Exercise 8.4

Section 23 - Group 6 Project Groups (Bosan Hsu, Fan Liu, Jimeng Yin, Michael Liu, Richard Wang, Zhuoqian Zhang

```r
library(ISLR)
attach(Default)
set.seed(1)
names(Default)

## [1] "default" "student" "balance" "income"

train = sample(1:nrow(Default),nrow(Default)/2)
Default.train = Default[train, ]
Default.test = Default[-train, ]

glm.Default = glm(default~., data=Default, subset=train, family = binomial)
glm.probs = predict(glm.Default, newdata = Default.test, type = "response")
glm.pred = ifelse(glm.probs > 0.5, 'Yes', 'No')
table(Default.test$default, glm.pred)

##      glm.pred
##        No  Yes
##   No  4825   18
##   Yes  112   45

mean(Default.test$default != glm.pred)

## [1] 0.026

library(randomForest)

## Warning: 程辑包'randomForest'是用 R 版本 4.3.2 来建造的

## randomForest 4.7-1.1

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

set.seed(1)
bag.Default = randomForest(default~., data=Default, subset=train, mtry=3, ntree=250)
bag.probs = predict(bag.Default, newdata = Default.test, type = "prob")
bag.pred = ifelse(bag.probs[,2] > 0.5, 'Yes', 'No')
table(Default.test$default, bag.pred)

##      bag.pred
##        No  Yes
```

```
##   No   4799    44
##   Yes  108     49
```

```r
mean(Default.test$default != bag.pred)
```

```
## [1] 0.0304
```

```r
set.seed(1)
rf.Default = randomForest(default~., data=Default, subset=train, mtry=2,
 ntree=250)
rf.probs = predict(rf.Default, newdata = Default.test, type = "prob")
rf.pred = ifelse(rf.probs[,2] > 0.5, 'Yes', 'No')
table(Default.test$default, rf.pred)
```

```
##       rf.pred
##         No  Yes
##   No  4805   38
##   Yes  108   49
```

```r
mean(Default.test$default != rf.pred)
```

```
## [1] 0.0292
```

```r
library(gbm)
```

```
## Warning: 程辑包'gbm'是用 R 版本 4.3.2 来建造的
```

```
## Loaded gbm 2.1.8.1
```

```r
Default.train$default = as.numeric(Default.train$default) - 1
Default.test$default = as.numeric(Default.test$default) - 1
set.seed(1)
boost.fit = gbm(default~., data = Default.train, distribution = "bernou
lli", n.trees = 250)
boost.probs = predict(boost.fit, newdata = Default.test, type = "respon
se")
```

```
## Using 250 trees...
```

```r
boost.pred = ifelse(boost.probs > 0.5, 1, 0)
table(Default.test$default, boost.pred)
```

```
##    boost.pred
##        0    1
##   0 4816   27
##   1  106   51
```

```r
mean(Default.test$default != boost.pred)
```

```
## [1] 0.0266
```

```r
set.seed(1)
lm.Default <- lm(default ~ ., data = Default.train)
```

```
lm.probs <- predict(lm.Default, newdata = Default.test, type = "respons
e")
lm.pred <- ifelse(lm.probs > 0.5, 1, 0)
table(Default.test$default, lm.pred)

##    lm.pred
##       0
##   0 4843
##   1  157

mean(Default.test$default != lm.pred)

## [1] 0.0314
```

The accuracy of logistic is the best.Linear is the worst.