Project Two: Logistic Regression and Random Forests

Scenario

A data analyst is researching risk factors for heart disease at a university hospital. There is access to a large set of historical data that can be used to analyze patterns between different health indicators (e.g. fasting blood sugar, maximum heart rate, etc.) and the presence of heart disease.

Different logistic regression models were created to predict whether or not a person is at risk for heart disease. This could be used to evaluate medical records and look for risks that might not be obvious to human doctors. A classification random forest model was created to predict the risk of heart disease and a regression random forest model to predict the maximum heart rate achieved.

These important variables are used in the modeling:

Variable	What does it represent?
age	The person's age in years
sex	The person's sex (1 = male, 0 = female)
cp	The type of chest pain experienced (0=no pain, 1=typical angina, 2=atypical angina, 3=non-anginal pain)
trestbps	The person's resting blood pressure
chol	The person's cholesterol measurement in mg/dl
fbs	The person's fasting blood sugar is greater than 120 mg/dl (1 = true, 0 = false)
restecg	Resting electrocardiographic measurement (0=normal, 1=having ST-T wave abnormality, 2=showing probable or definite left ventricular hypertrophy by Estes' criteria)
thalach	The person's maximum heart rate achieved
exang	Exercise-induced angina (1=yes, 0=no)
oldpeak	ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot)
slope	The slope of the peak exercise ST segment (1=upsloping, 2=flat, 3=downsloping)
ca	The number of major vessels (0-3)
target	Heart disease (0=no, 1=yes)

R code is used in a Jupyter Notebook environment

Data set preparation:

```
print("This step will first install three R packages. Please wait until the packages are fully installed.")
print("Once the installation is complete, this step will print 'Installation complete!'")

install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")

print("Installation complete!")
```

```
[1] "This step will first install three R packages. Please wait until the packages are fully installed."
[1] "Once the installation is complete, this step will print 'Installation complete!'"
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
also installing the dependency 'pbapply'

Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
also installing the dependency 'plyr'

Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
[1] "Installation complete!"
```

```
heart_data <- read.csv(file="heart_disease.csv", header=TRUE, sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>	<fct>	<fct>
62	1	2	130	231	0	1	146	0	1.8	1	3	3	1
58	0	0	130	197	0	1	131	0	0.6	1	0	2	1
60	0	3	150	240	0	1	171	0	0.9	2	0	2	1
63	1	0	140	187	0	0	144	1	4.0	2	2	3	0
62	1	0	120	267	0	1	99	1	1.8	1	2	3	0
63	0	2	135	252	0	0	172	0	0.0	2	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
48	1	0	124	274	0	0	166	0	0.5	1	0	3	0

```
[1] "Number of variables"
```

14

```
[1] "Number of rows"
```

303

Model # 1 – First Logistic Regression Model

```
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")

heart_data <- read.csv(file="heart_disease.csv", header=TRUE, sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)

# Create the first model
print("Logistic regression model 1")
logit1 <- glm(target ~ age + trestbps + thalach, data = heart_data, family = "binomial")

summary(logit1)

library(ResourceSelection)

print("Hosmer-Lemeshow Goodness of Fit Test")
hl = hoslem.test(logit1$y, fitted(logit1), g=50)
hl

# predict heart disease or no heart disease for the dataset using the model
default_model_data <- heart_data[c('age', 'trestbps', 'thalach')]
pred <- predict(logit1, newdata=default_model_data, type='response')

# if the predicted probability of heart disease is >=0.50 then predict heart disease (default='1'), otherwise predict no heart
# disease (default='0')
depvar_pred = as.factor(ifelse(pred >= 0.5, '1', '0'))
```

```r
# this creates the confusion matrix
conf.matrix <- table(heart_data$target, depvar_pred)[c('0','1'),c('0','1')]
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ": default=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matrix), sep = ": default=")

# print nicely formatted confusion matrix
print("Confusion Matrix")
format(conf.matrix,justify="centre",digit=2)

library(pROC)

labels <- heart_data$target
predictions = logit1$fitted.values

roc <- roc(labels ~ predictions)

# Print Area under the Curve (AUC)
print("Area Under the Curve (AUC)")
round(auc(roc),4)

# Print ROC Curve
print("ROC Curve")

# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc, legacy.axes = TRUE)

# Prediction of heart disease if age=50, resting blood pressure is 122, and max heart rate is 140
print("Prediction: age=50, trestbps=122, thalach=140")
newdata1 <- data.frame(age=50, trestbps=122, thalach=140)
round(predict(logit1, newdata1, type='response'), 4)

# Prediction of heart disease if age=50, resting blood pressure is 130, and max heart rate is 165
print("Prediction: age=50, trestbps=130, thalach=165")
newdata1 <- data.frame(age=50, trestbps=130, thalach=165)
round(predict(logit1, newdata1, type='response'), 4)
```

```
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>	<fct>	<fct>
62	1	2	130	231	0	1	146	0	1.8	1	3	3	1
58	0	0	130	197	0	1	131	0	0.6	1	0	2	1
60	0	3	150	240	0	1	171	0	0.9	2	0	2	1
63	1	0	140	187	0	0	144	1	4.0	2	2	3	0
62	1	0	120	267	0	1	99	1	1.8	1	2	3	0
63	0	2	135	252	0	0	172	0	0.0	2	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
48	1	0	124	274	0	0	166	0	0.5	1	0	3	0

```
[1] "Number of variables"
```

```
14
```

```
[1] "Number of rows"
```

```
303
```

```
[1] "Logistic regression model 1"
```

```
Call:
glm(formula = target ~ age + trestbps + thalach, family = "binomial",
    data = heart_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0257  -1.0069   0.5688   0.9203   2.0476

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.576198   1.633928  -2.189   0.0286 *
age         -0.009424   0.016080  -0.586   0.5578
trestbps    -0.016019   0.007767  -2.063   0.0392 *
thalach      0.042697   0.006950   6.144 8.06e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 417.64  on 302  degrees of freedom
Residual deviance: 353.28  on 299  degrees of freedom
AIC: 361.28

Number of Fisher Scoring iterations: 3
```

```
ResourceSelection 0.3-5          2019-07-22
```

```
[1] "Hosmer-Lemeshow Goodness of Fit Test"
```

```
        Hosmer and Lemeshow goodness of fit (GOF) test

data:  logit1$y, fitted(logit1)
X-squared = 41.978, df = 48, p-value = 0.7168
```

```
[1] "Confusion Matrix"
```

A matrix: 2 × 2 of type chr

	Prediction: default=0	Prediction: default=1
Actual: default=0	83	55
Actual: default=1	38	127

```
Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var

Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
[1] "Area Under the Curve (AUC)"
```

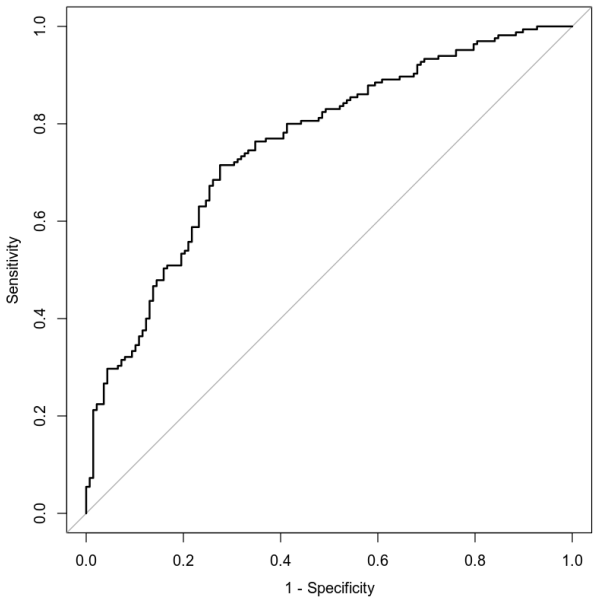
0.7575

```
[1] "ROC Curve"
[1] "Prediction: age=50, trestbps=122, thalach=140"
```

1: 0.4939

```
[1] "Prediction: age=50, trestbps=130, thalach=165"
```

1: 0.714



Model # 2 – Second Logistic Regression Model

```
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")

heart_data <- read.csv(file="heart_disease.csv", header=TRUE, sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)
```

```
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>	<fct>	<fct>
62	1	2	130	231	0	1	146	0	1.8	1	3	3	1
58	0	0	130	197	0	1	131	0	0.6	1	0	2	1
60	0	3	150	240	0	1	171	0	0.9	2	0	2	1
63	1	0	140	187	0	0	144	1	4.0	2	2	3	0
62	1	0	120	267	0	1	99	1	1.8	1	2	3	0
63	0	2	135	252	0	0	172	0	0.0	2	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
48	1	0	124	274	0	0	166	0	0.5	1	0	3	0

```
[1] "Number of variables"
```

14

```
[1] "Number of rows"
```

303

```r
logit2 <- glm(target ~ thalach + age + trestbps + cp + I(age^2) + age:thalach, data = heart_data, family = "binomial")

summary(logit2)

library(ResourceSelection)

print("Hosmer-Lemeshow Goodness of Fit Test")
h2 = hoslem.test(logit2$y, fitted(logit2), g=50)
h2

# predict heart disease or no heart disease for the dataset using the model
default_model_data2 <- heart_data[c('thalach', 'age', 'trestbps', 'cp')]
pred2 <- predict(logit2, newdata=default_model_data2, type='response')

# if the predicted probability of heart disease is >=0.50 then predict heart disease (default='1'), otherwise predict no heart
# disease (default='0')
depvar_pred2 = as.factor(ifelse(pred2 >= 0.5, '1', '0'))

# this creates the confusion matrix
conf.matrix <- table(heart_data$target, depvar_pred2)[c('0','1'),c('0','1')]
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ": default=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matrix), sep = ": default=")

# print nicely formatted confusion matrix
print("Confusion Matrix")
format(conf.matrix,justify="centre",digit=2)

library(pROC)

labels <- heart_data$target
predictions = logit2$fitted.values

roc <- roc(labels ~ predictions)

# Print Area under the Curve (AUC)
print("Area Under the Curve (AUC)")
round(auc(roc),4)

# Print ROC Curve
print("ROC Curve")

# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc, legacy.axes = TRUE)

# Prediction of heart disease if age=50, resting blood pressure=115, max heart rate=133, and cp='0'
print("Prediction: age=50, trestbps=115, thalach=133, cp='0'")
newdata2 <- data.frame(age=50, trestbps=115, thalach=133, cp='0')
round(predict(logit2, newdata2, type='response'), 4)
```

```r
# Prediction of heart disease if age=50, resting blood pressure=125, max heart rate=155, and cp='1'
print("Prediction: age=50, trestbps=125, thalach=155, cp='1'")
newdata2 <- data.frame(age=50, trestbps=125, thalach=155, cp='1')
round(predict(logit2, newdata2, type='response'), 4)
```

```
Call:
glm(formula = target ~ thalach + age + trestbps + cp + I(age^2) +
    age:thalach, family = "binomial", data = heart_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6961  -0.7537   0.2925   0.7123   2.3058

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.556e+01  1.054e+01  -1.476  0.13988
thalach      1.363e-01  5.119e-02   2.663  0.00775 **
age          1.744e-01  2.669e-01   0.653  0.51357
trestbps    -1.958e-02  8.978e-03  -2.181  0.02916 *
cp1          1.913e+00  4.437e-01   4.313 1.61e-05 ***
cp2          2.037e+00  3.473e-01   5.867 4.45e-09 ***
cp3          1.777e+00  5.477e-01   3.245  0.00117 **
I(age^2)     8.424e-04  1.750e-03   0.481  0.63025
thalach:age -1.867e-03  8.909e-04  -2.095  0.03616 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.64  on 302  degrees of freedom
Residual deviance: 293.67  on 294  degrees of freedom
AIC: 311.67

Number of Fisher Scoring iterations: 5
```

```
[1] "Hosmer-Lemeshow Goodness of Fit Test"
```

```
        Hosmer and Lemeshow goodness of fit (GOF) test

data:  logit2$y, fitted(logit2)
X-squared = 52, df = 48, p-value = 0.3209
```

```
[1] "Confusion Matrix"
```

A matrix: 2 × 2 of type chr

	Prediction: default=0	Prediction: default=1
Actual: default=0	102	36
Actual: default=1	36	129

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
[1] "Area Under the Curve (AUC)"
```

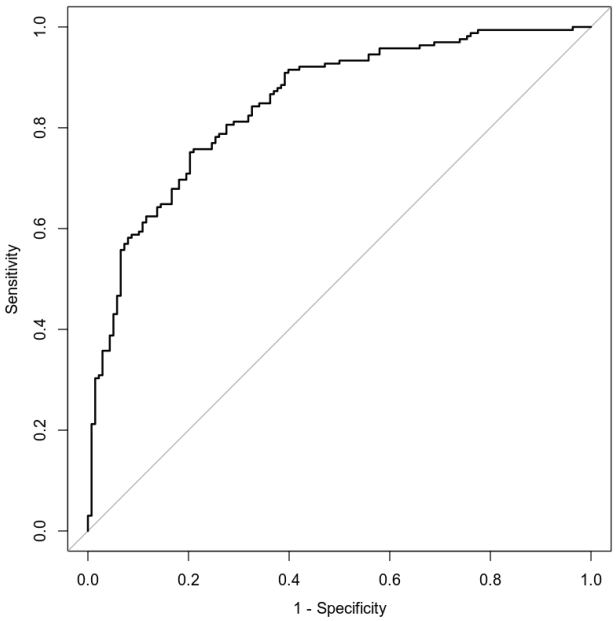
0.8478

```
[1] "ROC Curve"
[1] "Prediction: age=50, trestbps=115, thalach=133, cp='0'"
```

1: 0.2188

```
[1] "Prediction: age=50, trestbps=125, thalach=155, cp='1'"
```

1: 0.8007



Random Forest Classification Model

```
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")

heart_data <- read.csv(file="heart_disease.csv", header=TRUE, sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)
```

```
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>	<fct>	<fct>
62	1	2	130	231	0	1	146	0	1.8	1	3	3	1
58	0	0	130	197	0	1	131	0	0.6	1	0	2	1
60	0	3	150	240	0	1	171	0	0.9	2	0	2	1
63	1	0	140	187	0	0	144	1	4.0	2	2	3	0
62	1	0	120	267	0	1	99	1	1.8	1	2	3	0
63	0	2	135	252	0	0	172	0	0.0	2	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
48	1	0	124	274	0	0	166	0	0.5	1	0	3	0

```
[1] "Number of variables"
```

14

```
[1] "Number of rows"
```

303

```
set.seed(6522048)

# partition the dataset into training and testing data
samp.size = floor(0.85*nrow(heart_data))

# training set
print("Number of rows for the Training set")
train_ind = sample(seq_len(nrow(heart_data)), size = samp.size)
train.data = heart_data[train_ind,]
nrow(train.data)

# testing set
print("Number of rows for the Testing set")
test.data = heart_data[-train_ind,]
nrow(test.data)

library(randomForest)
```

```
[1] "Number of rows for the Training set"
```

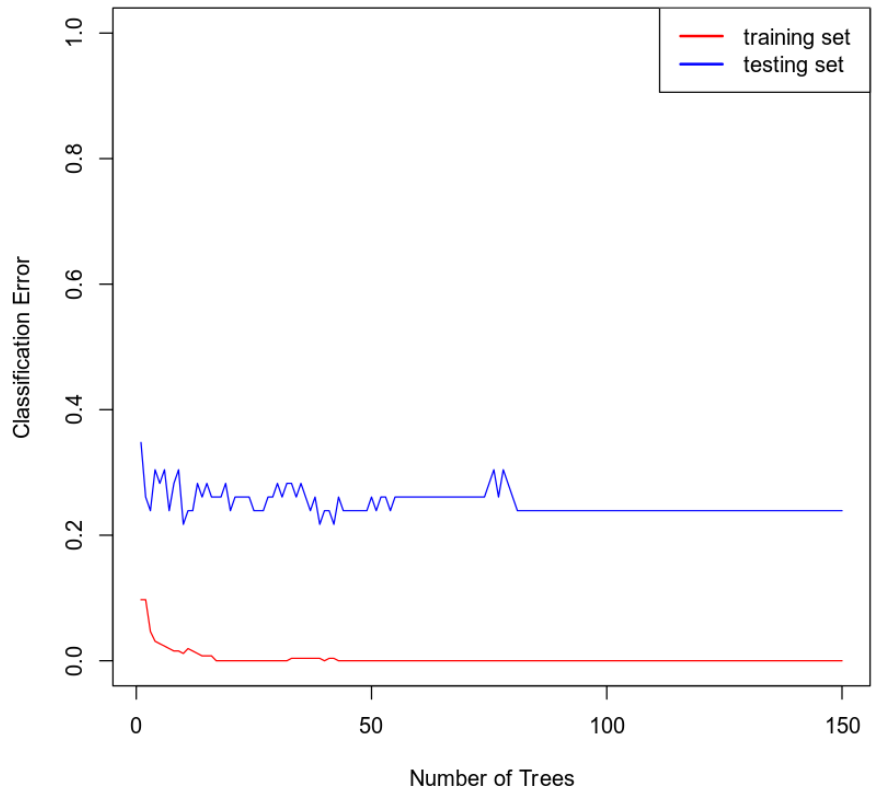
257

```
[1] "Number of rows for the Testing set"
```

46

```
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
```

```r
# Checking
#=================================================================
train = c()
test = c()
trees = c()

for(i in seq(from=1, to=150, by=1)) {
    #print(i)

    trees <- c(trees, i)
    set.seed(6522048)
    model_rf1 <- randomForest(target ~ age+sex+cp+trestbps+chol+restecg+exang+slope+ca, data=train.data, ntree = i)

    train.data.predict <- predict(model_rf1, train.data, type = "class")
    conf.matrix1 <- table(train.data$target, train.data.predict)
    train_error = 1-(sum(diag(conf.matrix1)))/sum(conf.matrix1)
    train <- c(train, train_error)

    test.data.predict <- predict(model_rf1, test.data, type = "class")
    conf.matrix2 <- table(test.data$target, test.data.predict)
    test_error = 1-(sum(diag(conf.matrix2)))/sum(conf.matrix2)
    test <- c(test, test_error)
}

plot(trees, train,type = "l",ylim=c(0,1),col = "red", xlab = "Number of Trees", ylab = "Classification Error")
lines(test, type = "l", col = "blue")
legend('topright',legend = c('training set','testing set'), col = c("red","blue"), lwd = 2 )

set.seed(6522048)

library(randomForest)
model_rf1 <- randomForest(target ~ age+sex+cp+trestbps+chol+restecg+exang+slope+ca, data=train.data, ntree = 20)
```



```r
# Confusion Matrix
print("=============================================================================================================================")
print('Confusion Matrix: TRAINING set based on Random Forest model built using 20 trees')
train.data.predict <- predict(model_rf1, train.data, type = "class")

# construct the confusion matrix
conf.matrix1 <- table(train.data$target, train.data.predict)[,c('0','1')]
rownames(conf.matrix1) <- paste("Actual", rownames(conf.matrix1), sep = ": ")
colnames(conf.matrix1) <- paste("Prediction", colnames(conf.matrix1), sep = ": ")

# print nicely formatted confusion matrix
format(conf.matrix1,justify="centre",digit=2)


print("=============================================================================================================================")
print('Confusion Matrix: TESTING set based on Random Forest model built using 20 trees')
test.data.predict <- predict(model_rf1, test.data, type = "class")

# construct the confusion matrix
conf.matrix2 <- table(test.data$target, test.data.predict)[,c('0','1')]
rownames(conf.matrix2) <- paste("Actual", rownames(conf.matrix2), sep = ": ")
colnames(conf.matrix2) <- paste("Prediction", colnames(conf.matrix2), sep = ": ")

# print nicely formatted confusion matrix
format(conf.matrix2,justify="centre",digit=2)
```

```
[1] "============================================================================================================================="
[1] "Confusion Matrix: TRAINING set based on Random Forest model built using 20 trees"
```

A matrix: 2 × 2 of type chr

	Prediction: 0	Prediction: 1
Actual: 0	120	0
Actual: 1	0	137

```
[1] "============================================================================================================================="
[1] "Confusion Matrix: TESTING set based on Random Forest model built using 20 trees"
```

A matrix: 2 × 2 of type chr

	Prediction: 0	Prediction: 1
Actual: 0	13	5
Actual: 1	6	22

Random Forest Regression Model

```
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")

heart_data <- read.csv(file="heart_disease.csv", header=TRUE, sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)
```

```
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>	<fct>	<fct>
62	1	2	130	231	0	1	146	0	1.8	1	3	3	1
58	0	0	130	197	0	1	131	0	0.6	1	0	2	1
60	0	3	150	240	0	1	171	0	0.9	2	0	2	1
63	1	0	140	187	0	0	144	1	4.0	2	2	3	0
62	1	0	120	267	0	1	99	1	1.8	1	2	3	0
63	0	2	135	252	0	0	172	0	0.0	2	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
48	1	0	124	274	0	0	166	0	0.5	1	0	3	0

```
[1] "Number of variables"
```

14

```
[1] "Number of rows"
```

303

```
set.seed(6522048)

# partition the dataset into training and testing data
samp.size = floor(0.80*nrow(heart_data))

# training set
print("Number of rows for the Training set")
train_ind = sample(seq_len(nrow(heart_data)), size = samp.size)
train.data = heart_data[train_ind,]
nrow(train.data)

# testing set
print("Number of rows for the Testing set")
test.data = heart_data[-train_ind,]
nrow(test.data)

library(randomForest)
```

```
[1] "Number of rows for the Training set"
```

242

```
[1] "Number of rows for the Testing set"
```

61

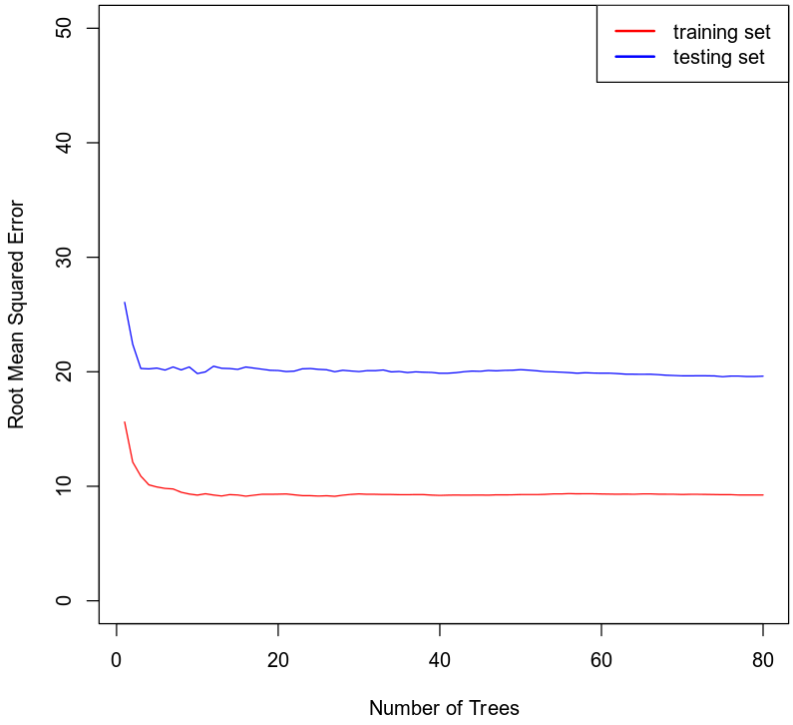
```
train = c()
test = c()
trees = c()

for(i in seq(from=1, to=80, by=1)) {
    set.seed(6522048)
    trees <- c(trees, i)
    model_rf2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol+restecg+exang+slope+ca, data=train.data, ntree = i)

    pred <- predict(model_rf2, newdata=train.data, type='response')
    rmse_train <-  RMSE(pred, train.data$thalach)
    rmse_train
    train <- c(train, rmse_train)

    pred <- predict(model_rf2, newdata=test.data, type='response')
    rmse_test <-  RMSE(pred, test.data$thalach)
    test <- c(test, rmse_test)
}

plot(trees, train,type = "l",col = "red", ylim=c(0,50), xlab = "Number of Trees", ylab = "Root Mean Squared Error")
lines(test, type = "l", col = "blue")
legend('topright',legend = c('training set','testing set'), col = c("red","blue"), lwd = 2 )

set.seed(6522048)
model_rf2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol+restecg+exang+slope+ca, data=train.data, ntree = 80)

# Root Mean Squared Error
RMSE = function(pred, obs) {
    return(sqrt( sum( (pred - obs)^2 )/length(pred) ) )
}



print("=============================================================================================================================")
print('Root Mean Squared Error: TRAINING set based on Random Forest model built using 80 trees')
pred <- predict(model_rf2, newdata=train.data, type='response')
round(RMSE(pred, train.data$thalach),4)


print("=============================================================================================================================")
print('Root Mean Squared Error: TESTING set based on Random Forest model built using 80 trees')
pred <- predict(model_rf2, newdata=test.data, type='response')
round(RMSE(pred, test.data$thalach),4)
```

```
[1] "============================================================================================================================="
[1] "Root Mean Squared Error: TRAINING set based on Random Forest model built using 80 trees"
```

9.2448

```
[1] "============================================================================================================================="
[1] "Root Mean Squared Error: TESTING set based on Random Forest model built using 80 trees"
```

19.6189


training set
testing set
Root Mean Squared Error
0
10
20
30
40
50
0
20
40
60
80
Number of Trees