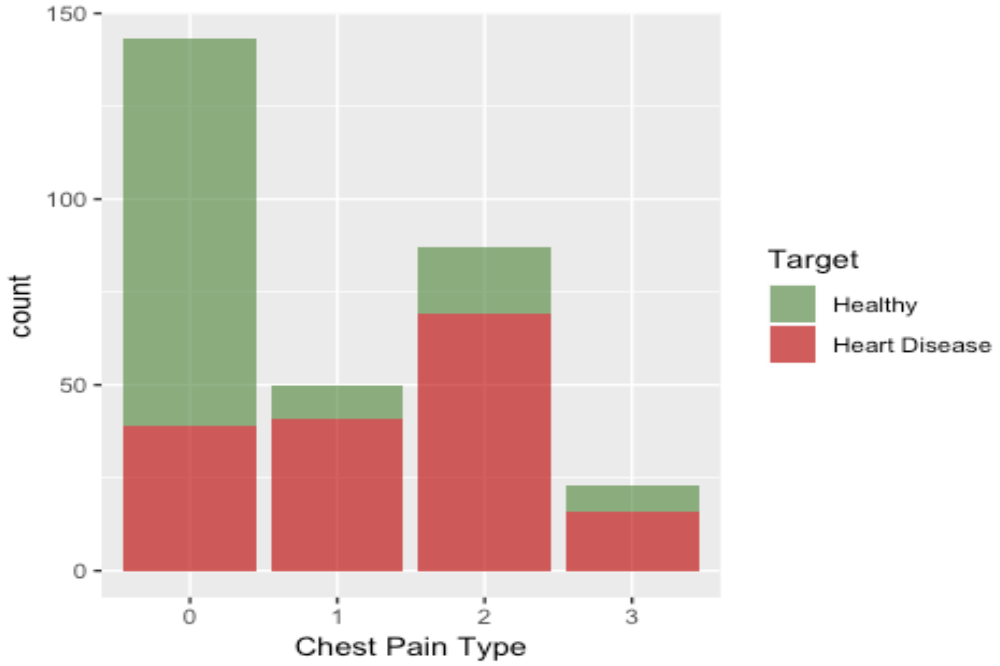
**Heart condition by age**

| target | n | min | max | median | avg |
|---|---|---|---|---|---|
| Healthy | 138 | 35 | 77 | 58 | 56.60145 |
| Heart Disease | 165 | 29 | 76 | 52 | 52.49697 |

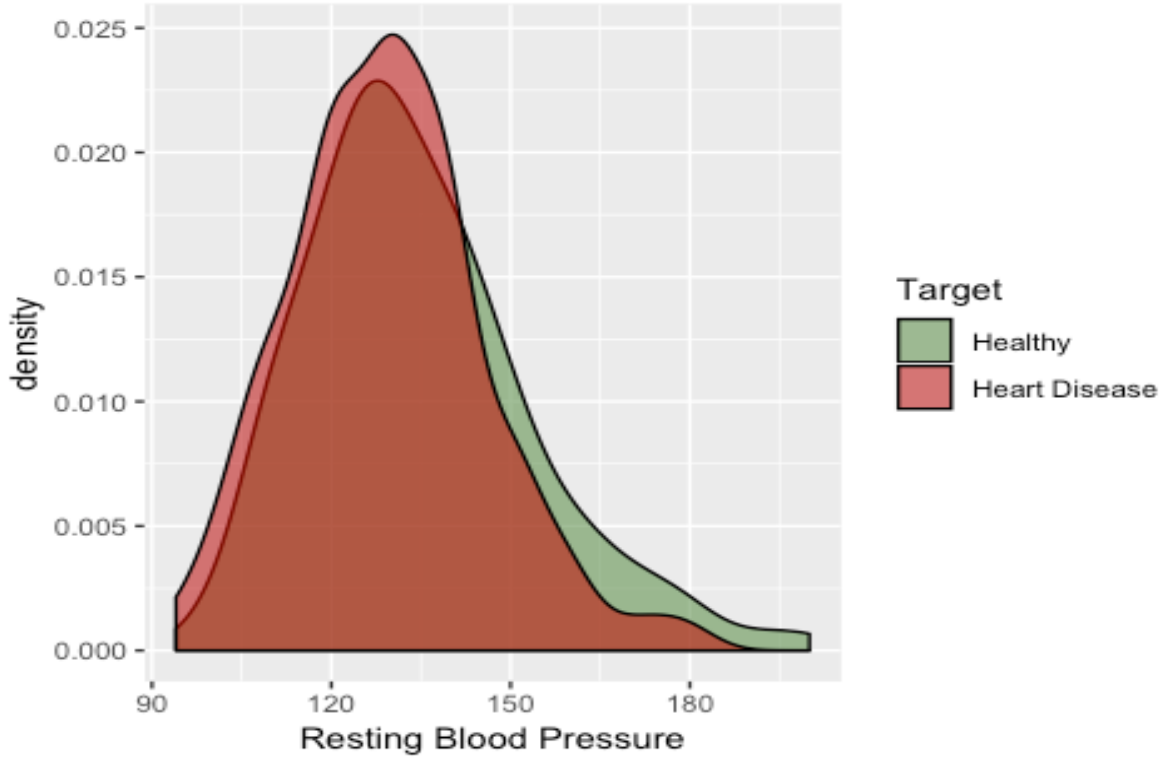

**Table of heart condition associated with different type chest pain**

| cp | target | n |
|---|---|---|
| 0 | Healthy | 104 |
| 0 | Heart Disease | 39 |
| 1 | Healthy | 9 |
| 1 | Heart Disease | 41 |
| 2 | Healthy | 18 |
| 2 | Heart Disease | 69 |
| 3 | Healthy | 7 |
| 3 | Heart Disease | 16 |

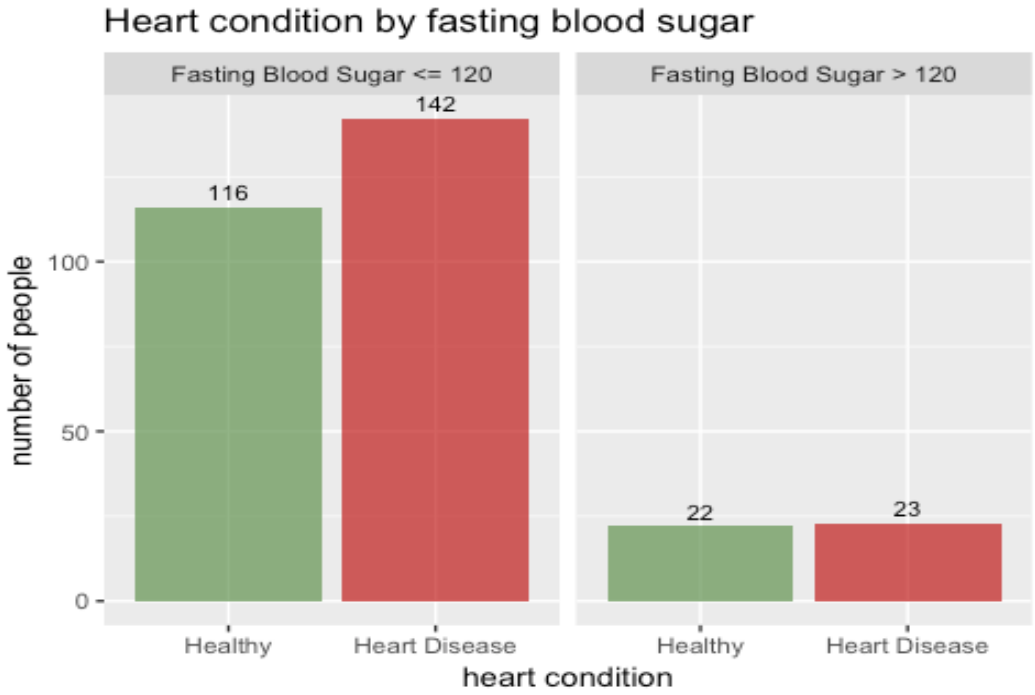
**Chest pain** of any type is *associated* with heart disease can be observed by the table above or graph below.



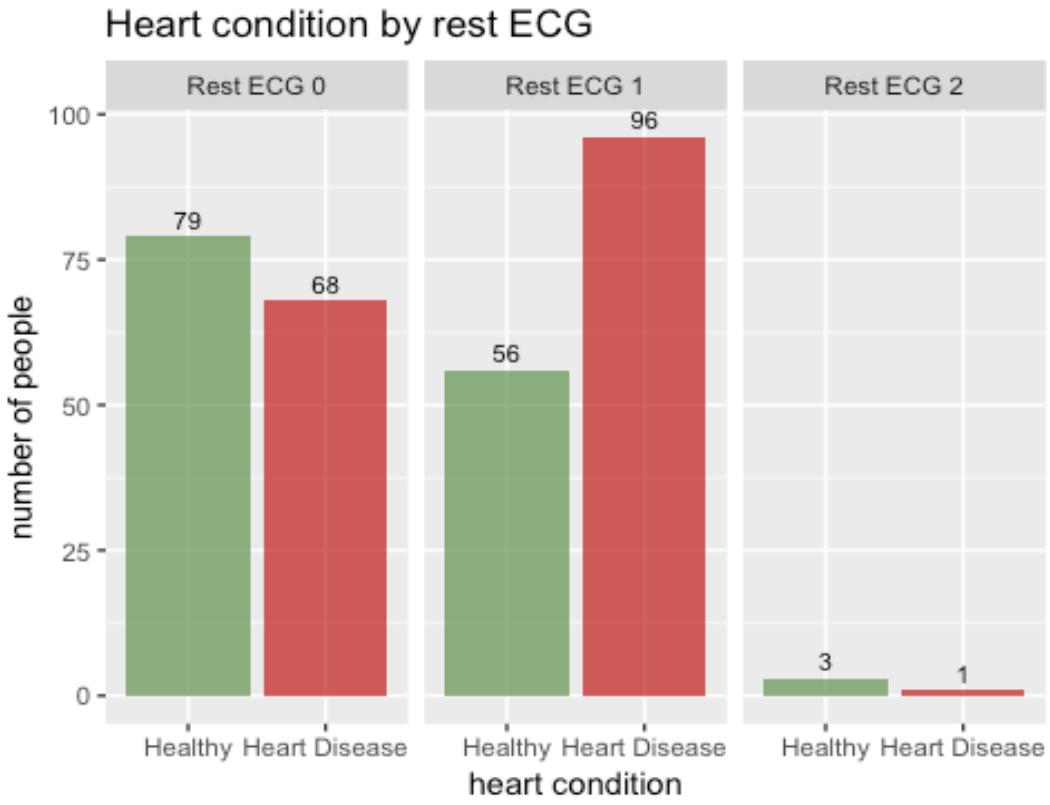

**Distribution between resting blood pressure and heart condition**
There's *no* noticable differences in blood pressure between healthy and heart disease subjects.



**Distribution between fasting blood sugar and heart condition**
The relationship is not significant.

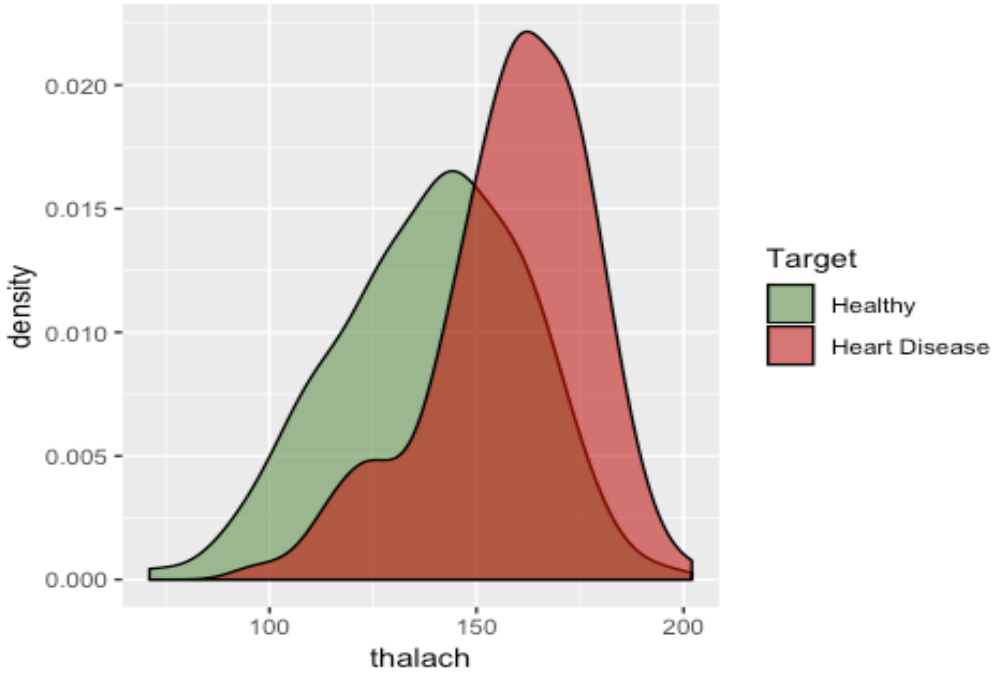



**Distribution between rest ECG and heart condition**
More subjects with rest ECG result 1 has heart disease.



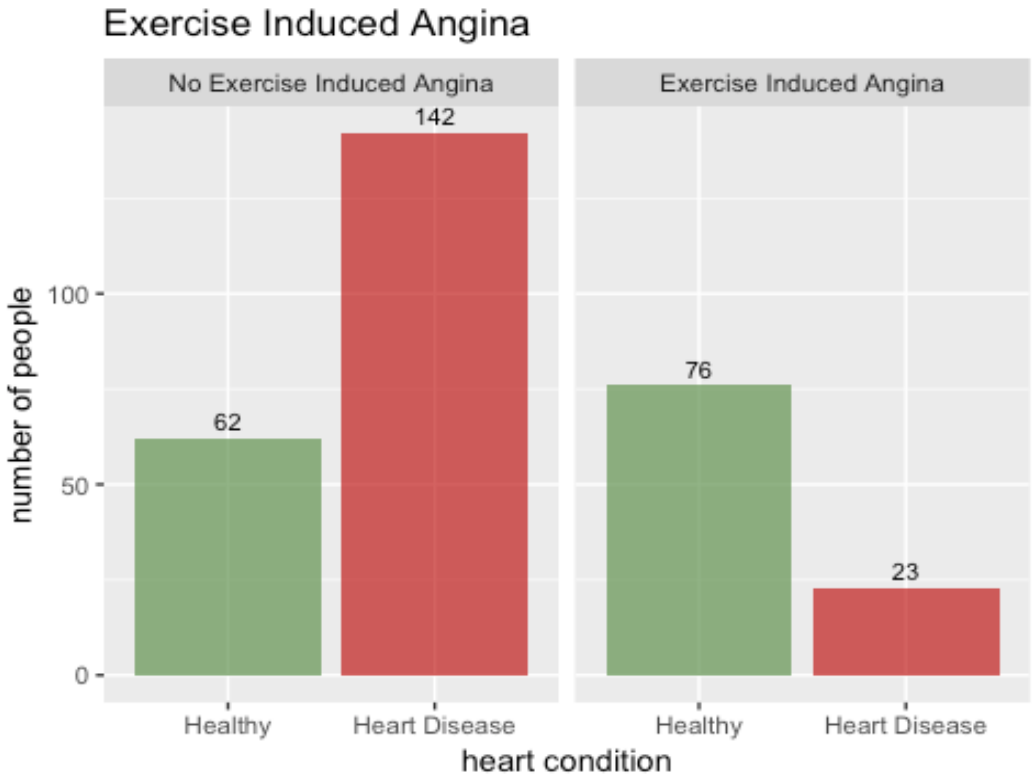
**Distribution between maximum heart rate achieved and heart condition**

Maximum heart rate (**thalac**) is on average higher in subjects with heart disease.



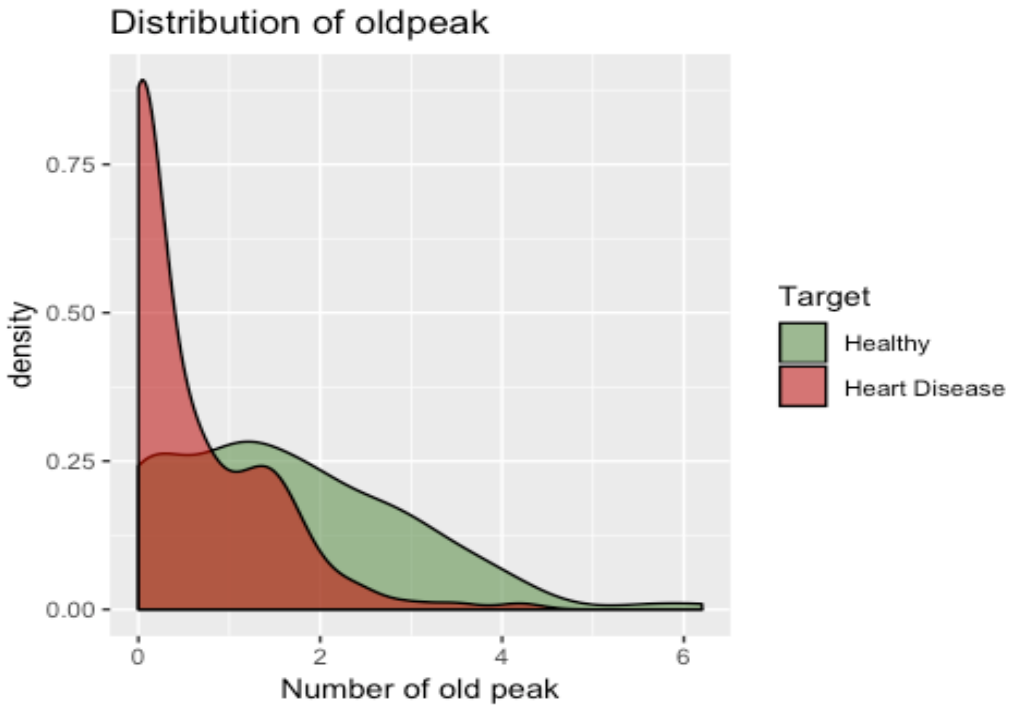

**Distribution between exercise induced angina and heart condition**

More subjects with no exercise induced angina (**exang**) have heart disease.

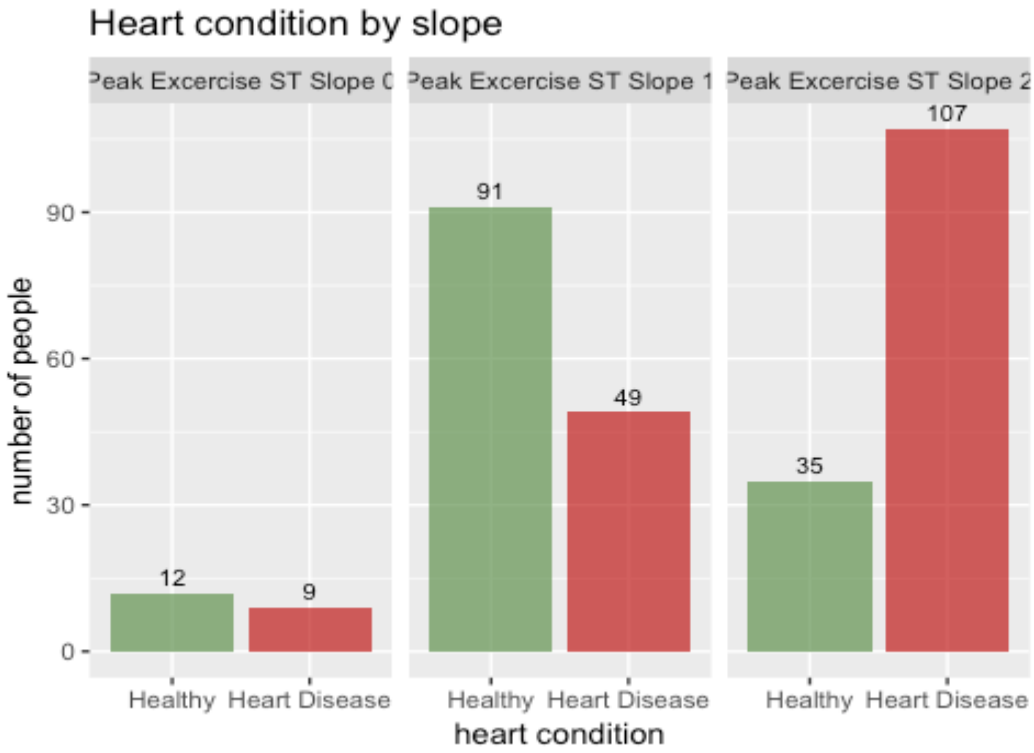


**Distribution between oldpeak exercise and heart condition**

Subject with heart disease has *significant* lower number of peaks of ST depression induced by exercise relative to rest.

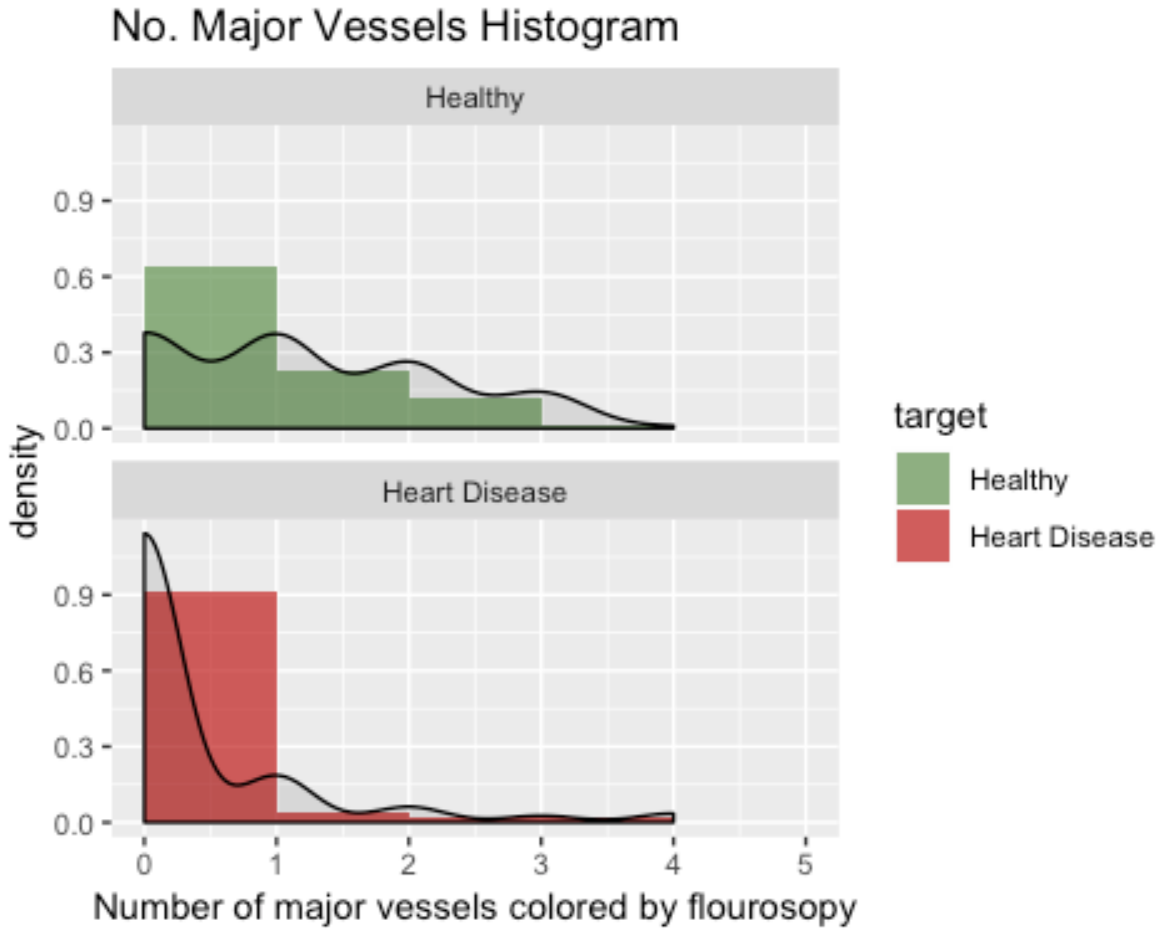



**Distribution between slope and heart condition**

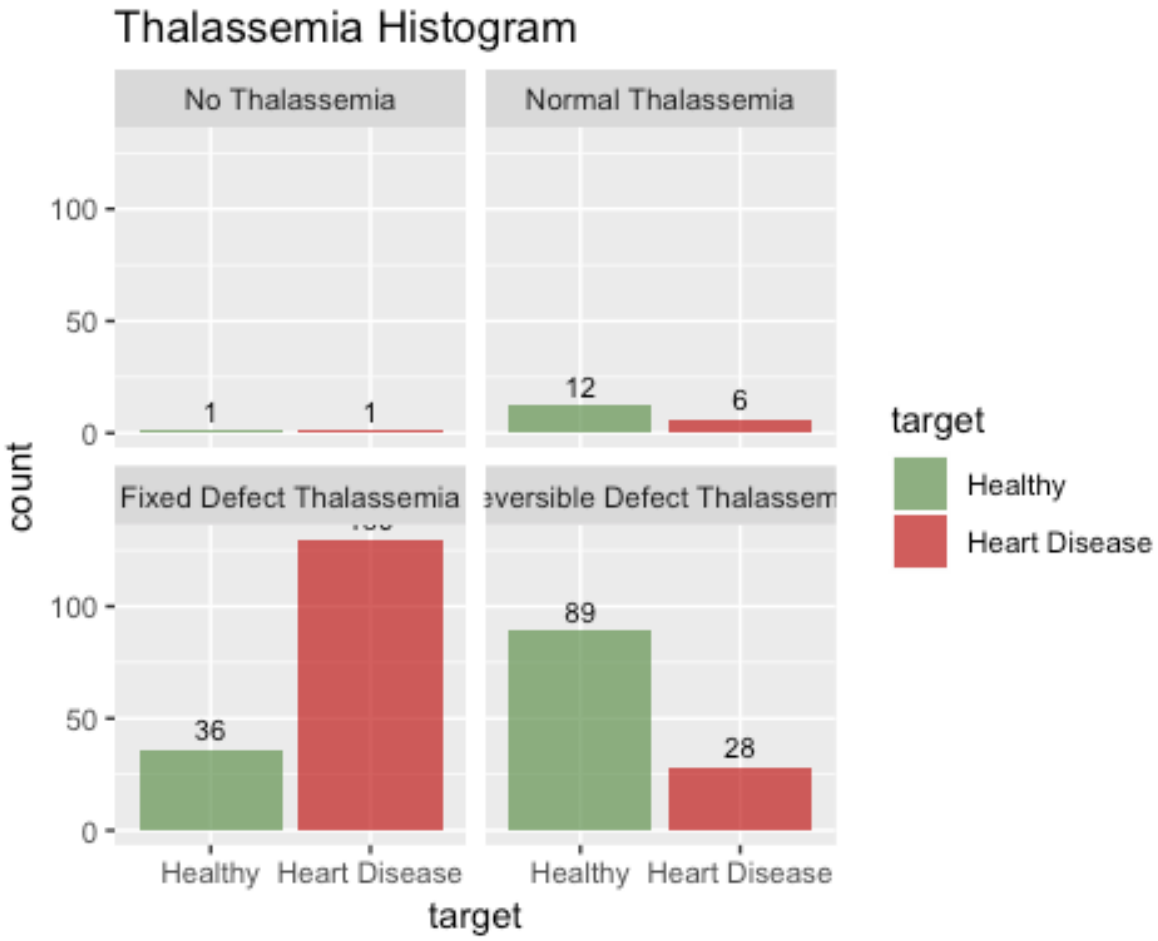


**Distribution between number of major vessels colored by flourosopy (ca) and heart condition**

*Majority* of subjects who have heart disease have *zero* (0) major vessels as observed by fluroscopy.



**Distribution between different type of Thalassemia and heart condition**
Fixed defect thalasemia (thal) has more subjects with heart disease.




## MODELS

### Ensemble of models using *all* variables

The full dataset is splitted into a training set and a testing set. The training set consists of 75% of the total values in the dataset, and the testing set consists of the remaining 25%.

```
#Spliting training set into two parts based on outcome: 75% and 25%
y <- heart$target

set.seed(1)
test_index = createDataPartition(y, times = 1, p=0.75,list=FALSE)
train_set <- heart[test_index,]
valid_set <- heart[-test_index,]
```

Ensemble method is used to capture linear and simple as well non-linear complex relationships in this data. It is done by using seventeen (17) different models and forming an ensemble of seventeen.

```
#############Ensembling with ALL predictors#############################
models <- c("glm", "lda",  "naive_bayes",  "svmLinear", "qda",
            "knn", "kknn", "rf", "ranger",  "wsrf", "Rborist",
            "avNNet", "monmlp", "adaboost", "gbm",
            "svmRadial", "svmRadialCost")

#set.seed(1)
fits <- lapply(models, function(model){
  print(model)
  train(target ~ ., method = model, data = train_set)
})
#fits
names(fits) <- models
```

All the trained models is in a list now. Next, creating a matrix of predictions for the test set.

```
pred <- sapply(fits, function(object)
  predict(object, newdata = valid_set))
dim(pred)

## [1] 75 17
```

**Average**

Accuracy for each model in the test set and the mean accuracy across all models can be computed using the following code:

```
acc <- colMeans(pred == valid_set$target)

acc_results <- data_frame(method = models, acc = acc)
acc_results %>% knitr::kable()
```

| method | acc |
|---|---|
| glm | 0.8533333 |
| lda | 0.8533333 |
| naive_bayes | 0.8400000 |
| svmLinear | 0.7866667 |
| qda | 0.8533333 |
| knn | 0.6400000 |
| kknn | 0.8400000 |
| rf | 0.8400000 |
| ranger | 0.8266667 |
| wsrf | 0.7600000 |

| | |
|---|---|
| Rborist | 0.6266667 |
| avNNet | 0.8133333 |
| monmlp | 0.8400000 |
| adaboost | 0.7733333 |
| gbm | 0.8133333 |
| svmRadial | 0.7733333 |
| svmRadialCost | 0.7733333 |

```
#Result of the mean accuracy across all models.
avg <- mean(acc)
avg

## [1] 0.7945098
```

**Majority Voting**

In majority voting, we'll assign the prediction for the observation as predicted by the majority of models. Since we have seventeen models for a binary classification task, a tie is not possible.

```
#building an ensemble prediction by majority vote and compute the accuracy of
the ensemble.
votes <- rowMeans(pred == "Heart Disease")
y_hat <- ifelse(votes > 0.5, "Heart Disease", "Healthy")

#What is the accuracy of the ensemble?
votes_avg <- mean(y_hat == valid_set$target)
votes_avg

## [1] 0.84

ind <- acc > mean(y_hat == valid_set$target)
sum(ind)
models[ind]
```

Individual methods perform better than the ensemble are **glm, lda, qda**

Using the accuracy estimates obtained from cross validation with the training data then find the mean accuracy of the new estimates.

```
acc_hat <- sapply(fits, function(fit) min(fit$results$Accuracy))
new_mean <- mean(acc_hat)
```

Now, only considering the methods with an estimated accuracy of greater than or equal to the new_mean of 0.7395918 when constructing the ensemble.

```
ind <- acc_hat >= new_mean
sum(ind)

## [1] 14
```

```
models[ind]
```

```
##  [1] "glm"           "lda"           "naive_bayes"   "svmLinear"
##  [5] "qda"           "kknn"          "rf"            "ranger"
##  [9] "wsrf"          "monmlp"        "adaboost"      "gbm"
## [13] "svmRadial"     "svmRadialCost"
```

```
votes <- rowMeans(pred[,ind] == "Heart Disease")
y_hat <- ifelse(votes >=0.5, "Heart Disease", "Healthy")
new_votes_avg <- mean(y_hat == valid_set$target)
new_votes_avg
```

```
## [1] 0.8533333
```

### Ensemble of models using *selected* variables

**Sex**, Chest Pain Type (**cp**), Excercise Induced Angina (**exang**), ST Depression (**oldpeak**) & number of vessels observed by fluroscopy (**ca**) are the 5 variables that have significant effect on heart disease. The rest of the variables are not included further the following ensembles.
Codes to construct an ensemble of models using **selected** 5 variables is shown for reference only.

```
#############Ensembles with SELECTED predictors#############################
#Spliting training set into two parts based on outcome: 75% and 25%
heart_selected <- heart[,c(2,3,9,10,12,14)]
summary(heart_selected)
y_selected <- heart_selected$target

set.seed(1)
test_index = createDataPartition(y_selected, times = 1, p=0.75,list=FALSE)
train_selected <- heart_selected[test_index,]
test_selected <- heart_selected[-test_index,]

models_selected <-c("glm", "lda",  "naive_bayes",  "svmLinear", "qda",
            "knn", "kknn", "rf", "ranger",  "wsrf", "Rborist",
            "avNNet", "monmlp", "adaboost", "gbm",
            "svmRadial", "svmRadialCost")


fits_selected <- lapply(models_selected, function(model){
  print(model)
  train(target ~ ., method = model, data = train_selected)
})

names(fits_selected) <- models_selected

#all the trained models is in a list now. Next, creating a matrix of predicti
```

```r
ons for the test set
pred_selected <- sapply(fits_selected, function(object)
  predict(object, newdata = test_selected))
dim(pred_selected)

head(pred_selected)

#Accuracy for each model in the test set
#and the mean accuracy across all models can be computed using the following
code:
acc2 <- colMeans(pred_selected == test_selected$target)

acc2_results <- data_frame(method = models_selected, acc_selected = acc2)
acc2_results %>% knitr::kable()

#Result of the mean accuracy across all models.
avg2 <- mean(acc2)
avg2

#Next, build an ensemble prediction by majority vote and compute the accuracy
of the ensemble.
#What is the accuracy of the ensemble
votes2 <- rowMeans(pred_selected == "Heart Disease")
y_hat2 <- ifelse(votes2 > 0.5, "Heart Disease", "Healthy")
votes_avg2 <- mean(y_hat2 == test_selected$target)
votes_avg2

#Which individual methods perform better than the ensemble
ind2 <- acc2 > mean(y_hat2 == test_selected$target)
sum(ind2)
models_selected[ind2]


#using the accuracy estimates obtained from cross validation with the trainin
g data
#finding mean accuracy of the new estimates
acc2_hat <- sapply(fits_selected, function(fit) min(fit$results$Accuracy))
acc2_hat
new_mean2 <- mean(acc2_hat)
new_mean2


#Now let's only consider the methods
#with an estimated accuracy of greater than or equal to the new_mean when con
structing the ensemble
ind2 <- acc2_hat >= new_mean2
sum(ind2)
models_selected[ind2]
```

```r
votes2 <- rowMeans(pred_selected[,ind2] == "Heart Disease")
y_hat2 <- ifelse(votes2 >=0.5, "Heart Disease", "Healthy")
new_votes_avg2 <- mean(y_hat2 == test_selected$target)
new_votes_avg2
```

## RESULTS

Results table includes:
*Ensembling models using **all** variables*
1. **acc** - accuracy for *each* model in the test set
2. the *mean* accuracy across all models (acc)
3. the *majority vote* accuracy of all models (acc)
4. **acc_hat** - accuracy estimates obtained from cross validation with the *training* data
5. the *mean* accuracy of the new estimates (acc_hat)
6. the *majority vote* accuracy of the new estimates (acc_hat)

*Ensembling models using **selected** 5 variables*
7. **acc_selected** - accuracy for *each* model in the selected test set
8. the *mean* accuracy across all models (acc_selected)
9. the *majority vote* accuracy of all models (acc_selected)
10. **hat_selected** - accuracy estimates obtained from cross validation with the *selected training* data
11. the *mean* accuracy of the new estimates (hat_selected)
12. the *majority vote* accuracy of the new estimates (hat_selected)

| method | acc | acc_hat | acc_selected | hat_selected |
|---|---|---|---|---|
| glm | 0.8533333 | 0.7723543 | 0.8666667 | 0.7835416 |
| lda | 0.8533333 | 0.7913839 | 0.8666667 | 0.7813092 |
| naive_bayes | 0.8400000 | 0.7829517 | 0.8000000 | 0.7760658 |
| svmLinear | 0.7866667 | 0.7913491 | 0.8666667 | 0.7722332 |
| qda | 0.8533333 | 0.7518760 | 0.8400000 | 0.7602315 |
| knn | 0.6400000 | 0.6273355 | 0.7733333 | 0.7676430 |
| kknn | 0.8400000 | 0.7651343 | 0.8533333 | 0.7428220 |
| rf | 0.8400000 | 0.7656575 | 0.8133333 | 0.7466681 |
| ranger | 0.8266667 | 0.7687739 | 0.8133333 | 0.7712318 |
| wsrf | 0.7600000 | 0.7493773 | 0.7466667 | 0.7853557 |
| Rborist | 0.6266667 | 0.5236218 | 0.5466667 | 0.5219430 |
| avNNet | 0.8133333 | 0.5699437 | 0.8400000 | 0.7695413 |
| monmlp | 0.8400000 | 0.7676304 | 0.8533333 | 0.7580840 |
| adaboost | 0.7733333 | 0.7697002 | 0.7866667 | 0.7652725 |
| gbm | 0.8133333 | 0.7857458 | 0.7733333 | 0.7973386 |

| | | | | |
|---|---|---|---|---|
| svmRadial | 0.7733333 | 0.7951387 | 0.8400000 | 0.7653703 |
| svmRadialCost | 0.7733333 | 0.7950865 | 0.8400000 | 0.7676060 |
| *Ensemble: Average* | 0.7945098 | 0.7395918 | 0.8070588 | 0.7548387 |
| *Ensemble: Majority Vote* | 0.8400000 | ***0.8533333*** | 0.8666667 | ***0.8666667*** |

## CONCLUSION

In machine learning, the final result of the predictions can be improve by combining the results of different algorithms. Ensembling is used in this project to improve the final accuracy of the model for the **heart** dataset.
The best accuracy of 0.8533333 is achieved with **ensemble majority vote** approach of 17 models using all variables. However, the accuracy is only improved by 1.34% to 0.8666667 with the same approach that uses **selected** 5 best predictors of heart disease *sex*, chest pain (*cp*), excercise induced angina (*exang*), ST depression induced by exercise (*oldpeak*), number of major vessels observed by fluroscopy (*ca*). Perhaps, experimenting with different parameters can yield a higher accuracy. However, it is a continuous process. For now, the **ensemble majority vote** approach with **selected** predictors described above is a winner.