Stats 415 hw9

Name: Li Hsuan Lin

UM_ID: 49109112

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────── tidyverse 1.2.1 ──
```

```
## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ dplyr   0.8.3
## ✓ tidyr   1.0.0     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0
```

```
## ── Conflicts ────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ISLR)
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
crab = crabs
str(crab)

## 'data.frame':    200 obs. of  8 variables:
##  $ sp   : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ FL   : num  8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
##  $ RW   : num  6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
##  $ CL   : num  16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
##  $ CW   : num  19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
##  $ BD   : num  7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

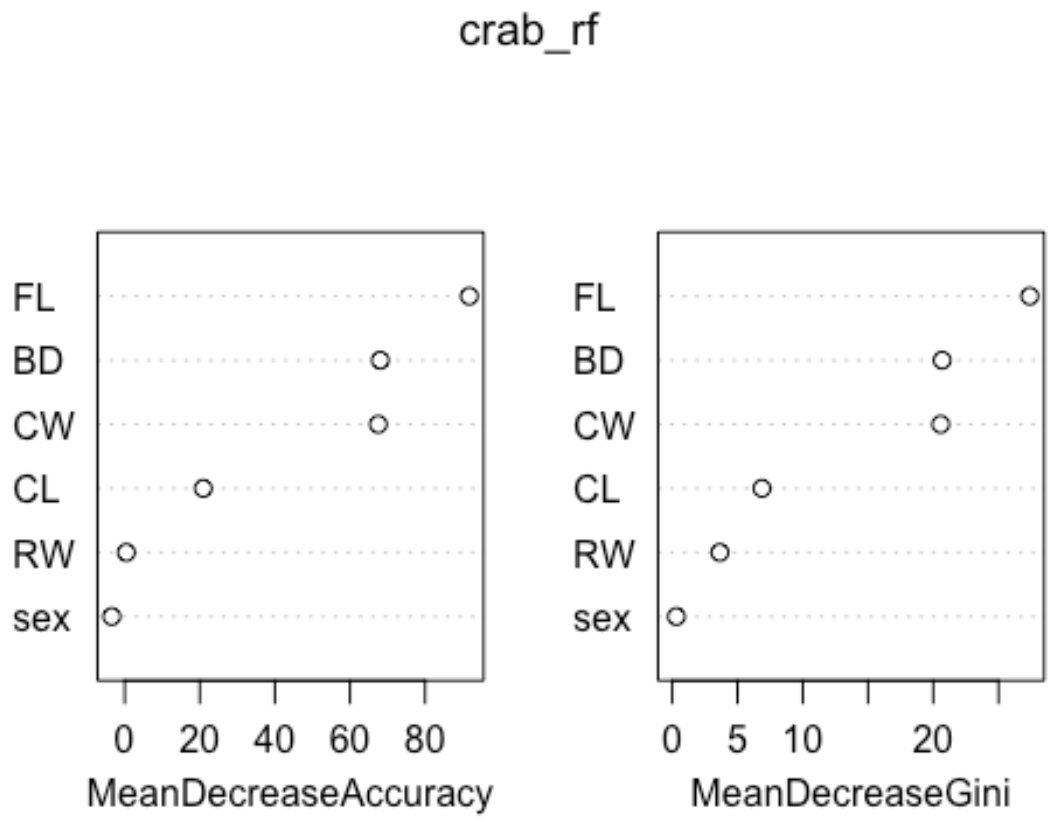
```
set.seed(6789)
RNGkind(sample.kind = "Rejection")

bm = which(crabs$sp == "B" & crabs$sex == "M") #blueMale
om = which(crabs$sp == "O" & crabs$sex == "M") #orangeMale
bf = which(crabs$sp == "B" & crabs$sex == "F") #blueFemale
of = which(crabs$sp == "O" & crabs$sex == "F") #orangeFemale

train_id = c(sample(bm, size = floor(0.80 * length(bm))),
sample(om, size = floor(0.80 * length(om))), sample(bf, size = floor(0.80 * l
ength(bf))),
sample(of, size = floor(0.80 * length(of))))

crab_train = crab[train_id,]
crab_test = crab[-train_id,]
```

(1a): As we can see from the output below, we can see that “FL”,“BD”,“CW” are the important variables. Compared to the single tree I had in the hw8, “FL”,“CW”, and “BD”, and “CL” are important variables for a single tree.

The training error is 0, and the testing error is 0.1764706

```
crab_rf = randomForest(sp ~ BD + CW + CL + RW + FL + sex, data=crab, subset=t
rain_id, importance=TRUE, mtry=5,ntree=1000)
```

```r
#variable importance plot
varImpPlot(crab_rf)
```




```r
# training error
train_rf_pred = predict(crab_rf, crabs[train_id,],type="class")
table(train_rf_pred, crabs$sp[train_id])

##
## train_rf_pred  B  O
##             B 80  0
##             O  0 80
```

```r
#testing error
test_rf_pred = predict(crab_rf, crabs[-train_id,],type="class")
table(test_rf_pred , crabs$sp[-train_id])
```

```
##
## test_rf_pred  B  O
##            B 16  2
##            O  4 18
```

```
6/34
```

```
## [1] 0.1764706
```

(1b): I pick M = 750, since this M gives the smallest testing error. The corresponding training error is 0.0125, and the testing error is 0.125.

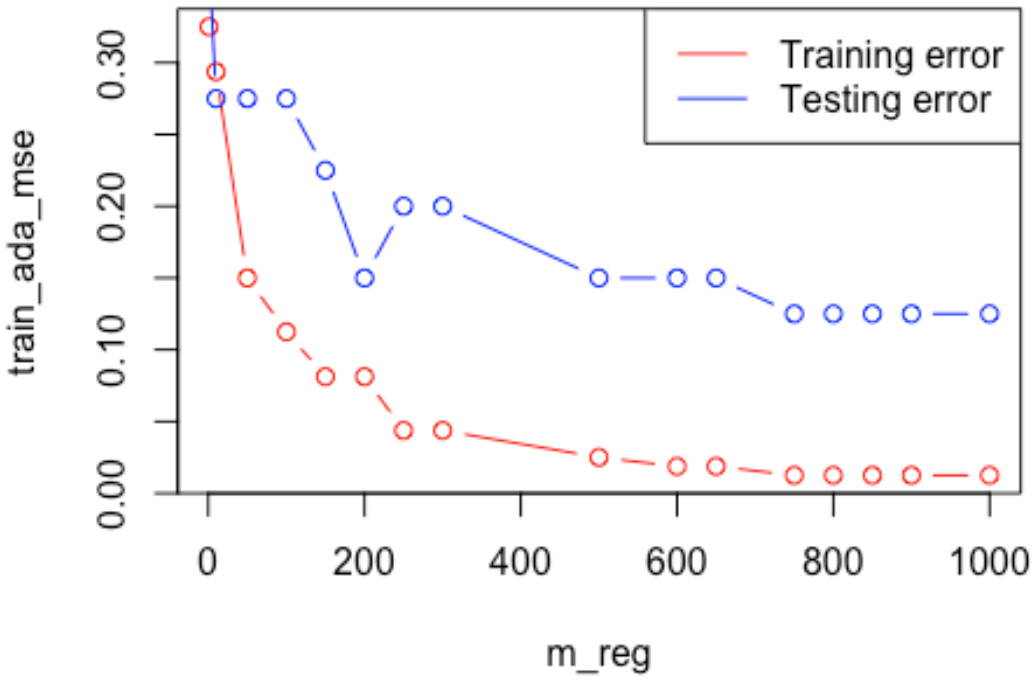
```
set.seed(6789)
m_reg = c(1,10,50,100,150,200,250,300,500,600,650,750,800,850,900,1000)

train_ada_mse = rep(NA,length(m_reg))
test_ada_mse = rep(NA,length(m_reg))



#recode variable
crab$sp = ifelse(crabs$sp=="B", 1, 0)
#use adaboost
for(k in 1:length(m_reg)){
  crab_ada = gbm(sp ~ BD + CW + CL + RW + FL + sex,
                 data=crab[train_id,], distribution = "adaboost", n.trees = m
_reg[k])

  #train error
  train_prob = predict(crab_ada,crab_train, n.trees =m_reg[k] , type = 'respo
nse')

  train_pred = ifelse(train_prob > 0.5, 1, 0)

  train_ada_mse[k] = mean(train_pred != crab$sp[train_id])



  #testing error
  test_prob = predict(crab_ada, crab_test, n.trees =m_reg[k] , type = "respon
se")

  test_pred = ifelse(test_prob > 0.5, 1, 0)

  test_ada_mse[k] = mean(test_pred != crab$sp[-train_id])
}
m_reg[which.min(test_ada_mse)]
```

```
## [1] 750
```

```
test_ada_mse[12]
```

```
## [1] 0.125

train_ada_mse[12]

## [1] 0.0125
```

```
plot(x = m_reg,y = train_ada_mse,col = "red",type ="b")
lines(x = m_reg,y = test_ada_mse,col = "blue",type = "b")
legend("topright", c("Training error", "Testing error"), col = c("red", "blue
"), lwd=1)
```



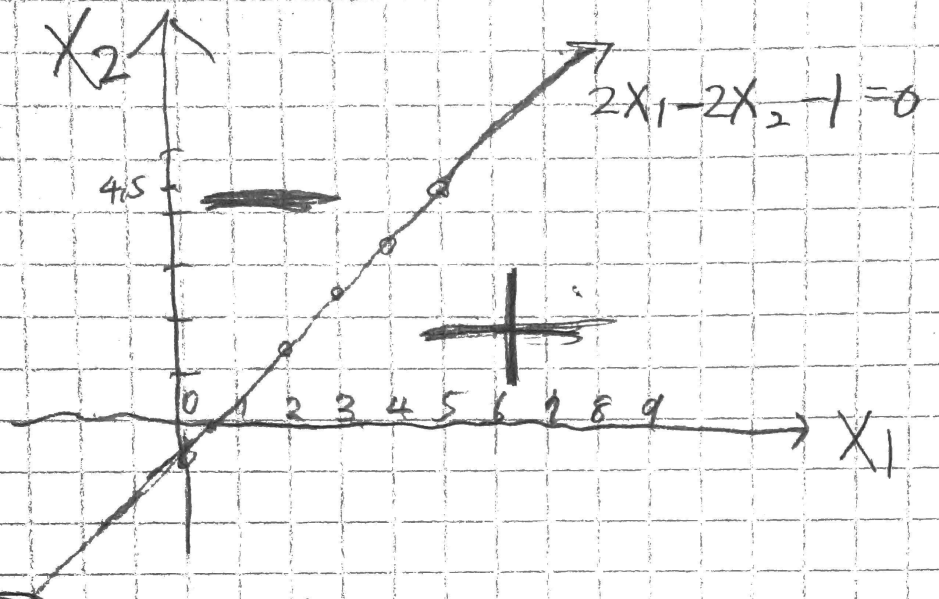
(1c): From hw8, we got the train error 0.05263158 and the test error 0.2903226. The Random forest gives the training error 0 and the testing error 0.1764706. The adaboost gives training error 0.0125 and testing error 0.125. In terms of performance, Adaboost performed the best as it gives the smallest testing error. The result are consistent across all methods as their training errors are quite small and the testing errors are relatively large.

Li Hsuan Lin (49109112)

(2a) hyperplane: $2X_1 - 2X_2 - 1 = 0$

Answer: plus sign: $2X_1 - 2X_2 - 1 > 0$

$\Leftrightarrow X_1 > X_2 + 0.5$



(2b) Using Slide 8 in Sum Lecture

$\beta = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$ is perpendicular to hyperplane

Let $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ be the point lie on the margin

Let X_0 be in the hyperplane, thus, $X_0^T\beta = \beta_0 = 1$

Using the formula $\left\langle \frac{\beta}{\|\beta\|}, X - X_0 \right\rangle = \frac{1}{\|\beta\|}(X^T\beta - X_0^T\beta) =$

$\left\langle \frac{\beta}{\|\beta\|}, X - X_0 \right\rangle = \sqrt{2}$

$\frac{1}{\sqrt{8}}\left(\begin{bmatrix} X_1 & X_2 \end{bmatrix}\begin{bmatrix} 2 \\ -2 \end{bmatrix} - 1\right) = \sqrt{2}$

$\frac{2X_1 - 2X_2 - 1}{\sqrt{8}} = \sqrt{2}$

$\Rightarrow 2X_1 - 2X_2 = 5$

$\left\langle \frac{\beta}{\|\beta\|}, X - X_0 \right\rangle = -\sqrt{2}$

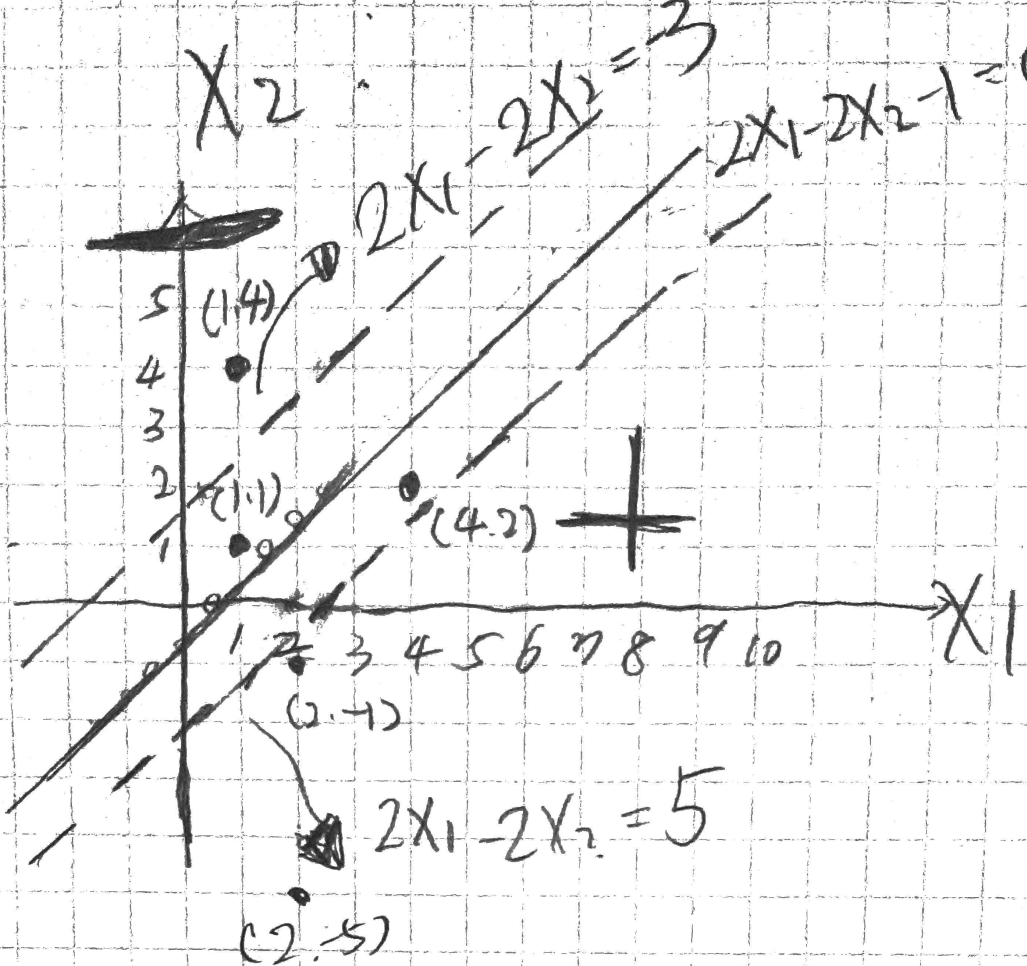
$\frac{1}{\sqrt{8}}\left(\begin{bmatrix} X_1 & X_2 \end{bmatrix}\begin{bmatrix} 2 \\ -2 \end{bmatrix} - 1\right) = -\sqrt{2}$

$\Rightarrow 2X_1 - 2X_2 = -3$

(1.2)

(2b)

(2c)



(2c)

+ : (2,-5), (2,-1), (4,2)

− : (1,1), (1,4)

(2d) sm/o (1,4), (2,-1), (2,-5) are classified correctly are outside of the margin, their slack values = 0