Homework 3

Amy Nguyen

```
library(tidyverse)
library(ISLR)
library(glmnet)
library(tree)
library(maptree)
library(randomForest)
library(gbm)
library(ROCR)
```

Predicting carseats sales using regularized regression methods

```
set.seed(123)
dat <- model.matrix(Sales~., Carseats)
train = sample(nrow(dat), 30)
x.train = dat[train, ]
y.train = Carseats[train, ]$Sales
# The rest as test data
x.test = dat[-train, ]
y.test = Carseats[-train, ]$Sales
```

###(a)

```
set.seed(123)
lambda.list.ridge = 1000 * exp(seq(0, log(1e-5), length = 100))
ridge_mod = glmnet(dat, Carseats$Sales, alpha = 0, lambda = lambda.list.ridge)
```

```
cv.out.ridge = cv.glmnet(x.train, y.train, alpha = 0, nfolds = 5)
bestlam = cv.out.ridge$lambda.min
bestlam
```

```
## [1] 0.1265465
```

```
ridge.mod = glmnet(x.train, y.train, alpha = 0, lambda = bestlam)
out <- glmnet(dat, Carseats$Sales, alpha = 0)
predict(out, type = "coefficients", s = bestlam)
```

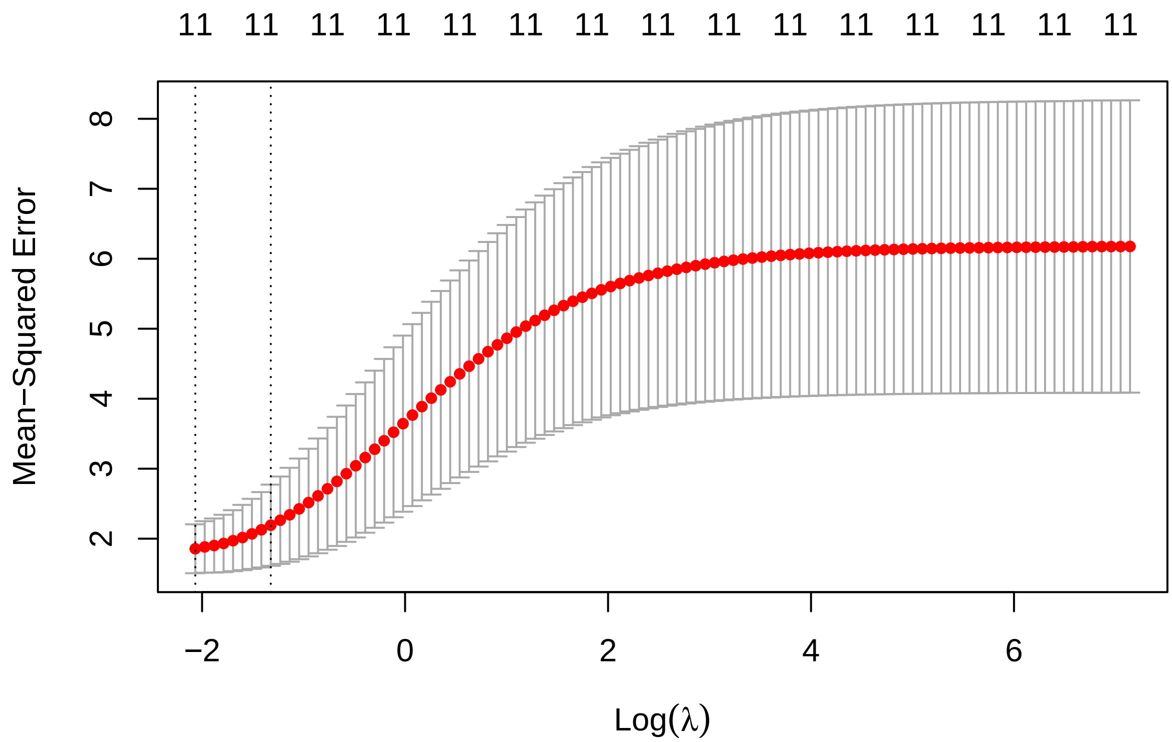
```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)      6.3085006155
## (Intercept)      .
## CompPrice        0.0806782688
## Income           0.0146576527
## Advertising      0.1116027505
## Population       0.0001900746
## Price           -0.0860089210
```

```
## ShelveLocGood     4.4188303312
## ShelveLocMedium   1.6718683096
## Age              -0.0434249353
## Education        -0.0209766364
## UrbanYes          0.0969684470
## USYes            -0.0749323836
```

The best tuning parameter is $\lambda = 0.1265$.

###(b)

```
plot(cv.out.ridge)
```



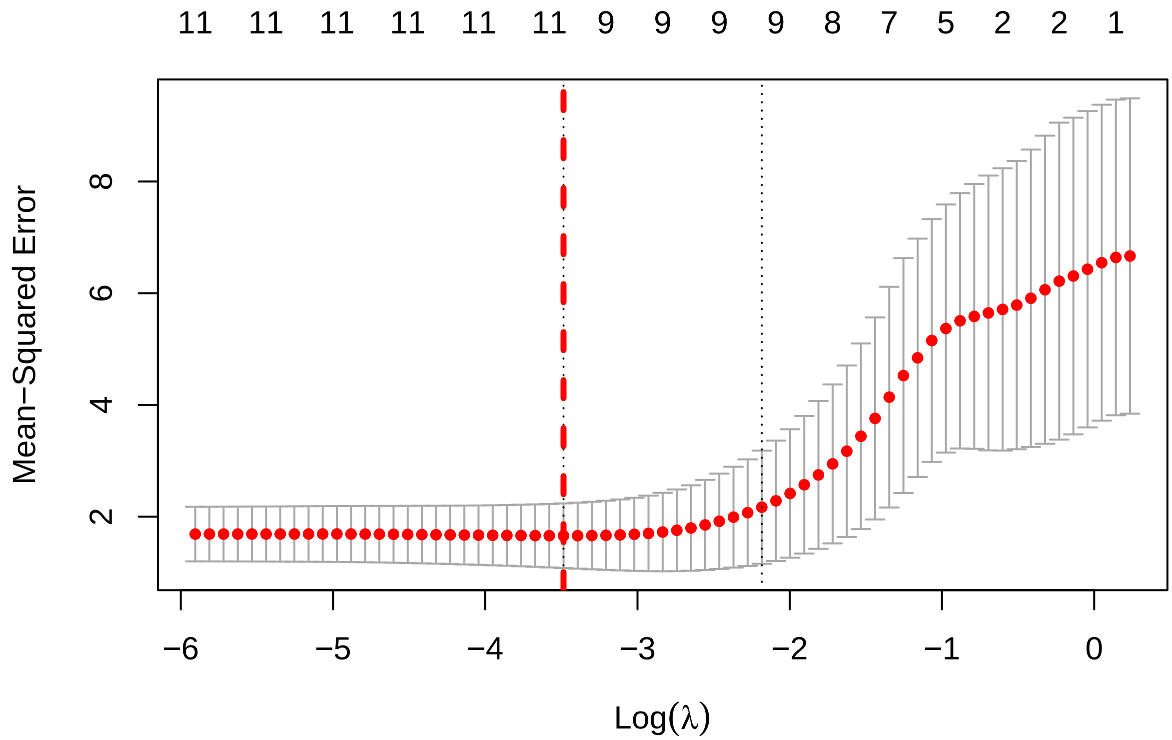

```
ridge.pred <- predict(ridge.mod, s = bestlam, newx = x.test)
mean((ridge.pred-y.test)^2)
```

```
## [1] 1.460228
```

The plot displays the training MSE as a function of λ and the test MSE 1.4602.

###(c)

```
lambda.list.lasso = 2 * exp(seq(0, log(1e-4), length = 100))
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = lambda.list.lasso)
set.seed(1)
cv.out.lasso = cv.glmnet(x.train, y.train, alpha = 1)
plot(cv.out.lasso)
abline(v = log(cv.out.lasso$lambda.min), col="red", lwd=3, lty=2)
```




```r
bestlam <- cv.out.lasso$lambda.min
bestlam
```

```
## [1] 0.03062589
```

```r
out = glmnet(dat, Carseats$Sales, alpha=1, lambda=lambda.list.lasso)
predict(out,type="coefficients",s=bestlam)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                             s1
## (Intercept)       5.9209155071
## (Intercept)       .
## CompPrice         0.0876757470
## Income            0.0144676586
## Advertising       0.1106031197
## Population        0.0000681292
## Price            -0.0919501258
## ShelveLocGood     4.6594335721
## ShelveLocMedium   1.8044057958
## Age              -0.0440606950
## Education        -0.0100955332
## UrbanYes          0.0453929531
## USYes             .
```

Using 10-fold CV, the optimal tuning parameter λ is 0.0306. The

```r
lasso.pred = predict(lasso.mod, s = bestlam, newx = x.test)
mean((lasso.pred-y.test)^2)
```

```
## [1] 1.468843
```

The test MSE associated with the tuning parameter is 1.709, which is slightly larger, yet very similar to the test MSE of the ridge regression chosen by cross-validation.

Analyzing Drug Use

```
drug <- read_csv('drug.csv',
                 col_names = c('ID','Age','Gender','Education',
                               'Country','Ethnicity','Nscore',
                               'Escore','Oscore','Ascore','Cscore',
                               'Impulsive','SS','Alcohol','Amphet','Amyl',
                               'Benzos','Caff','Cannabis','Choc',
                               'Coke','Crack','Ecstasy','Heroin','Ketamine',
                               'Legalh','LSD','Meth','Mushrooms',
                               'Nicotine','Semer','VSA'))
```

###(a)

```
drug <- drug %>% mutate(recent_cannabis_use =
                                factor(ifelse(Cannabis >= "CL3", "Yes", "No"),
                                       levels=c("No", "Yes")))

class(drug$recent_cannabis_use)
```

```
## [1] "factor"
```

```
levels(drug$recent_cannabis_use)
```

```
## [1] "No"  "Yes"
```

(b)

```
drug_subset <- drug %>% select(Age:SS, recent_cannabis_use)
```

(c)

```
set.seed(1)
train = sample(1:nrow(drug_subset), 1100)
drug_train <- drug_subset[train,]
drug_test <- drug_subset[-train,]
```

(d)

```
glm.fit <- glm(recent_cannabis_use ~ ., data=drug_train, family=binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = recent_cannabis_use ~ ., family = binomial, data = drug_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8883  -0.6072   0.1572   0.5557   2.5543
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.01280    0.22038   4.596 4.31e-06 ***
## Age         -0.84977    0.10690  -7.949 1.88e-15 ***
## Gender      -0.47832    0.18770  -2.548  0.01082 *
```

```
## Education   -0.42603    0.09084  -4.690 2.73e-06 ***
## Country     -0.95550    0.13521  -7.067 1.59e-12 ***
## Ethnicity    1.08760    0.61864   1.758  0.07874 .
## Nscore      -0.08237    0.10476  -0.786  0.43174
## Escore      -0.08420    0.11283  -0.746  0.45553
## Oscore       0.73603    0.10590   6.950 3.65e-12 ***
## Ascore      -0.02724    0.09619  -0.283  0.77701
## Cscore      -0.34296    0.10546  -3.252  0.00115 **
## Impulsive   -0.16715    0.11703  -1.428  0.15321
## SS           0.62867    0.12386   5.076 3.86e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1523.00  on 1099  degrees of freedom
## Residual deviance:  895.35  on 1087  degrees of freedom
## AIC: 921.35
##
## Number of Fisher Scoring iterations: 5
```

(e)

```
tree.drug <- tree(recent_cannabis_use~., data = drug_train)
```

(f)

```
set.seed(3)
cv = cv.tree(tree.drug, FUN=prune.misclass, K=5)
cv$size
```

```
## [1] 7 6 4 2 1
```

```
cv$dev
```

```
## [1] 253 253 258 278 527
```

```
best_size = min(cv$size[cv$dev == min(cv$dev)])
best_size
```

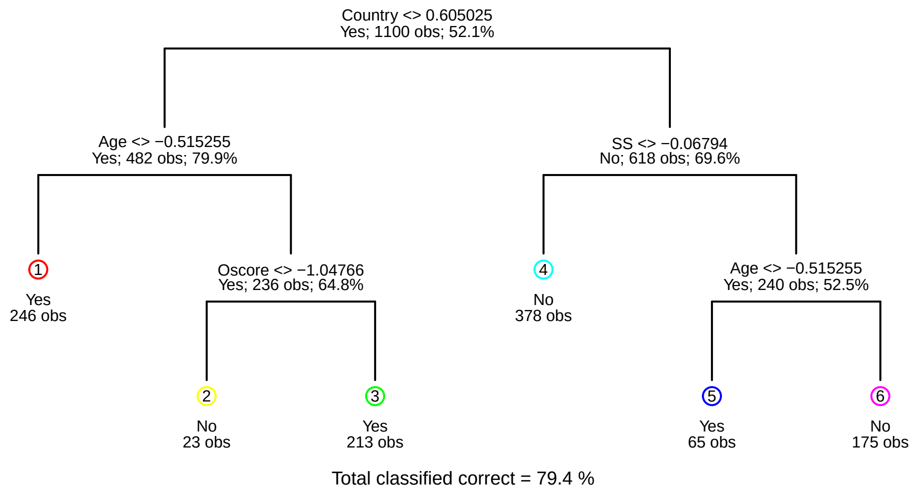
```
## [1] 6
```

There is a tie between tree of size 6 and size 7 with the same minimum cross validated error rate of 253. The best size obtained tree is of size 6.

(g)

```
pruned.drug = prune.misclass(tree.drug, best = best_size)
draw.tree(pruned.drug, nodeinfo = TRUE, cex = 0.5)
title("Classification Tree for Drug Use on Training Set")
```

Classification Tree for Drug Use on Training Set



Country is the first variable split in the decision tree.

(h)

```
pred.drug = predict(pruned.drug, drug_test, type = "class")
error = table(pred.drug, drug_test$recent_cannabis_use)
error
```

```
##
## pred.drug  No Yes
##       No  298  85
##       Yes  61 341
```

```
TPR = error[2,2]/sum(error[c(1,2),2])
TPR
```

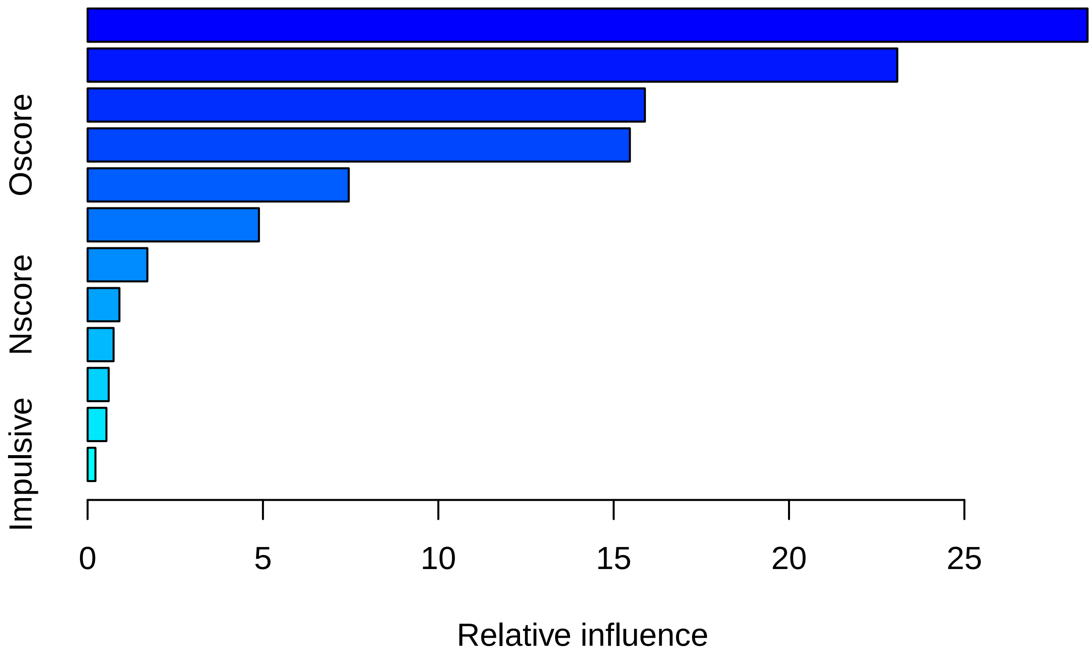
```
## [1] 0.8004695
```

```
FPR = error[2,1]/sum(error[c(1,2),1])
FPR
```

```
## [1] 0.1699164
```

(i)

```
drug.gbm <- gbm(recent_cannabis_use ~ ., data=drug_train, distribution="gaussian", n.trees=1000, shrinka
summary(drug.gbm)
```




```
##                 var    rel.inf
## Country     Country 28.5125788
## Age             Age 23.0869456
## SS               SS 15.8905516
## Oscore       Oscore 15.4623741
## Cscore       Cscore  7.4473145
## Education Education  4.8857914
## Gender       Gender  1.7037889
## Nscore       Nscore  0.9071954
## Ethnicity Ethnicity  0.7410988
## Escore       Escore  0.6024034
## Ascore       Ascore  0.5365710
## Impulsive Impulsive  0.2233866
```

The most important predictors are Age and SS having the most influence, and also Nscore and Impulsive.

(j)

```r
set.seed(131)
drug.random <- randomForest(recent_cannabis_use ~ ., data=drug_train, importance=TRUE)
drug.random
```

```
##
## Call:
##  randomForest(formula = recent_cannabis_use ~ ., data = drug_train,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 18.91%
## Confusion matrix:
##      No Yes class.error
## No  417 110   0.2087287
```

```
## Yes  98 475   0.1710297
```

```r
importance(drug.random)
```

```
##                    No         Yes MeanDecreaseAccuracy MeanDecreaseGini
## Age       31.62981434 31.6464444            43.432165        70.164842
## Gender    10.08550145  4.9233922            10.887755        13.867941
## Education 10.18388704  9.0684014            14.325813        36.023330
## Country   54.28995563 22.8292190            52.859087        81.515436
## Ethnicity  0.07829972  3.5170960             2.635203         5.281693
## Nscore     4.14674935  1.7008700             4.202319        40.274430
## Escore     6.90182120 -1.8409118             3.880629        37.227429
## Oscore    26.61007337 23.1812997            35.726320        76.788426
## Ascore     4.57229978  3.1729270             5.660958        36.802276
## Cscore    14.92408633  9.2539460            17.192369        52.905107
## Impulsive 14.60273949  0.3031873            11.626674        30.163588
## SS        30.88557521 18.2149686            34.917883        65.632835
```

The out of bag estimate error is 18.91%. 3 variables were randomly considered at each split in the trees and 500 trees were used. The order of important variables are very similar for random forrest and boosting models, but are some differences such as between Age and Oscore.

(k)

```r
set.seed(123)
prob_boost = predict(drug.gbm, newdata = drug_test, type = "response")
```

```
## Using 1000 trees...
```

```r
yhat_boost = ifelse(prob_boost >= 0.2, "Yes", "No")
prob_rand = predict(drug.random, newdata = drug_test, type = "prob")
yhat_tree = ifelse(prob_rand[, 2] >= 0.2, "Yes", "No")
boost.matrix = table(true = drug_test$recent_cannabis_use, pred = yhat_boost)
rf.matrix = table(true = drug_test$recent_cannabis_use, pred = yhat_tree)

boost.matrix
```

```
##      pred
## true  Yes
##   No  359
##   Yes 426
```

```r
rf.matrix
```

```
##      pred
## true   No Yes
##   No  152 207
##   Yes  12 414
```

In the random tree model, 414/621 of people predicted and did in fact use cannabis recently.