lab1

Erik Tedhamre

8 December 2018

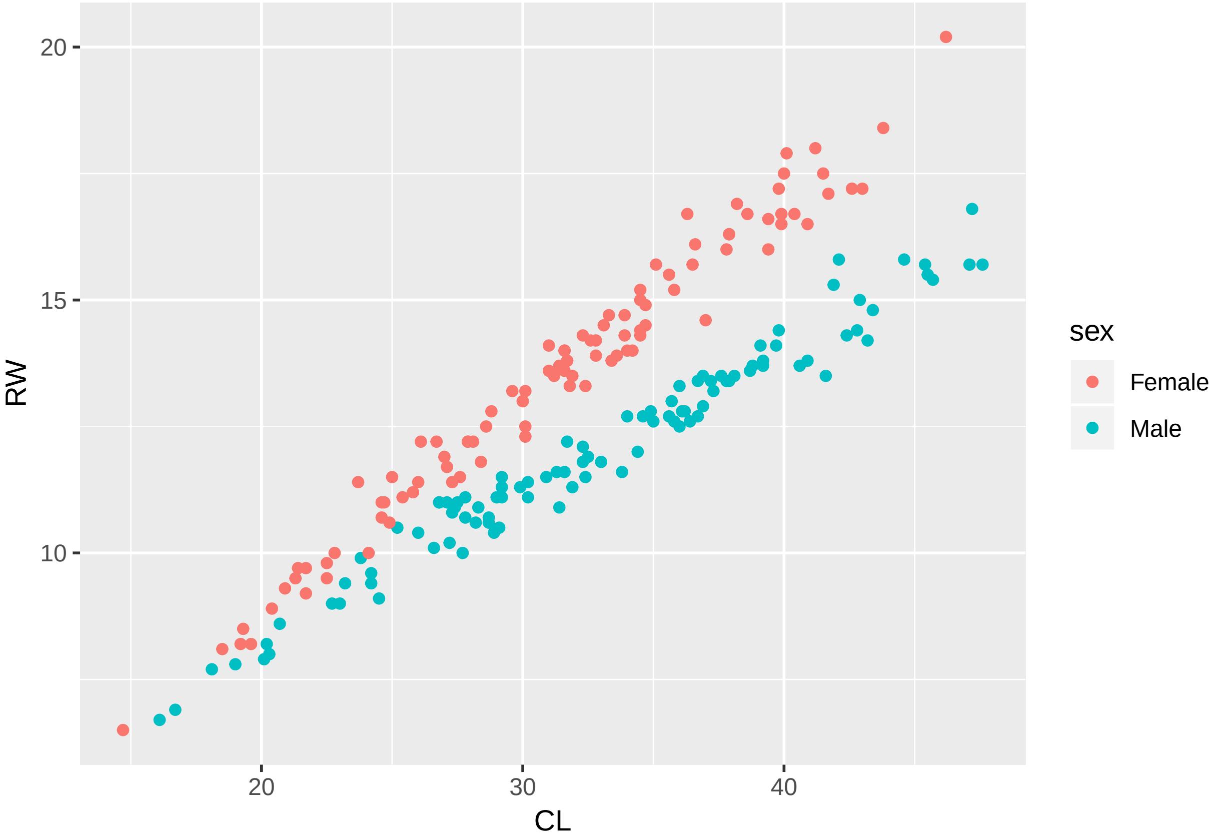
R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Assignment 1.1

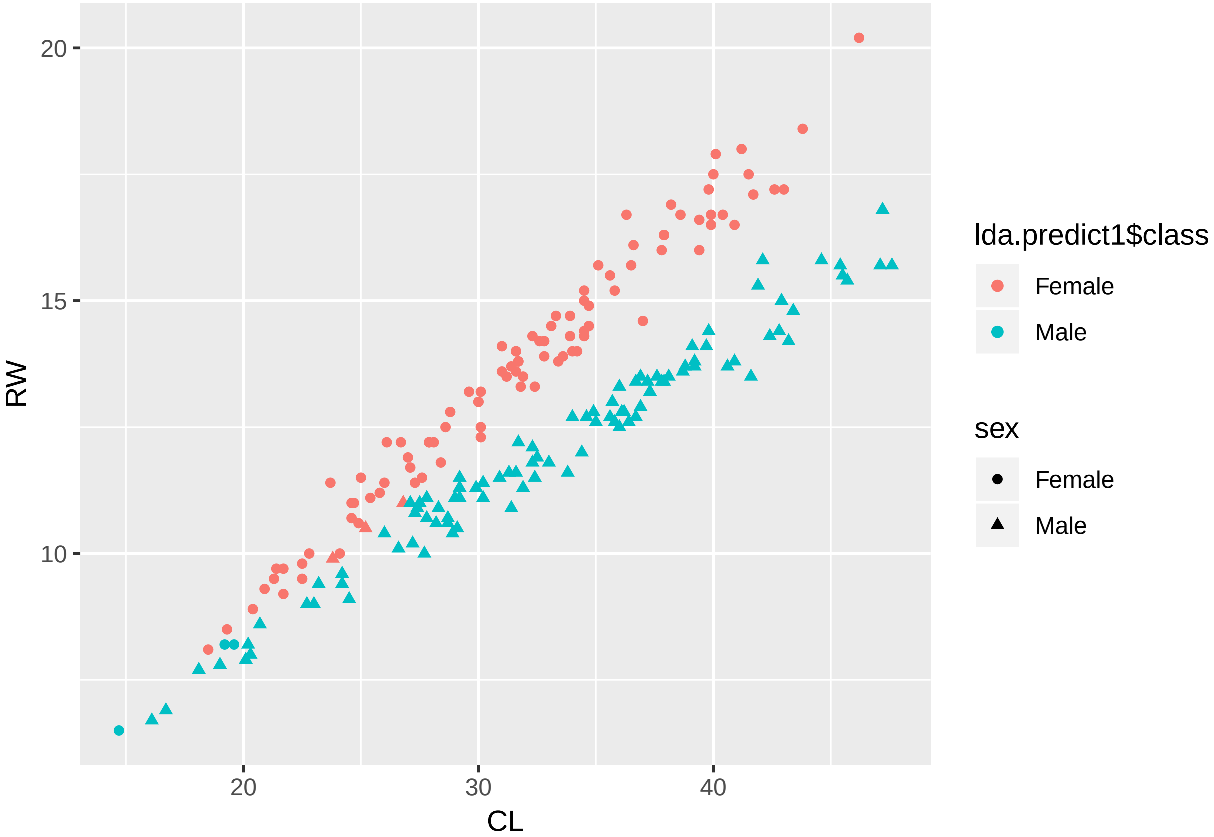
```r
library("ggplot2")
library("MASS")
data <- data.frame(read.csv("australian-crabs.csv"))
ggplot(data = data, mapping = aes(CL, RW, color = sex)) + geom_point()
```



A plot of carapace length versus rear width where the observations are colored by sex. Looking at the graph the data seems reasonably easy to classify by linear discriminant analysis. Because there seems to be a line between the two sexes.

Assignment 1.2

```
lda.model1 <- lda(sex ~ CL + RW, data = data)
lda.predict1 <- predict(lda.model1, data)
ggplot.0.5 <- ggplot(data = data, mapping = aes(CL, RW, color = lda.predict1$class, shape = sex )) + ge
ggplot.0.5
```

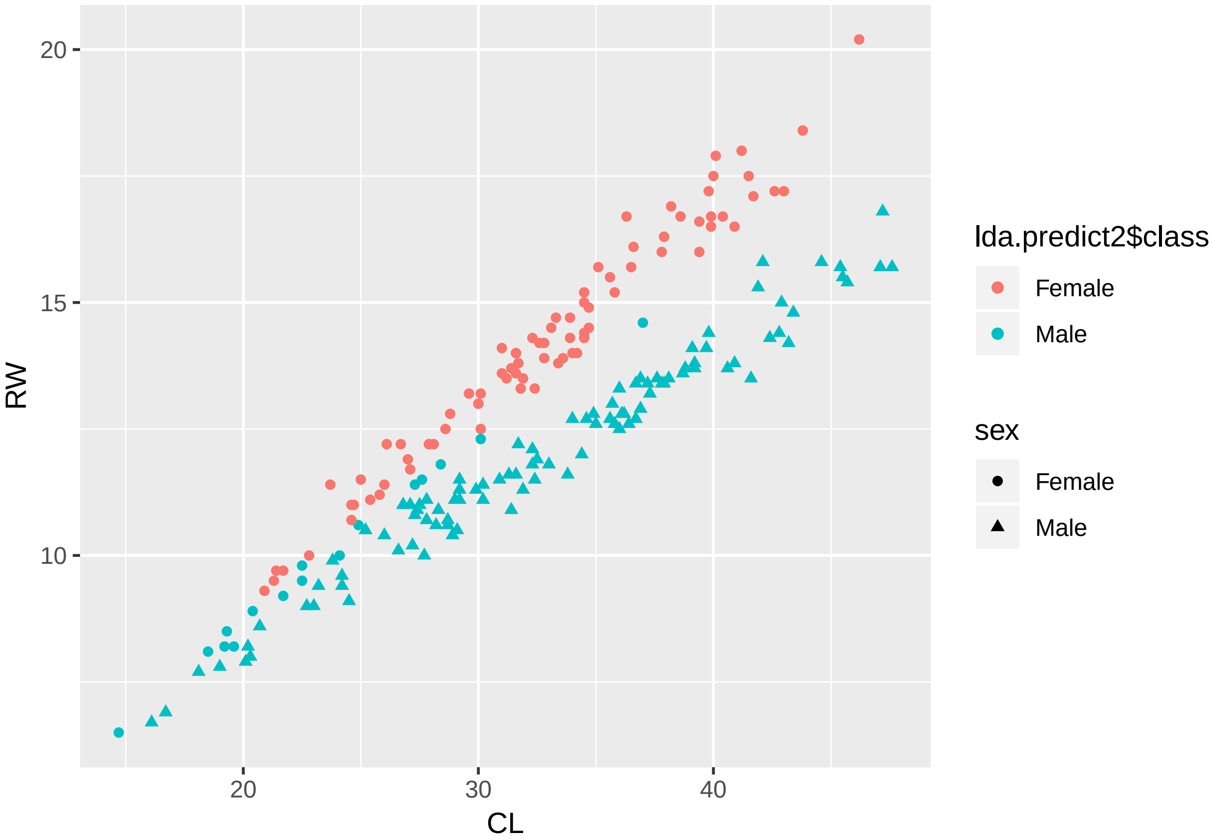


```
mcr.0.5 <- mean(lda.predict1$class != data$sex)
```

The missclassification rate for the linear discriminant analysis is **0.035**. This is pretty reasonable considering we saw on the original graph that there was one area with a bit of an overlap. If this is good enough for actual use is hard to say, it mostly depends on how much we would lose on an incorrect classification.

Assignment 1.3

```
lda.model2 <- lda(sex ~ CL + RW, data = data, prior = c(Female = 0.1, Male = 0.9))
lda.predict2 <- predict(lda.model2, data)
ggplot(data = data, mapping = aes(CL, RW, color = lda.predict2$class, shape = sex )) + geom_point()
```



The number of males increased since we are assuming a wheighted distribution. Especially the areas containing both types of observations are now classified as only males instead of both.

```
mcr.0.9 <- mean(lda.predict2$class != data$sex)
```

The missclassification rate for the weighted linear discriminant analysis is **0.08**. ## Assignment 1.4

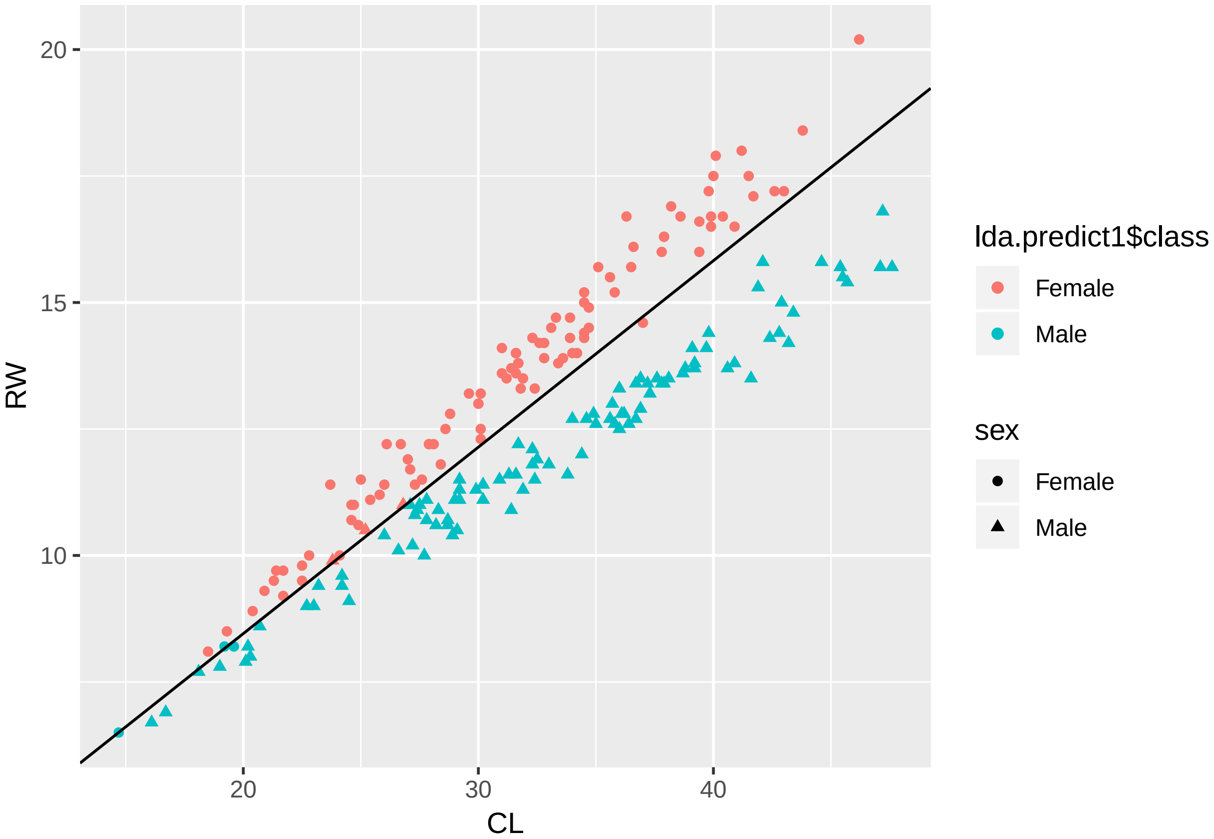
```
glm.model <- glm(as.factor(sex) ~ CL + RW, family = binomial, data = data)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
glm.predict <- predict(glm.model, data, type = 'response')
mcr.glm <- mean(glm.predict != data$sex)
```

The missclassification rate is **0.035** which is the same as the original linear discriminant analysis.

```
glm.predict.0.5 <- ifelse(glm.predict > 0.5, "Male", "Female")
glm.slope <- coef(glm.model)[2]/(-coef(glm.model)[3])
glm.intercept <- coef(glm.model)[1]/(-coef(glm.model)[3])
ggplot.0.5 + geom_abline(slope = glm.slope, intercept = glm.intercept)
```



The decision line is drawn in the graph.

Assignment 2

Splitting the data into partitions

```
library("e1071")
library("MASS")
library("tree")
library("ggplot2")
setwd("~/TDDE01/lab2")
data.credit <- data.frame(read.csv("creditscoring.csv"))
n=dim(data.credit)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.5))
train=data.credit[id,]
id1=setdiff(1:n, id)
set.seed(12345)
id2=sample(id1, floor(n*0.25))
valid=data.credit[id2,]
id3=setdiff(id1,id2)
test=data.credit[id3,]
```

The models used in calculating the following confusion matrices.

```
##       pred.dev.train
##         bad good
##   bad    61   86
##   good  20  333
```

Confusion matrix for deviance on train data.

```
##       pred.dev.test
##        bad good
##   bad   28   48
##   good  19  155
```

Confusion matrix for deviance on test data.

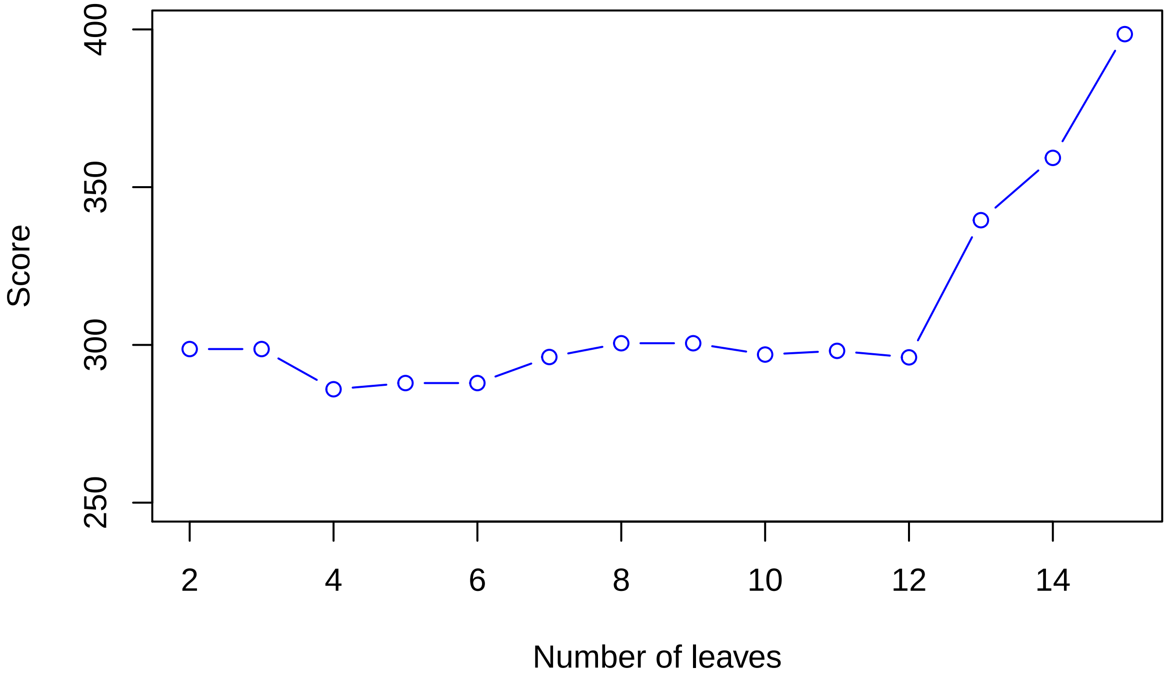
```
##       pred.gini.train
##        bad good
##   bad   66   81
##   good  38  315
```

Confusion matrix for gini on train data.

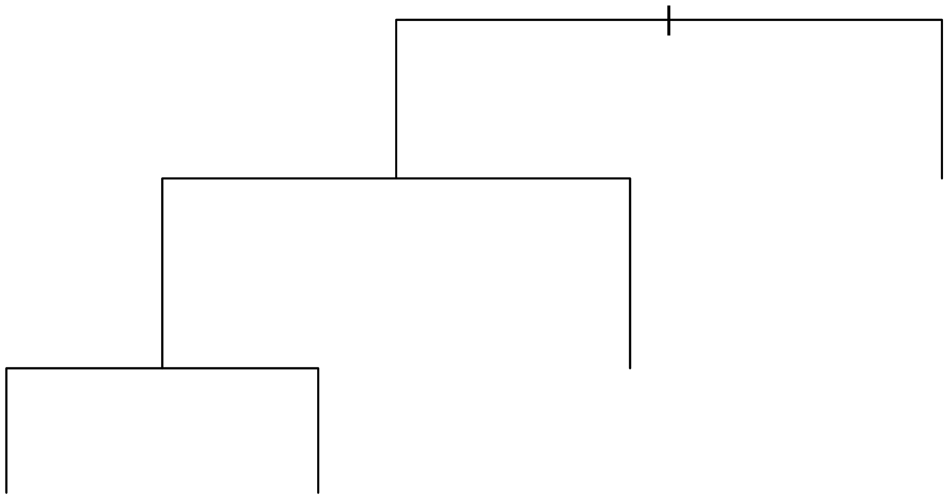
```
##       pred.gini.test
##        bad good
##   bad   18   58
##   good  35  139
```

Confusion matrix for gini on test data. The confusion matrix is the best for deviance compared to gini based on the number of correct predictions for the test data.

Assignment 2.3



Looking at the graph we see a minimum value for 4.



Something about the tree structure. The optimal depth of the tree is three which can be seen in the graph.

```
##       Yfit
##        bad good
##   bad   23   54
##   good  12  161
```

Confusion matrix for the validation data for the tree data

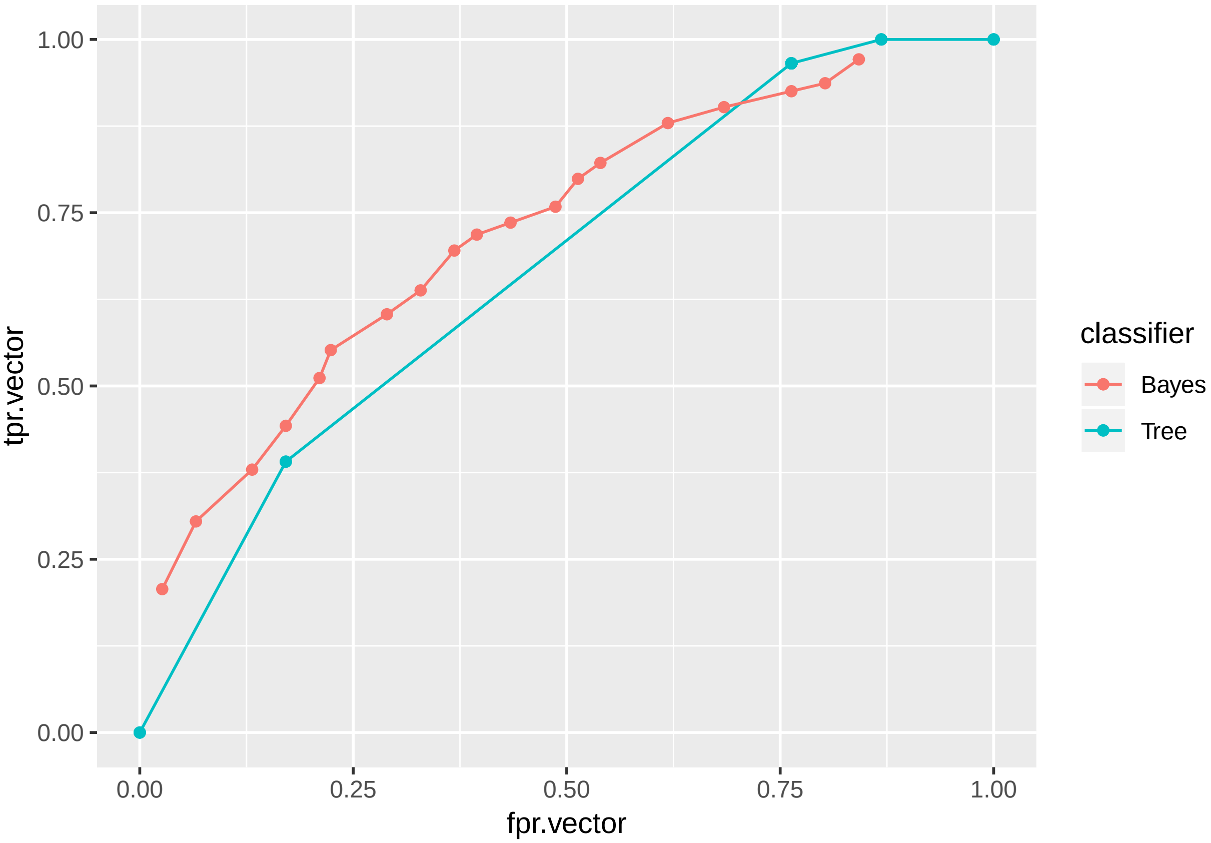
```
##
## Classification tree:
## snip.tree(tree = fit, nodes = c(5L, 3L, 9L))
## Variables actually used in tree construction:
## [1] "savings"  "duration" "history"
## Number of terminal nodes:  4
## Residual mean deviance:  1.117 = 547.5 / 490
## Misclassification error rate: 0.251 = 124 / 494
```

As seen above the variables used in the tree are “savings”, “duration” and “history”.

```
##
## Yfit.bayes.train bad good
##             bad   95   98
##             good  52  255
```

```
##
## Yfit.bayes.test bad good
##            bad   46   49
##            good  30  125
```

The tree prediction is a bit better than the bayesian prediction.


1.00
0.75
0.50
0.25
0.00
tpr.vector
0.00
0.25
0.50
0.75
1.00
fpr.vector
classifier
Bayes
Tree