Lab Report 3

By: Abhishek Shah, Ravi Seth, Tanya Ralliaram

```
# Insert necessary packages
library('glmnet')
library('boot')
library('caret')
library('ISLR')
library('plotly')
library('gridExtra')
library('tree')
```

```
## Warning: package 'tree' was built under R version 4.0.4
```

```
library('rpart')
```

```
## Warning: package 'rpart' was built under R version 4.0.4
```

```
library('rpart.plot')
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.4
```

```
library('rattle')
```

```
## Warning: package 'rattle' was built under R version 4.0.4
```

```
library('MLmetrics')
```

```
## Warning: package 'MLmetrics' was built under R version 4.0.4
```

```
library('e1071')
```

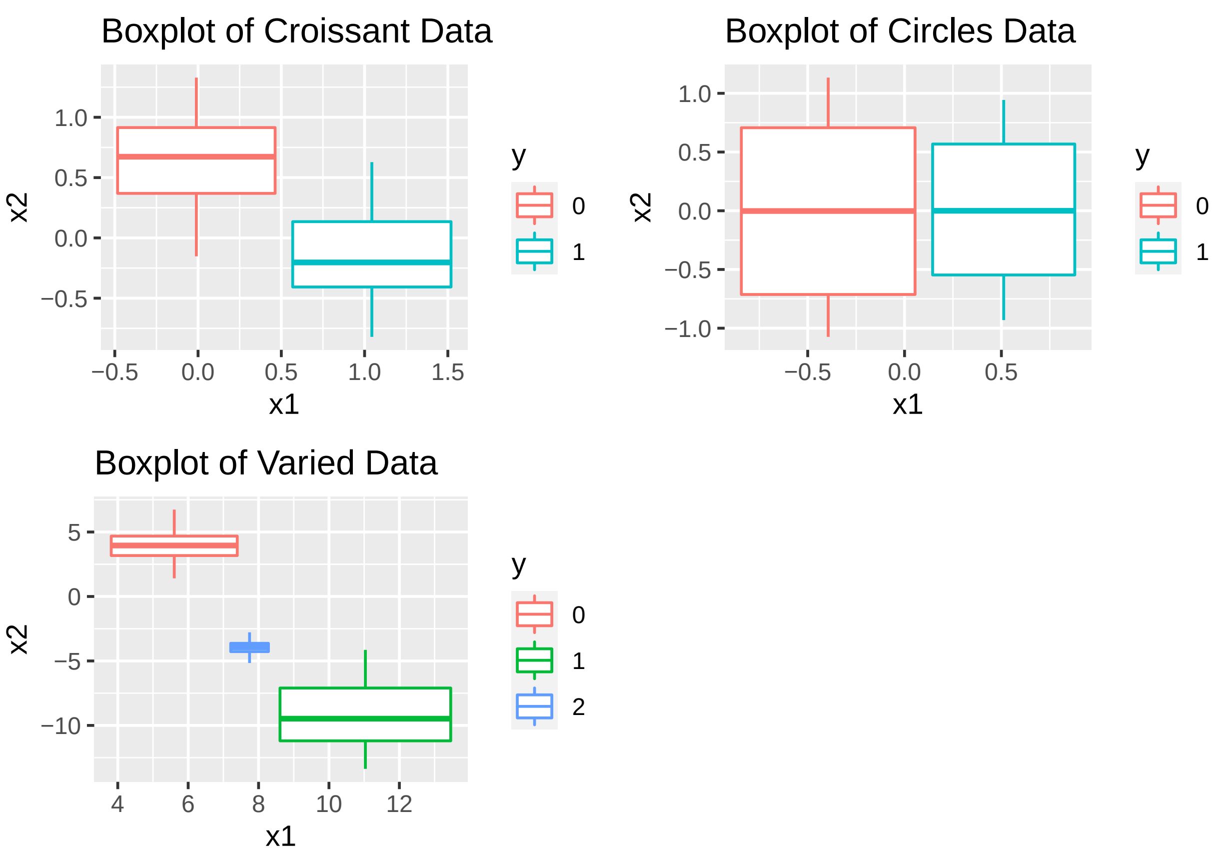
```
## Warning: package 'e1071' was built under R version 4.0.4
```

Question 1: Classification

```r
# Read in data
croissant <- read.csv("data/croissant.csv")[,-1]
circles <- read.csv("data/circles.csv")[,-1]
varied <- read.csv("data/varied.csv")[,-1]
```

1.1: Preprocess and Plot

```r
croissant$y <- as.factor(croissant$y)
circles$y <- as.factor(circles$y)
varied$y <- as.factor(varied$y)

cro <- ggplot(data = croissant) +
  geom_boxplot(aes(x = x1, y=x2, colour=y)) +
  ggtitle("Boxplot of Croissant Data")
cir <- ggplot(data = circles) +
  geom_boxplot(aes(x = x1, y=x2, colour=y)) +
  ggtitle("Boxplot of Circles Data")
var <- ggplot(data = varied) +
  geom_boxplot(aes(x = x1, y=x2, colour=y)) +
  ggtitle("Boxplot of Varied Data")

grid.arrange(cro, cir, var, ncol=2)
```



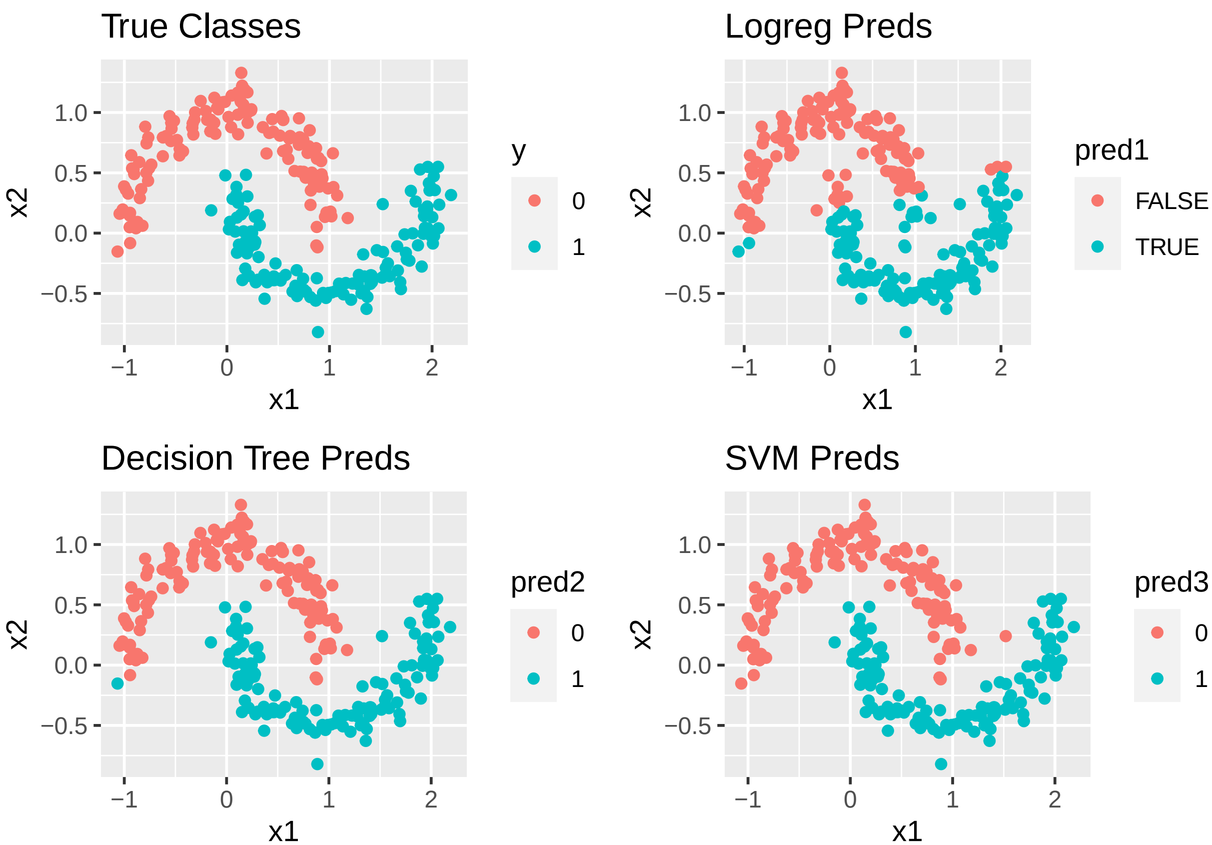
1.2-1.4 for Croissant Data

```
## Question 1.2
set.seed(112)

train_inds <- sample(1:nrow(croissant), floor(nrow(croissant)*0.5))
train <- croissant[ train_inds, ]
test  <- croissant[-train_inds, ]

y.train <- train$y
x.train <- model.matrix(y ~ .,train)[,-1]
x.test  <- model.matrix(y ~ .,test)[,-1]

## Question 1.3

# Logistic Regression
lreg <- glm(y ~ ., data=train, family = "binomial")
pred1 <- predict(lreg, newdata=as.data.frame(x.test), type = "response") > 0.5
lreg_acc <- mean(pred1 == (test$y==1))
lreg_con <- table(predict=pred1,actual=(test$y))

# Decision Tree
dtree <- tree(y~., data=train)
pred2 <- predict(dtree, as.data.frame(x.test), type = "class")
dtree_acc <- Accuracy(pred2,test$y)
dtree_con <- table(predict=pred2,actual=(test$y))

# SVM
svmfit <- svm(y~.,data=train, kernel ="radial", gamma=1,cost=1)
pred3 <- predict(svmfit,as.data.frame(x.test), type="class")
print('We chose radial as the kernel as it best fits the shape of the data
            and thus should lead to a better prediction.')
```

```
## [1] "We chose radial as the kernel as it best fits the shape of the data \n            and thus
```

```
svm_acc <- Accuracy(pred3,test$y)
svm_con <- table(predict=pred3,actual=(test$y))

g1 <- ggplot(test, aes(x1,x2,colour=y)) +
  geom_point() +
  ggtitle("True Classes")
g2 <- ggplot(test, aes(x1,x2,colour=pred1)) +
  geom_point() +
  ggtitle("Logreg Preds")
g3 <- ggplot(test, aes(x1,x2,colour=pred2)) +
  geom_point() +
  ggtitle("Decision Tree Preds")
g4 <- ggplot(test, aes(x1,x2,colour=pred3)) +
  geom_point() +
```

```r
  ggtitle("SVM Preds")

grid.arrange(g1,g2,g3,g4,ncol=2)
```




```r
print('Looking at the four plots, we can see that Logistic Regression has
      the most misclassifications and that Decision Tree performs as well as SVM.')
```

```
## [1] "Looking at the four plots, we can see that Logistic Regression has \n      the most miscla
```

```r
sprintf("Logistic Regression Accuracy: %f", lreg_acc)
```

```
## [1] "Logistic Regression Accuracy: 0.904000"
```

```r
sprintf("Decision Tree Accuracy: %f", dtree_acc)
```

```
## [1] "Decision Tree Accuracy: 0.996000"
```

```r
sprintf("SVM Accuracy: %f", svm_acc)
```

```
## [1] "SVM Accuracy: 0.996000"
```

```r
print("In terms of accuracy, SVM and Decision Tree are the highest.
      Logistic Regression is the lowest out of the three.")
```

```
## [1] "In terms of accuracy, SVM and Decision Tree are the highest. \n      Logistic Regression i
```

```r
lreg_con
```

```
##        actual
## predict   0   1
##   FALSE 112  12
##   TRUE   12 114
```

```r
dtree_con
```

```
##        actual
## predict   0   1
##       0 123   0
##       1   1 126
```

```r
svm_con
```

```
##        actual
## predict   0   1
##       0 124   1
##       1   0 125
```

SVM and Decision Tree are the least biased as they both only have one misclassification. SVM has zero False Positive (FP) and one False Negatives (FN). Decision Tree has 1 FP and 0 FN and Logistic Regression has 12 FP and FN.

```r
## Question 1.4 for Croissant Data

# Logistic Regression
print('Logistic Regression')
```

```
## [1] "Logistic Regression"
```

```r
set.seed(112)
lreg.control <- trainControl(method = 'cv', number = 10)
lreg.cv <- train(y ~ .,
               data = train,
               trControl = lreg.control,
               method = "glm",
               family=binomial())

# summary(lreg.cv)
```

```r
lreg.best <- lreg.cv$finalModel
# lreg.best

pred4 <- predict(lreg.cv, test, type = "raw")
lreg_acc <- Accuracy(pred4,test$y)
lreg_con <- table(predict=pred4,actual=(test$y))

# Decision Tree
print('Decision Tree')
```

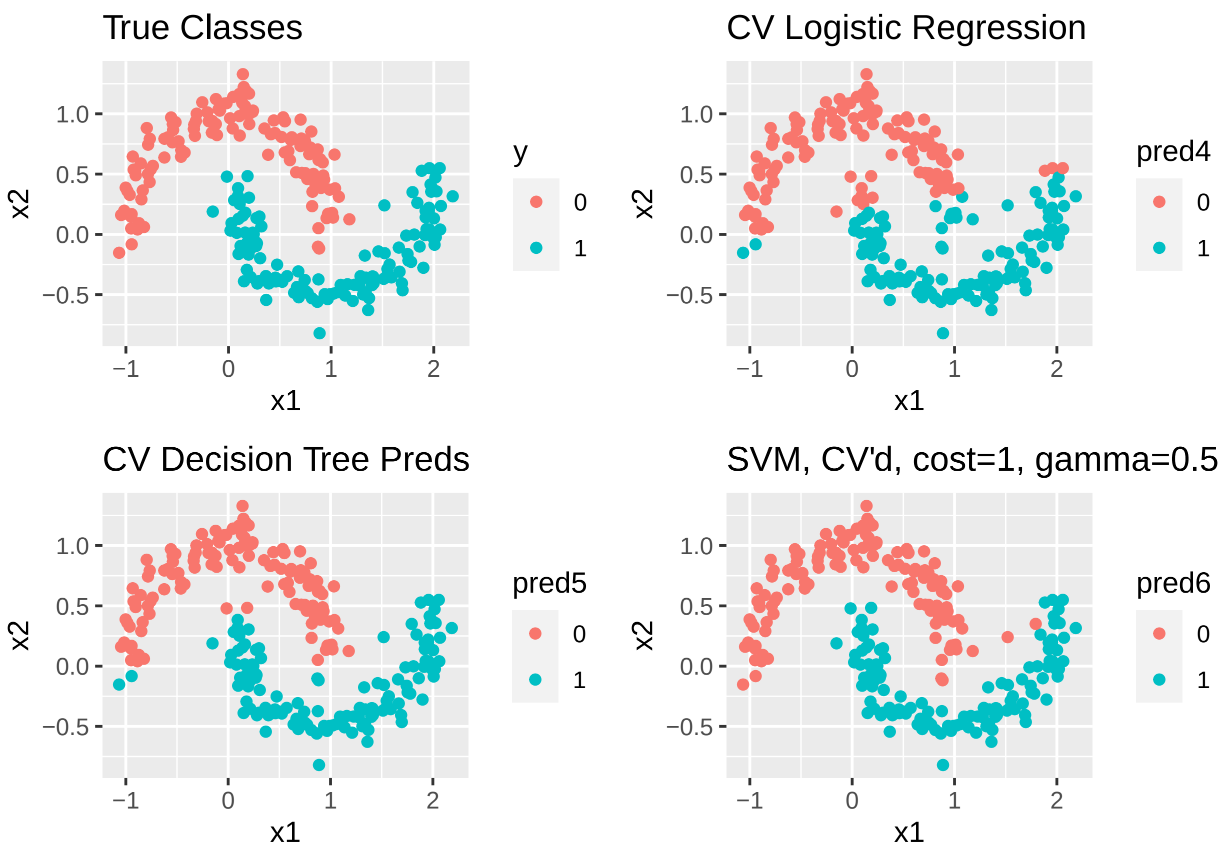
```
## [1] "Decision Tree"
```

```r
set.seed(112)

# perform 10-fold cross validation repeated 3 times
dtree.control = trainControl(method = 'repeatedcv', number = 10, repeats = 3)
dtree.cv <- train(y ~ .,
                  data = train,
                  method = "rpart",
                  trControl = dtree.control,
                  tuneLength = 15)

# summary(dtree.cv)
dtree.best <- dtree.cv$finalModel
# dtree.best

pred5 <- predict(dtree.cv, test, type = "raw")
dtree_acc <- Accuracy(pred5,test$y)
dtree_con <- table(predict=pred5,actual=(test$y))

# SVM
set.seed(112)
svmfit <- svm(y~.,data=train, kernel ="radial", gamma=1,cost=1)
tune.out <- tune(svm, y~., data=train, kernel ="radial",
              ranges =list(cost=c(0.01, 0.05, .1 ,1 ,10 ,100 ,1000),
                           gamma=c(0.5,1,2,3,4)))
pred6 <- predict(tune.out$best.model,test)

print('We chose radial as the kernel as it best fits the shape of the data
            and thus should lead to a better prediction.')
```

```
## [1] "We chose radial as the kernel as it best fits the shape of the data \n            and thus
```

```r
svm_acc <- Accuracy(pred6,test$y)
svm_con <- table(predict=pred6,actual=(test$y))

# summary(tune.out)
```

```r
g1 <- ggplot(test, aes(x1,x2,colour=y)) +
  geom_point() +
  ggtitle("True Classes")
g2 <- ggplot(test, aes(x1,x2,colour=pred4)) +
  geom_point() +
  ggtitle("CV Logistic Regression")
g3 <- ggplot(test, aes(x1,x2,colour=pred5)) +
  geom_point() +
  ggtitle("CV Decision Tree Preds")
g4 <- ggplot(test, aes(x1,x2,colour=pred6)) +
  geom_point() +
  ggtitle("SVM, CV'd, cost=1, gamma=0.5")

grid.arrange(g1,g2,g3,g4,ncol=2)
```




```r
sprintf("Logistic Regression Accuracy: %f", lreg_acc)
```

```
## [1] "Logistic Regression Accuracy: 0.904000"
```

```r
sprintf("Decision Tree Accuracy: %f", dtree_acc)
```

```
## [1] "Decision Tree Accuracy: 0.976000"
```

```r
sprintf("SVM Accuracy: %f", svm_acc)
```

```
## [1] "SVM Accuracy: 0.992000"
```

```r
lreg_con
```

```
##        actual
## predict   0   1
##       0 112  12
##       1  12 114
```

```r
dtree_con
```

```
##        actual
## predict   0   1
##       0 120   2
##       1   4 124
```

```r
svm_con
```

```
##        actual
## predict   0   1
##       0 124   2
##       1   0 124
```

When Cross Validation was added, SVM has the highest accuracy, and Logistic regression still has the lowest. The accuracy for SVM and Decision Tree however got a little worse compared to 1.3.

For bias, SVM was the least biased out of the three as it had zero False Positive (FP) and 2 False Negatives (FN). Decision Tree has 4 FP and 2 FN and Logistics Regression has 12 FP and 12 FN.

Overall, the results appear slightly worse after performing CV for SVM and Decision Tree.

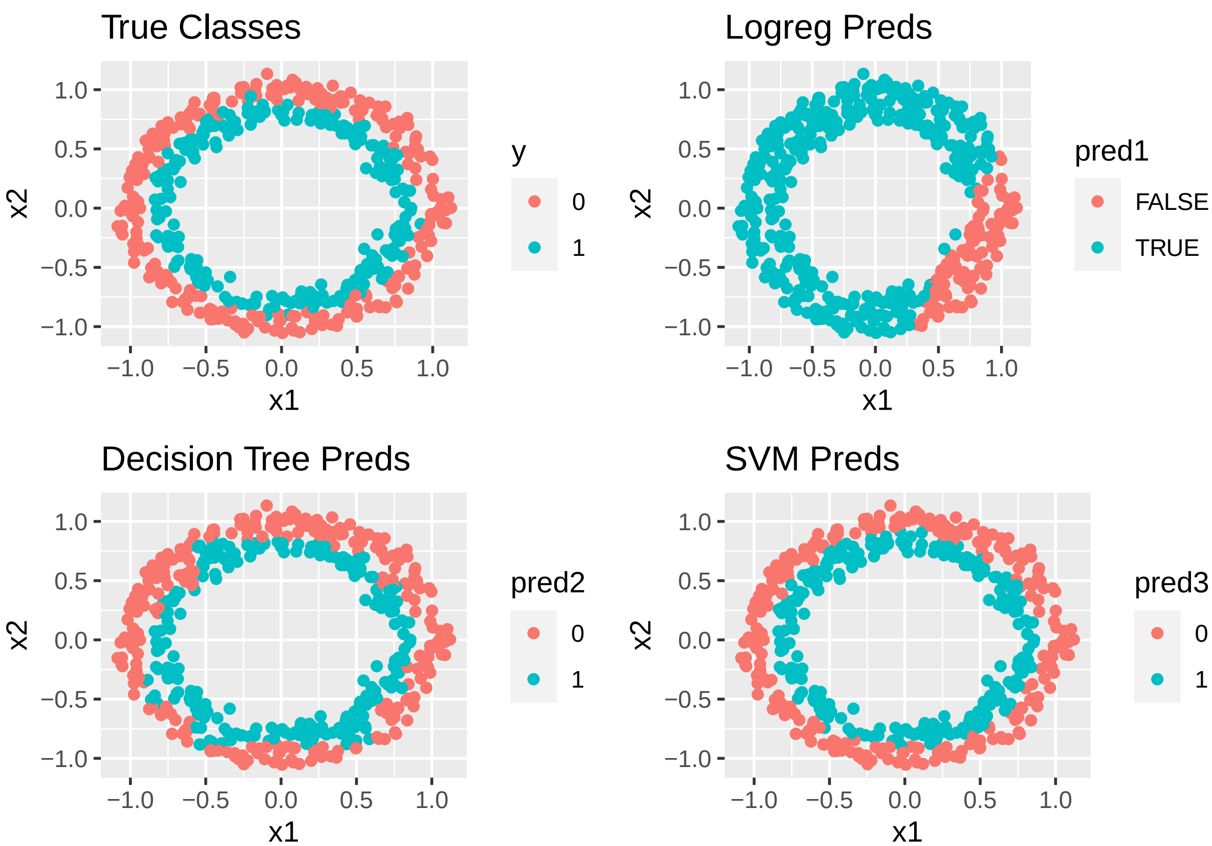
1.2-1.4 for Circle Data

```r
## Question 1.2
set.seed(112)

train_inds <- sample(1:nrow(circles), floor(nrow(circles)*0.5))
train <- circles[ train_inds, ]
test  <- circles[-train_inds, ]

y.train <- train$y
x.train <- model.matrix(y ~ .,train)[,-1]
x.test  <- model.matrix(y ~ .,test)[,-1]

## Question 1.3

# Logistic Regression
lreg <- glm(y ~ ., data=train, family = "binomial")
pred1 <- predict(lreg, newdata=as.data.frame(x.test), type = "response") > 0.5
lreg_acc <- mean(pred1 == (test$y==1))
lreg_con <- table(predict=pred1,actual=(test$y))

# Decision Tree
dtree <- tree(y~., data=train)
pred2 <- predict(dtree, as.data.frame(x.test), type = "class")
dtree_acc <- Accuracy(pred2,test$y)
dtree_con <- table(predict=pred2,actual=(test$y))

# SVM
svmfit <- svm(y~.,data=train, kernel ="radial", gamma=1,cost=1)
pred3 <- predict(svmfit,as.data.frame(x.test), type="class")
print('We chose radial as the kernel as it best fits the shape of the data
            and thus should lead to a better prediction.')
```

```
## [1] "We chose radial as the kernel as it best fits the shape of the data \n            and thus
```

```r
svm_acc <- Accuracy(pred3,test$y)
svm_con <- table(predict=pred3,actual=(test$y))

g1 <- ggplot(test, aes(x1,x2,colour=y)) +
  geom_point() +
  ggtitle("True Classes")
g2 <- ggplot(test, aes(x1,x2,colour=pred1)) +
  geom_point() +
  ggtitle("Logreg Preds")
g3 <- ggplot(test, aes(x1,x2,colour=pred2)) +
  geom_point() +
  ggtitle("Decision Tree Preds")
g4 <- ggplot(test, aes(x1,x2,colour=pred3)) +
  geom_point() +
```

```r
  ggtitle("SVM Preds")

grid.arrange(g1,g2,g3,g4,ncol=2)
```




```r
print('Looking at the four plots, we can see that Logisitc Regression has
      a lot of misclassifications and that Decision Tree performs well, but has noticeable
      misflassications. SVM seems to perform the best.')
```

```
## [1] "Looking at the four plots, we can see that Logisitc Regression has \n      a lot of miscla
```

```r
sprintf("Logisitc Regression Accuracy: %f", lreg_acc)
```

```
## [1] "Logisitc Regression Accuracy: 0.506000"
```

```r
sprintf("Decision Tree Accuracy: %f", dtree_acc)
```

```
## [1] "Decision Tree Accuracy: 0.896000"
```

```r
sprintf("SVM Accuracy: %f", svm_acc)
```

```
## [1] "SVM Accuracy: 0.972000"
```

```r
print("In terms of accuracy, SVM has the highest and Decision Tree was second highest.
      Logistic Regression has the lowest out of the three.")
```

```
## [1] "In terms of accuracy, SVM has the highest and Decision Tree was second highest. \n      Lo
```

```r
lreg_con
```

```
##        actual
## predict   0   1
##   FALSE  58  45
##   TRUE  202 195
```

```r
dtree_con
```

```
##        actual
## predict   0   1
##       0 236  28
##       1  24 212
```

```r
svm_con
```

```
##        actual
## predict   0   1
##       0 253   7
##       1   7 233
```

SVM was the least biased out of the three as it had 7 False Positives (FP) and 7 False Negatives (FN). Decision Tree has 24 FP and 28 FN and Logistic Regression has 202 FP and 45 FN.

```r
## Question 1.4 for Circles Data

# Logistic Regression
print('Logistic Regression')
```

```
## [1] "Logistic Regression"
```

```r
set.seed(112)
lreg.control <- trainControl(method = 'cv', number = 10)
lreg.cv <- train(y ~ .,
               data = train,
               trControl = lreg.control,
               method = "glm",
               family=binomial())

# summary(lreg.cv)
lreg.best <- lreg.cv$finalModel
```

```r
# lreg.best

pred4 <- predict(lreg.cv, test, type = "raw")
lreg_acc <- Accuracy(pred4,test$y)
lreg_con <- table(predict=pred4,actual=(test$y))

# Decision Tree
print('Decision Tree')
```

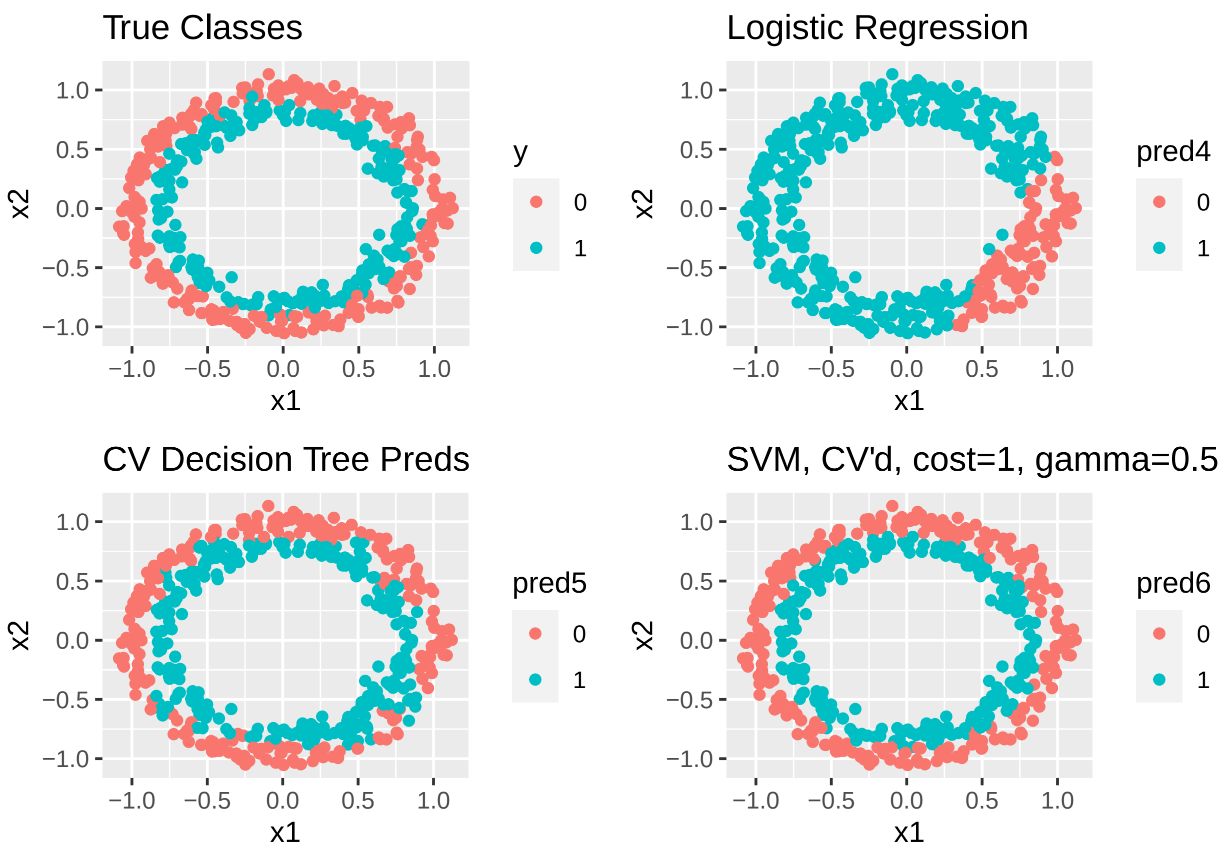
```
## [1] "Decision Tree"
```

```r
set.seed(112)

# perform 10-fold cross validation repeated 3 times
caret.control = trainControl(method = 'repeatedcv', number = 10, repeats = 3)
dtree.cv <- train(y ~ .,
                  data = train,
                  method = "rpart",
                  trControl = caret.control,
                  tuneLength = 15)

# dtree.cv
dtree.best <- dtree.cv$finalModel
# dtree.best

pred5 <- predict(dtree.cv, test, type = "raw")
dtree_acc <- Accuracy(pred5,test$y)
dtree_con <- table(predict=pred5,actual=(test$y))


# SVM
set.seed(112)
svmfit <- svm(y~.,data=train, kernel ="radial", gamma=1,cost=1)
tune.out <- tune(svm, y~., data=train, kernel ="radial",
              ranges =list(cost=c(0.01, 0.05, .1 ,1 ,10 ,100 ,1000),
                           gamma=c(0.5,1,2,3,4)))
pred6 <- predict(tune.out$best.model,test)

print('We chose radial as the kernel as it best fits the shape of the data
            and thus should lead to a better prediction.')
```

```
## [1] "We chose radial as the kernel as it best fits the shape of the data \n            and thus
```

```r
svm_acc <- Accuracy(pred6,test$y)
svm_con <- table(predict=pred6,actual=(test$y))

# summary(tune.out)
```

```r
g1 <- ggplot(test, aes(x1,x2,colour=y)) +
  geom_point() +
  ggtitle("True Classes")
g2 <- ggplot(test, aes(x1,x2,colour=pred4)) +
  geom_point() +
  ggtitle("Logistic Regression")
g3 <- ggplot(test, aes(x1,x2,colour=pred5)) +
  geom_point() +
  ggtitle("CV Decision Tree Preds")
g4 <- ggplot(test, aes(x1,x2,colour=pred6)) +
  geom_point() +
  ggtitle("SVM, CV'd, cost=1, gamma=0.5")

grid.arrange(g1,g2,g3,g4,ncol=2)
```




```r
sprintf("Logistic Regression Accuracy: %f", lreg_acc)
```

```
## [1] "Logistic Regression Accuracy: 0.506000"
```

```r
sprintf("Decision Tree Accuracy: %f", dtree_acc)
```

```
## [1] "Decision Tree Accuracy: 0.906000"
```

```r
sprintf("SVM Accuracy: %f", svm_acc)
```

```
## [1] "SVM Accuracy: 0.974000"
```

```r
lreg_con
```

```
##        actual
## predict   0   1
##       0  58  45
##       1 202 195
```

```r
dtree_con
```

```
##        actual
## predict   0   1
##       0 227  14
##       1  33 226
```

```r
svm_con
```

```
##        actual
## predict   0   1
##       0 253   6
##       1   7 234
```

When Cross Validation was added, the accuracy for SVM and Decision Tree slightly improved. SVM has the highest accuracy, and Logistic regression still has the lowest at only 50.6%.

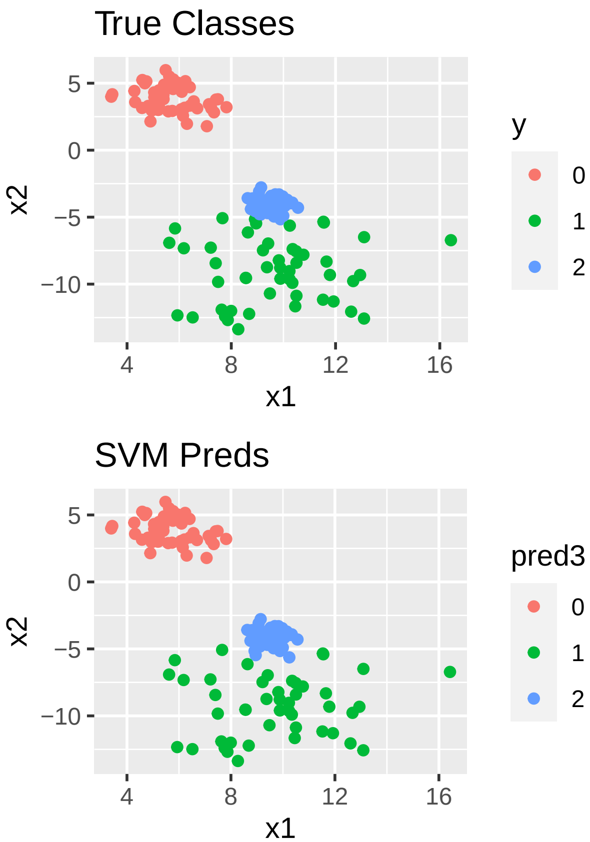
For bias, SVM was the least biased out of the three with 7 False Positive (FP) and 6 False Negatives (FN). Decision Tree has 33 FP and 14 FN and Logistic Regression has 202 FP and 45 FN, indicating that it was overall more likely to predict 1 instead of 0.

1.2-1.4 for Varied Data

```r
## Question 1.2
set.seed(112)

train_inds <- sample(1:nrow(varied), floor(nrow(varied)*0.5))
train <- varied[ train_inds, ]
test  <- varied[-train_inds, ]

y.train <- train$y
x.train <- model.matrix(y ~ .,train)[,-1]
x.test  <- model.matrix(y ~ .,test)[,-1]

## Question 1.3

# Decision Tree
dtree <- tree(y~., data=train)
pred2 <- predict(dtree, as.data.frame(x.test), type = "class")
dtree_acc <- Accuracy(pred2,test$y)
dtree_con <- table(predict=pred2,actual=(test$y))

# SVM
svmfit <- svm(y~.,data=train, kernel ="radial", gamma=1,cost=1)
pred3 <- predict(svmfit,as.data.frame(x.test), type="class")
print('We chose radial as the kernel as it best fits the shape of the data
            and thus should lead to a better prediction.')
```

```
## [1] "We chose radial as the kernel as it best fits the shape of the data \n            and thus
```

```r
svm_acc <- Accuracy(pred3,test$y)
svm_con <- table(predict=pred3,actual=(test$y))

g1 <- ggplot(test, aes(x1,x2,colour=y)) +
  geom_point() +
  ggtitle("True Classes")
g3 <- ggplot(test, aes(x1,x2,colour=pred2)) +
  geom_point() +
  ggtitle("Decision Tree Preds")
g4 <- ggplot(test, aes(x1,x2,colour=pred3)) +
  geom_point() +
  ggtitle("SVM Preds")

grid.arrange(g1,g3,g4,ncol=2)
```




```r
print('Looking at the three plots, Decision Tree and SVM seem to perform equally well.')
```

```
## [1] "Looking at the three plots, Decision Tree and SVM seem to perform equally well."
```

```r
sprintf("Decision Tree Accuracy: %f", dtree_acc)
```

```
## [1] "Decision Tree Accuracy: 0.986667"
```

```r
sprintf("SVM Accuracy: %f", svm_acc)
```

```
## [1] "SVM Accuracy: 0.980000"
```

```r
print("In terms of accuracy, Decision Tree performed a little bit better than SVM.")
```

```
## [1] "In terms of accuracy, Decision Tree performed a little bit better than SVM."
```

```r
dtree_con
```

```
##        actual
## predict  0  1  2
##       0 49  0  0
##       1  0 49  2
##       2  0  0 50
```

```
svm_con
```

```
##        actual
## predict  0  1  2
##       0 49  0  0
##       1  0 46  0
##       2  0  3 52
```

Decision Tree was the least biased has it had two false negatives for Class 2. SVM had 3 false positives for Class 1.

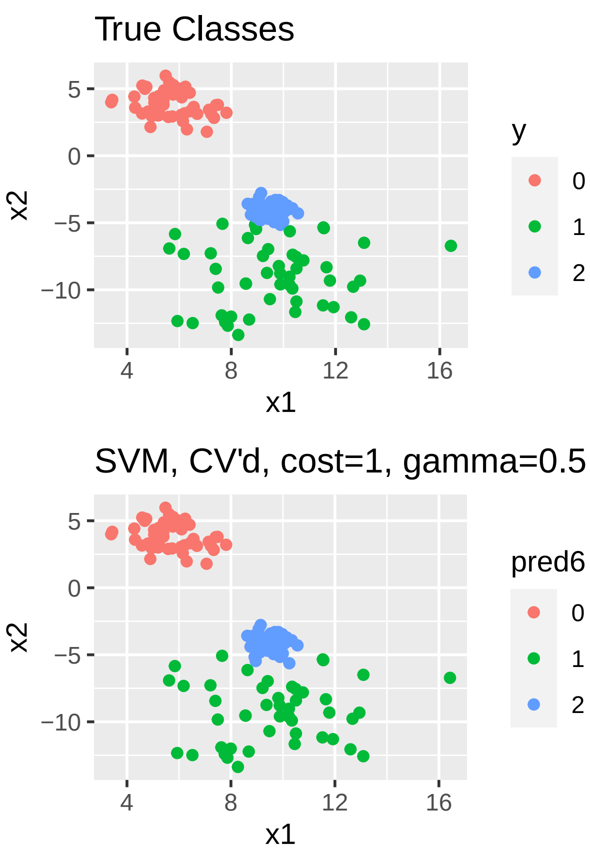
```
## Question 1.4 for Varied Data

# Decision Tree
print('Decision Tree')
```

```
## [1] "Decision Tree"
```

```
set.seed(112)

# perform 10-fold cross validation repeated 3 times
caret.control = trainControl(method = 'repeatedcv', number = 10, repeats = 3)
dtree.cv <- train(y ~ .,
                  data = train,
                  method = "rpart",
                  trControl = caret.control,
                  tuneLength = 15)

# dtree.cv
dtree.best <- dtree.cv$finalModel
# dtree.best

pred5 <- predict(dtree.cv, test, type = "raw")
dtree_acc <- Accuracy(pred5,test$y)
dtree_con <- table(predict=pred5,actual=(test$y))


# SVM
set.seed(112)
svmfit <- svm(y~.,data=train, kernel ="radial", gamma=1,cost=1)
tune.out <- tune(svm, y~., data=train, kernel ="radial",
              ranges =list(cost=c(0.01, 0.05, .1 ,1 ,10 ,100 ,1000),
                           gamma=c(0.5,1,2,3,4)))
pred6 <- predict(tune.out$best.model,test)

print('We chose radial as the kernel as it best fits the shape of the data
            and thus should lead to a better prediction.')
```

```
## [1] "We chose radial as the kernel as it best fits the shape of the data \n            and thus
```

```r
svm_acc <- Accuracy(pred6,test$y)
svm_con <- table(predict=pred6,actual=(test$y))

# summary(tune.out)


g1 <- ggplot(test, aes(x1,x2,colour=y)) +
  geom_point() +
  ggtitle("True Classes")
g3 <- ggplot(test, aes(x1,x2,colour=pred5)) +
  geom_point() +
  ggtitle("CV Decision Tree Preds")
g4 <- ggplot(test, aes(x1,x2,colour=pred6)) +
  geom_point() +
  ggtitle("SVM, CV'd, cost=1, gamma=0.5")

grid.arrange(g1,g3,g4,ncol=2)
```




```r
sprintf("Decision Tree Accuracy: %f", dtree_acc)
```

```
## [1] "Decision Tree Accuracy: 0.986667"
```

```r
sprintf("SVM Accuracy: %f", svm_acc)
```

```
## [1] "SVM Accuracy: 0.980000"
```

```r
dtree_con
```

```
##        actual
## predict  0  1  2
##       0 49  0  0
##       1  0 49  2
##       2  0  0 50
```

```r
svm_con
```

```
##        actual
## predict  0  1  2
##       0 49  0  0
##       1  0 46  0
##       2  0  3 52
```

When Cross Validation was added, both models have high accuracy, with decision tree slightly higher than SVM. The accuracy is identical to before CV was performed.

The bias results are also identical to before CV was performed.

Question 2: Tree-based methods

2.1. Preprocess

```r
# 1
library("ISLR")
completeRows <- complete.cases(Hitters)
hitters <- Hitters[completeRows,]
hitters$Salary <- log(hitters$Salary) # Q2 (Converted to log before dataset is split)

Heart <- read.csv("data/Heart.csv")[-1] # Q3 (removed row identifier)
completeHeartRows <- complete.cases(Heart)
heart <- Heart[completeHeartRows, ]
heart$AHD <- as.factor(heart$AHD)

set.seed(112)
train_inds <- sample(1:nrow(hitters), floor(nrow(hitters)*0.7))
train.hitters <- hitters[ train_inds, ]
test.hitters  <- hitters[-train_inds, ]

train_inds <- sample(1:nrow(heart), floor(nrow(heart)*0.7))
train.heart <- heart[ train_inds, ]
test.heart  <- heart[-train_inds, ]

head(hitters)
```

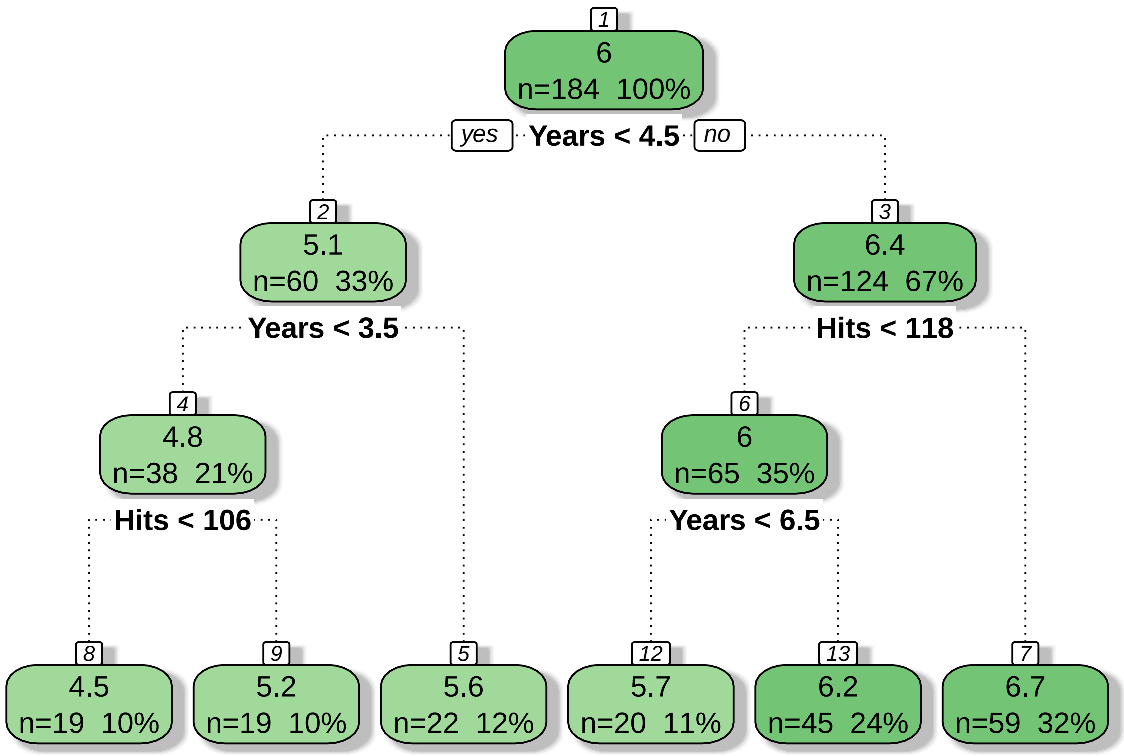
```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     69
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    225
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     12
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133     19
## -Al Newman          185   37     1   23   8    21     2    214    42      1
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48   46     33      N        E     805      40      4
## -Alfredo Griffin    501  336    194      A        W     282     421     25
## -Al Newman           30    9     24      N        E      76     127      7
##                     Salary NewLeague
## -Alan Ashby       6.163315         N
## -Alvin Davis      6.173786         A
## -Andre Dawson     6.214608         N
## -Andres Galarraga 4.516339         N
## -Alfredo Griffin  6.620073         A
## -Al Newman        4.248495         A
```

```r
head(heart)
```

```
##   Age Sex    ChestPain RestBP Chol Fbs RestECG MaxHR ExAng Oldpeak Slope Ca
## 1  63   1      typical    145  233   1       2   150     0     2.3     3  0
## 2  67   1 asymptomatic    160  286   0       2   108     1     1.5     2  3
## 3  67   1 asymptomatic    120  229   0       2   129     1     2.6     2  2
## 4  37   1   nonanginal    130  250   0       0   187     0     3.5     3  0
## 5  41   0   nontypical    130  204   0       2   172     0     1.4     1  0
## 6  56   1   nontypical    120  236   0       0   178     0     0.8     1  0
##         Thal AHD
## 1      fixed  No
## 2     normal Yes
## 3 reversable Yes
## 4     normal  No
## 5     normal  No
## 6     normal  No
```

2.2. Decision Trees for Regression

```r
# 1
set.seed(112)

dtree_hitters <- rpart(Salary ~ Hits + Years, data=train.hitters)

# 2
fancyRpartPlot(dtree_hitters, caption ="")
```




```
# 3
print("Based on the decision tree, the output is the node labelled 7.
      The player's salary should be around 6.7")
```

```
## [1] "Based on the decision tree, the output is the node labelled 7.\n      The player's salary
```

```
# 4
preds2.2 <- predict(dtree_hitters,test.hitters, type="vector")
SSE.tree <- sum((test.hitters$Salary - preds2.2)^2)
sprintf("Regressor Decision Tree SSE: %f", SSE.tree)
```

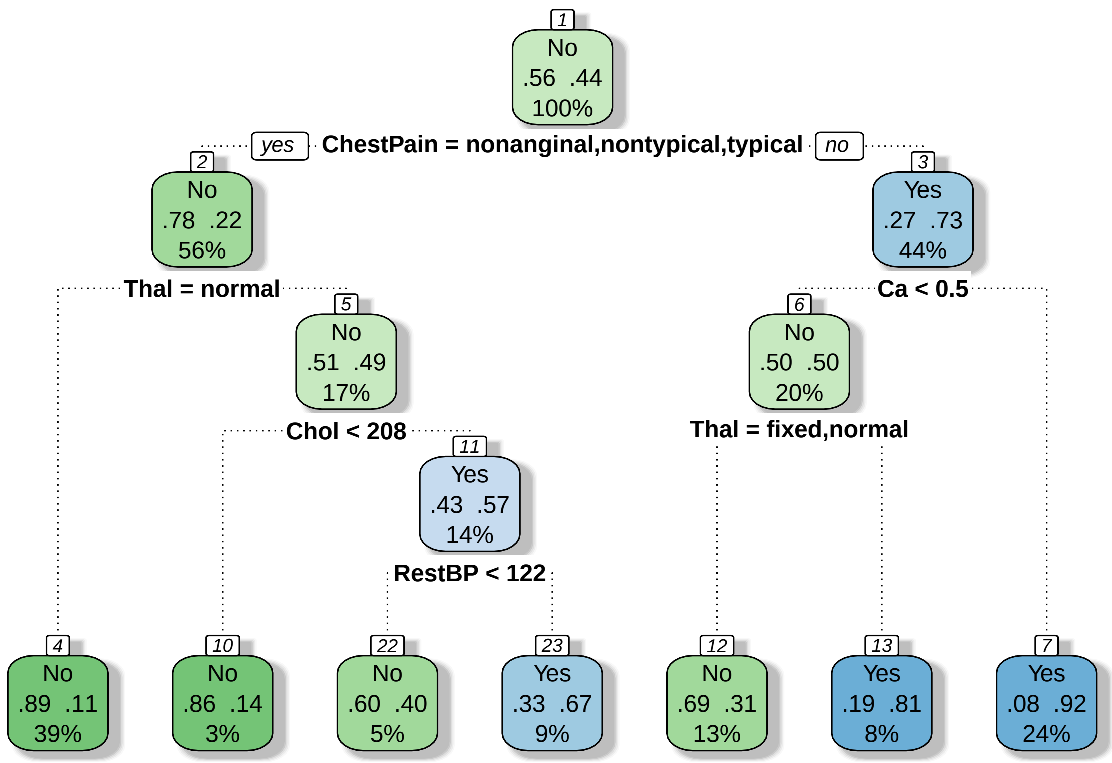
```
## [1] "Regressor Decision Tree SSE: 32.960801"
```

```
preds2.2[0:4]
```

```
##     -Andre Dawson -Andres Galarraga    -Andres Thomas     -Alex Trevino
##          6.733644          4.514404          4.514404          6.177275
```

2.3. Decision Trees for Classification

```r
# 1
set.seed(112)

dtree_heart <- rpart(AHD ~ ., data=train.heart)

# 2
fancyRpartPlot(dtree_heart, caption ="")
```




```r
# 3
preds2.3 <- predict(dtree_heart,test.heart, type="class")
accuracy <- Accuracy(preds2.3,test.heart$AHD)

sprintf("Classification Decision Tree Accuracy %f %%", accuracy*100)
```

```
## [1] "Classification Decision Tree Accuracy 87.777778 %"
```

```r
# 4
conf <- ConfusionMatrix(preds2.3, test.heart$AHD)
conf
```

```
##        y_pred
## y_true No Yes
##     No  40   5
##     Yes  6  39
```

2.4. Bagging: Regression

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.4

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':
##
##     importance

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

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

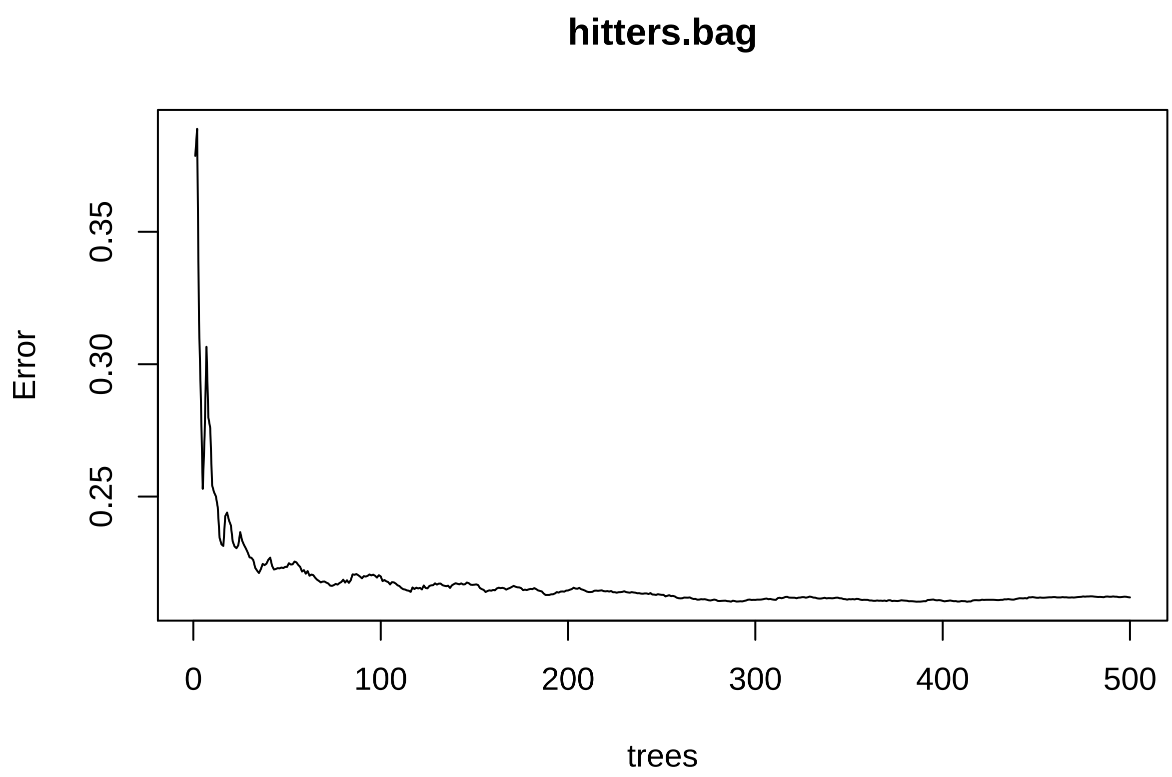
```r
library(apricom)
```

```
## Warning: package 'apricom' was built under R version 4.0.4
```

```r
set.seed(112)

# 1
print("Filtering out the NA values is done in the pre-processing step")
```

```
## [1] "Filtering out the NA values is done in the pre-processing step"
```

```r
# 2
hitters.bag <- randomForest(Salary ~ . , data = train.hitters,mtry = ncol(train.hitters)-1)
plot(hitters.bag)
```




```
# 3
preds.hittersBag <- predict(hitters.bag,test.hitters)
preds.hittersBag[0:4]
```

```
##     -Andre Dawson -Andres Galarraga    -Andres Thomas     -Alex Trevino
##          6.705258          4.633184          4.671069          6.042472
```

```
# 4
sse.bagging <- sum((test.hitters$Salary - preds.hittersBag)^2)
sprintf("Bagging Regression SSE: %f", sse.bagging)
```

```
## [1] "Bagging Regression SSE: 25.480927"
```

```
# 5
print("The SSE from bagging is 25.48093 and is lower than the SSE from
      regression tree which is 32.9608 ")
```

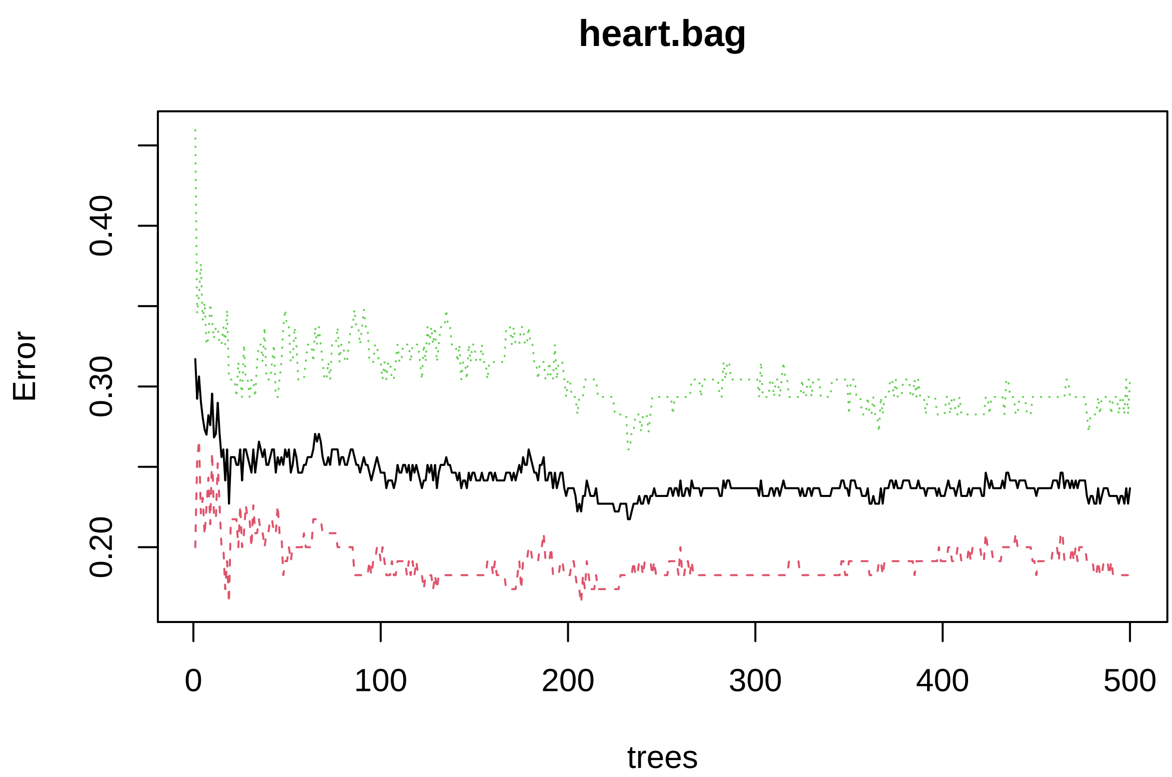
```
## [1] "The SSE from bagging is 25.48093 and is lower than the SSE from \n      regression tree wh
```

2.5. Bagging: Classification

```r
set.seed(112)

# 1
print("Filtering out the NA values is done in the pre-processing step")
```

```
## [1] "Filtering out the NA values is done in the pre-processing step"
```

```r
# 2
heart.bag <- randomForest(AHD ~ . , data = train.heart)
plot(heart.bag)
```




```r
# 3
preds.heartBag <- predict(heart.bag, test.heart, type = "class")
preds.heartBag[0:4]
```

```
##   1   4   6  10
##  No  No  No Yes
## Levels: No Yes
```

```r
# 4
accuracy.bagging <- Accuracy(preds.heartBag,test.heart$AHD)
sprintf("Bagging Classification Accuracy: %f %%", accuracy.bagging*100)
```

```
## [1] "Bagging Classification Accuracy: 88.888889 %"
```

```
ConfusionMatrix(preds.heartBag, test.heart$AHD)
```

```
##        y_pred
## y_true No Yes
##    No  40   5
##    Yes  5  40
```

```
# 5
print("The accuracy from bagging is 88.89% which is higher than the accuracy
      from classification tree which is 87.78%")
```

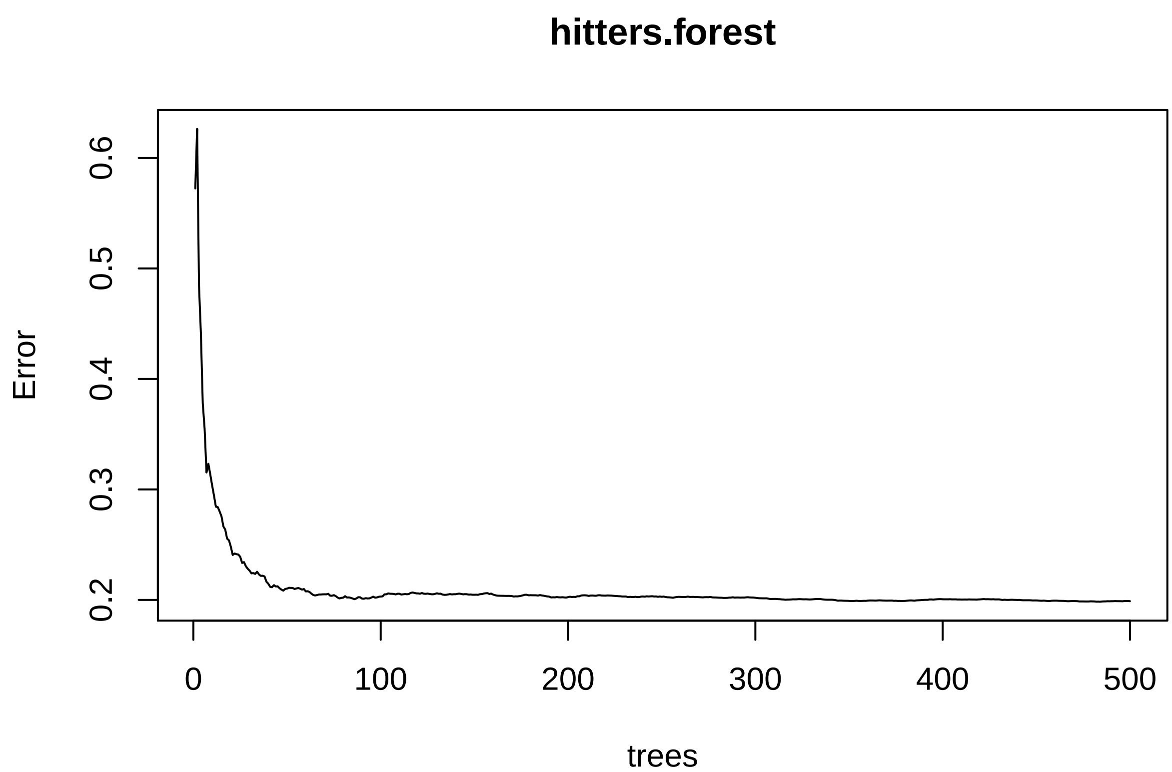
```
## [1] "The accuracy from bagging is 88.89% which is higher than the accuracy \n      from classif
```

2.6. Random Forest: Regression

```
set.seed(21)

# 1
sprintf("Instead of doing na.action, I instead removed the NA values, which is
        done in the pre-processing step")
```

```
## [1] "Instead of doing na.action, I instead removed the NA values, which is \n        done in th
```

```
# 2
m <- ceiling((ncol(train.hitters)-1)/3)
hitters.forest <- randomForest(Salary ~ . , data = train.hitters, mtry = m, importance=T)
plot(hitters.forest)
```




```
# 3
preds.hittersForest <- predict(hitters.forest,test.hitters)
preds.hittersForest[0:4]
```

```
##     -Andre Dawson -Andres Galarraga    -Andres Thomas     -Alex Trevino
##          6.729212          4.645201          4.630330          6.013597
```

```
# 4
sse.forest <- sum((test.hitters$Salary - preds.hittersForest)^2)
sprintf("Forest SSE: %f", sse.forest)
```

```
## [1] "Forest SSE: 23.306591"
```

```
# 5
print("The SSE from the random forest is 23.306591 which is lower than both the
      SSE from bagging (which is 25.48093) and the SSE from regression tree
      (which is 32.9608)")
```

```
## [1] "The SSE from the random forest is 23.306591 which is lower than both the \n      SSE from
```