SDSHW2

Bailey Brady, Andrew Chen, Cristian Sigala, Cherry Sun

3/21/2021

Question 5 Section 5.4

a)

```
default_model1 = glm(default~income+balance, family = 'binomial')
summary(default_model1)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

b)

i.

```
default_samplesize = floor(0.75 * nrow(Default))
set.seed(1000)
default_train_index <- sample(seq_len(nrow(Default)), size = default_samplesize)
Default_train <- Default[default_train_index,]
Default_test <- Default[-default_train_index,]
```

ii.

```r
default_model2 = glm(default~income+balance, data = Default_train, family = 'binomial')
summary(default_model2)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial",
##     data = Default_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4637  -0.1423  -0.0571  -0.0213   3.7203
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.153e+01  5.023e-01 -22.956  < 2e-16 ***
## income       2.174e-05  5.788e-06   3.756 0.000173 ***
## balance      5.614e-03  2.634e-04  21.314  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2144.8  on 7499  degrees of freedom
## Residual deviance: 1164.7  on 7497  degrees of freedom
## AIC: 1170.7
##
## Number of Fisher Scoring iterations: 8
```

iii.

```r
default_predict = predict(default_model2, Default_test, type = "response")
yhat_predict_test <- ifelse(default_predict > 0.5, 1, 0)
```

iv.

```r
table_def_test <- table(y = Default_test$default, yhat = yhat_predict_test)
accuracy_def_test <- sum(diag(table_def_test))/ sum(table_def_test)
1-accuracy_def_test
```

```
## [1] 0.0268
```

c)

1.

```r
default_samplesize = floor(0.75 * nrow(Default))
default_train_index <- sample(seq_len(nrow(Default)), size = default_samplesize)
Default_train <- Default[default_train_index,]
Default_test <- Default[-default_train_index,]




default_model2 = glm(default~income+balance, data = Default_train, family = 'binomial')
```

```r
summary(default_model2)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial",
##     data = Default_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4332  -0.1402  -0.0555  -0.0201   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.139e+01  5.058e-01 -22.509   <2e-16 ***
## income       1.309e-05  5.805e-06   2.255   0.0241 *
## balance      5.702e-03  2.688e-04  21.214   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2144.8  on 7499  degrees of freedom
## Residual deviance: 1162.1  on 7497  degrees of freedom
## AIC: 1168.1
##
## Number of Fisher Scoring iterations: 8
```

```r
default_predict = predict(default_model2, Default_test, type = "response")
yhat_predict_test <- ifelse(default_predict > 0.5, 1, 0)


table_def_test <- table(y = Default_test$default, yhat = yhat_predict_test)

accuracy_def_test <- sum(diag(table_def_test))/ sum(table_def_test)
1-accuracy_def_test
```

```
## [1] 0.0256
```

2.

```r
default_samplesize = floor(0.75 * nrow(Default))
default_train_index <- sample(seq_len(nrow(Default)), size = default_samplesize)
Default_train <- Default[default_train_index,]
Default_test <- Default[-default_train_index,]




default_model2 = glm(default~income+balance, data = Default_train, family = 'binomial')
summary(default_model2)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial",
##     data = Default_train)
##
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.1984  -0.1483  -0.0599  -0.0223   3.6962
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.121e+01  4.904e-01 -22.861  < 2e-16 ***
## income       1.661e-05  5.789e-06   2.868  0.00413 **
## balance      5.539e-03  2.578e-04  21.483  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2198.9  on 7499  degrees of freedom
## Residual deviance: 1213.4  on 7497  degrees of freedom
## AIC: 1219.4
##
## Number of Fisher Scoring iterations: 8
```

```r
default_predict = predict(default_model2, Default_test, type = "response")
yhat_predict_test <- ifelse(default_predict > 0.5, 1, 0)


table_def_test <- table(y = Default_test$default, yhat = yhat_predict_test)
accuracy_def_test <- sum(diag(table_def_test))/ sum(table_def_test)
1-accuracy_def_test
```

```
## [1] 0.0216
```

3.

```r
default_samplesize = floor(0.75 * nrow(Default))
default_train_index <- sample(seq_len(nrow(Default)), size = default_samplesize)
Default_train <- Default[default_train_index,]
Default_test <- Default[-default_train_index,]




default_model2 = glm(default~income+balance, data = Default_train, family = 'binomial')
summary(default_model2)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial",
##     data = Default_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4621  -0.1432  -0.0554  -0.0201   3.7516
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.155e+01  5.015e-01 -23.030  < 2e-16 ***
## income       1.691e-05  5.746e-06   2.943  0.00325 **
## balance      5.722e-03  2.636e-04  21.704  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2219.0  on 7499  degrees of freedom
## Residual deviance: 1181.3  on 7497  degrees of freedom
## AIC: 1187.3
##
## Number of Fisher Scoring iterations: 8
```

```
default_predict = predict(default_model2, Default_test, type = "response")
yhat_predict_test <- ifelse(default_predict > 0.5, 1, 0)


table_def_test <- table(y = Default_test$default, yhat = yhat_predict_test)
accuracy_def_test <- sum(diag(table_def_test))/ sum(table_def_test)
1-accuracy_def_test
```

```
## [1] 0.0268
```

d)

```
default_samplesize = floor(0.75 * nrow(Default))
default_train_index <- sample(seq_len(nrow(Default)), size = default_samplesize)
Default_train <- Default[default_train_index,]
Default_test <- Default[-default_train_index,]



default_model2 = glm(default~income+balance+student, data = Default_train, family = 'binomial')
summary(default_model2)
```

```
##
## Call:
## glm(formula = default ~ income + balance + student, family = "binomial",
##     data = Default_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4598  -0.1504  -0.0602  -0.0225   3.6713
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.063e+01  5.463e-01 -19.465  < 2e-16 ***
## income       5.640e-06  9.164e-06   0.615  0.53826
## balance      5.574e-03  2.548e-04  21.872  < 2e-16 ***
## studentYes  -7.174e-01  2.668e-01  -2.689  0.00717 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2265.8  on 7499  degrees of freedom
```

```
## Residual deviance: 1236.4  on 7496  degrees of freedom
## AIC: 1244.4
##
## Number of Fisher Scoring iterations: 8
```

```
default_predict = predict(default_model2, Default_test, type = "response")
yhat_predict_test <- ifelse(default_predict > 0.5, 1, 0)


table_def_test <- table(y = Default_test$default, yhat = yhat_predict_test)
accuracy_def_test <- sum(diag(table_def_test))/ sum(table_def_test)
1-accuracy_def_test
```

```
## [1] 0.024
```

It does not seem that adding the student variables leads to a reduction in the test error rate

```
detach(Default)
```

Question 7 Section 5.4

a

```
#Weekly <- read.csv("Weekly.csv")
head(Weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

```
regression_a <- glm(as.factor(Direction) ~ Lag1 + Lag2, data= Weekly, family = 'binomial')
summary(regression_a)
```

```
##
## Call:
## glm(formula = as.factor(Direction) ~ Lag1 + Lag2, family = "binomial",
##     data = Weekly)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.623  -1.261   1.001   1.083   1.506
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22122    0.06147   3.599 0.000319 ***
## Lag1        -0.03872    0.02622  -1.477 0.139672
```

```
## Lag2         0.06025    0.02655   2.270 0.023232 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1488.2  on 1086  degrees of freedom
## AIC: 1494.2
##
## Number of Fisher Scoring iterations: 4
```

b

```r
regression_b <- glm(as.factor(Direction) ~ Lag1 + Lag2, data= Weekly[-1,], family = 'binomial')
summary(regression_b)
```

```
##
## Call:
## glm(formula = as.factor(Direction) ~ Lag1 + Lag2, family = "binomial",
##     data = Weekly[-1, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6258  -1.2617   0.9999   1.0819   1.5071
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22324    0.06150   3.630 0.000283 ***
## Lag1        -0.03843    0.02622  -1.466 0.142683
## Lag2         0.06085    0.02656   2.291 0.021971 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1494.6  on 1087  degrees of freedom
## Residual deviance: 1486.5  on 1085  degrees of freedom
## AIC: 1492.5
##
## Number of Fisher Scoring iterations: 4
```

c

```r
predicted_dir <- predict(regression_b, newdata = Weekly[1,], type = 'response') > 0.5
predicted_dir
```

```
##    1
## TRUE
```

```r
head(Weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

Using the model from b, predicted_dir outputs true which means that it's predicting that the first obversation will go up. However, we can see from that dataset that the first observation goes down. Therefore, out obversation is not correctly classified

#d

```r
n<- nrow(Weekly)
error_matrix<- matrix(0, nrow = n) #creating the matrix with the number of rows

for(i in 1:n){
    regression <- glm(as.factor(Direction) ~ Lag1 + Lag2, data= Weekly[-i,], family = 'binomial')
    post_prob <- predict(regression, newdata = Weekly[i,], type = 'response') >0.5
    #make the prediction
    current <- Weekly[i,]
    up <- current$Direction == 'Up' #getting the actual class label
    if(post_prob != up){ #if the prediction is incorrect, error matrix = 1
      error_matrix[i] <- 1
    }else{ #if the prection is correct, error matrix = 0
      error_matrix[i] <- 0
    }
}
head(error_matrix)
```

```
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    0
## [4,]    1
## [5,]    0
## [6,]    1
```

e

```r
mean(error_matrix)
```

```
## [1] 0.4499541
```

getting the average of error_matrix the test error is 0.4499541 which is 44.99541%

Question 9 in Section 6.8

a

```r
head(College)
```

```
##                              Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
## Agnes Scott College              Yes  417    349    137        60        89
## Alaska Pacific University        Yes  193    146     55        16        44
## Albertson College                Yes  587    479    158        38        62
##                              F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
## Agnes Scott College                  510          63    12960       5450   450
## Alaska Pacific University            249         869     7560       4120   800
## Albertson College                    678          41    13500       3335   500
##                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
## Agnes Scott College               875  92       97       7.7          37  19016
## Alaska Pacific University        1500  76       72      11.9           2  10922
## Albertson College                 675  67       73       9.4          11   9727
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

```r
set.seed(1)

index <- sample(1:nrow(College), size = nrow(College) * 0.3)
train <- College[index,]
test <- College[-index,]

head(train)
```

```
##                                 Private Apps Accept Enroll Top10perc Top25perc
## University of Southern Colorado      No 1401   1239    605        10        34
## College of Notre Dame               Yes  344    264     97        11        42
## Salisbury State University           No 4216   2290    736        20        52
## Regis College                       Yes  427    385    143        18        38
## La Salle University                 Yes 2929   1834    622        20        56
## Illinois State University            No 8681   6695   2408        10        35
```

```
##                                F.Undergrad P.Undergrad Outstate Room.Board
## University of Southern Colorado        3716         675     7100       4380
## College of Notre Dame                   500         331    12600       5520
## Salisbury State University             4296        1027     5130       4690
## Regis College                           581         533    12700       5800
## La Salle University                    2738        1662    12600       5610
## Illinois State University             15701        1823     7799       3403
##                                Books Personal PhD Terminal S.F.Ratio
## University of Southern Colorado   540     2948  63       88      19.4
## College of Notre Dame             630     2250  77       80      10.4
## Salisbury State University        600     1450  73       75      17.9
## Regis College                     450      700  81       85      10.3
## La Salle University               450     3160  90       90      15.1
## Illinois State University         537     2605  77       84      21.0
##                                perc.alumni Expend Grad.Rate
## University of Southern Colorado           0   5389        36
## College of Notre Dame                     7   9773        43
## Salisbury State University               18   5125        56
## Regis College                            37  11758        84
## La Salle University                       9   9084        84
## Illinois State University                16   5569        54
```

```
head(test)
```

```
##                          Private Apps Accept Enroll Top10perc Top25perc
## Adelphi University           Yes 2186   1924    512        16        29
## Adrian College               Yes 1428   1097    336        22        50
## Agnes Scott College          Yes  417    349    137        60        89
## Alaska Pacific University    Yes  193    146     55        16        44
## Albertson College            Yes  587    479    158        38        62
## Albertus Magnus College      Yes  353    340    103        17        45
##                          F.Undergrad P.Undergrad Outstate Room.Board Books
## Adelphi University              2683        1227    12280       6450   750
## Adrian College                  1036          99    11250       3750   400
## Agnes Scott College              510          63    12960       5450   450
## Alaska Pacific University        249         869     7560       4120   800
## Albertson College                678          41    13500       3335   500
## Albertus Magnus College          416         230    13290       5720   500
##                          Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Adelphi University           1500  29       30      12.2          16  10527
## Adrian College               1165  53       66      12.9          30   8735
## Agnes Scott College           875  92       97       7.7          37  19016
## Alaska Pacific University    1500  76       72      11.9           2  10922
## Albertson College             675  67       73       9.4          11   9727
## Albertus Magnus College      1500  90       93      11.5          26   8861
##                          Grad.Rate
## Adelphi University              56
## Adrian College                  54
## Agnes Scott College             59
## Alaska Pacific University       15
## Albertson College               55
## Albertus Magnus College         63
```

b)

```r
lm6.8 <- lm(Apps ~ ., data = train)
pred <- predict(lm6.8, test)

y_test <- test$Apps
y_train <- train$Apps

RMSE(pred, y_test)
```

```
## [1] 1182.814
```

```r
error <- postResample(pred, y_test)
error
```

```
##         RMSE     Rsquared          MAE
## 1182.8142919    0.8945488  676.8722609
```

c)

```r
train_matrix <- model.matrix(Apps ~ ., data = train)
test_matrix <- model.matrix(Apps ~ ., data = test)
grid <- 10^seq(4, -2, length = 100)
ridge <- cv.glmnet(train_matrix, y_train, alpha = 0, lambda = grid, thresh = 1e-12)
best_lambda <- ridge$lambda.min
best_lambda
```

```
## [1] 0.01
```

```r
pred_ridge <- predict(ridge, newx = test_matrix, s = best_lambda)
error2 <- postResample(pred_ridge, y_test)
error2
```

```
##         RMSE     Rsquared          MAE
## 1182.8009141    0.8945511  676.8630136
```

```r
extract.coef(ridge)
```

```
##                       Value SE Coefficient
## X.Intercept. -197.47473828 NA (Intercept)
## PrivateYes   -180.38813612 NA  PrivateYes
## Accept          1.90690758 NA      Accept
## Enroll         -1.45578542 NA      Enroll
## Top10perc      62.93787747 NA   Top10perc
## Top25perc     -22.01200822 NA   Top25perc
## F.Undergrad     0.06572420 NA F.Undergrad
## P.Undergrad    -0.04219349 NA P.Undergrad
## Outstate       -0.10657172 NA    Outstate
## Room.Board      0.20111879 NA  Room.Board
## Books           0.20257431 NA       Books
## Personal       -0.09686444 NA    Personal
## PhD           -22.11998245 NA         PhD
## Terminal       10.62121069 NA    Terminal
## S.F.Ratio      23.76937608 NA   S.F.Ratio
```

```
## perc.alumni    -3.15148253 NA perc.alumni
## Expend          0.04550276 NA      Expend
## Grad.Rate       3.23766923 NA   Grad.Rate
```

d)

```r
lasso <- cv.glmnet(train_matrix, y_train, alpha = 1, lambda = grid, thresh = 1e-12)
best_lambda <- lasso$lambda.min
best_lambda
```

```
## [1] 0.01
```

```r
pred_lasso <- predict(lasso, newx = test_matrix, s = best_lambda)
error3 <- postResample(pred_lasso, y_test)
error3
```

```
##         RMSE     Rsquared          MAE
## 1182.7816825    0.8945544  676.8382816
```

```r
extract.coef(lasso)
```

```
##                      Value SE Coefficient
## X.Intercept. -197.33471098 NA (Intercept)
## PrivateYes   -180.45079619 NA  PrivateYes
## Accept          1.90689311 NA      Accept
## Enroll         -1.45529673 NA      Enroll
## Top10perc      62.93150131 NA   Top10perc
## Top25perc     -22.00644725 NA   Top25perc
## F.Undergrad     0.06562954 NA F.Undergrad
## P.Undergrad    -0.04215883 NA P.Undergrad
## Outstate       -0.10655848 NA    Outstate
## Room.Board      0.20110095 NA  Room.Board
## Books           0.20252577 NA       Books
## Personal       -0.09682944 NA    Personal
## PhD           -22.11512632 NA         PhD
## Terminal       10.61571444 NA    Terminal
## S.F.Ratio      23.76491268 NA   S.F.Ratio
## perc.alumni    -3.15046969 NA perc.alumni
## Expend          0.04549975 NA      Expend
## Grad.Rate       3.23487668 NA   Grad.Rate
```

Based on all the errors it seems like all of the models were quite similar in results. They each had about 0.89 R-squared value which is pretty high. Therefore, we can say that the models all performed very well and had high accuracy.

Question 10 in Section 6.8

A

```
set.seed(1)
p = 20
n = 1000
x = matrix(rnorm(n*p),n,p)
beta <- rnorm(p)
beta[1] <- 0
beta[4] <- 0
beta[6] <- 0
beta[10] <- 0
beta[20] <- 0
epsilon <- rnorm(p)

y <- x%*%beta + epsilon
```

B

```
data <- data.frame(y,x)

dim<-dim(data)[1]
set.seed(5)
rows<-sample(1:dim,dim/10)
test<-data[rows,]
dim(test)
```

```
## [1] 100  21
```

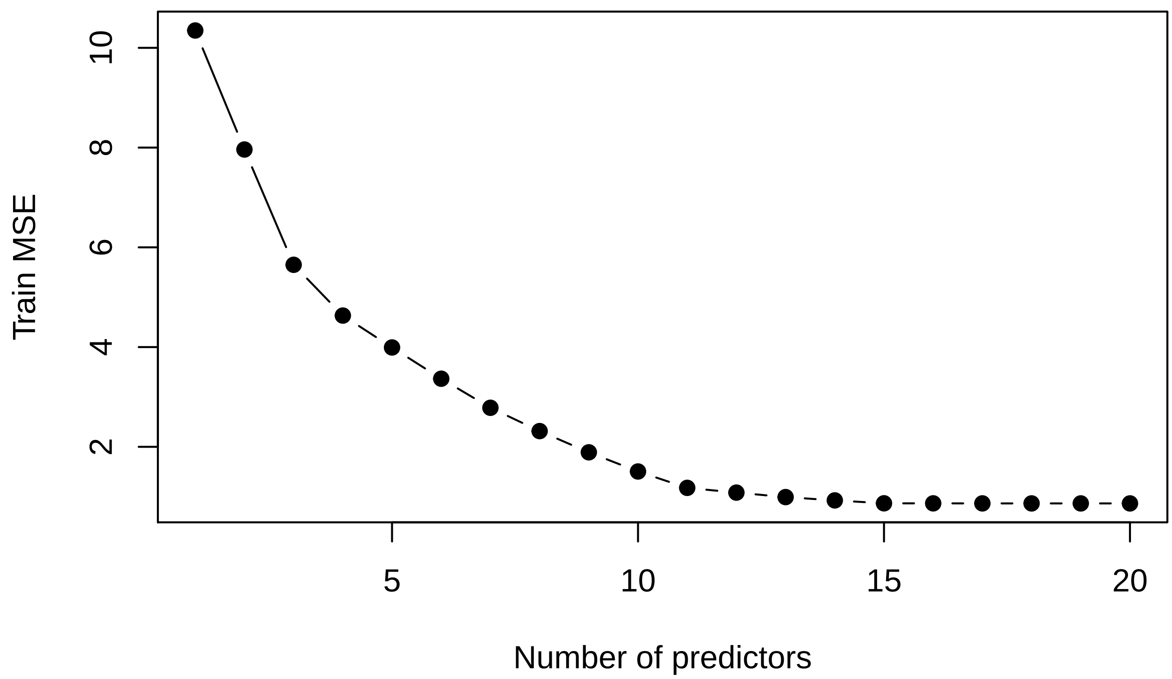
```
train<-data[-rows,]
dim(train)
```

```
## [1] 900  21
```

C

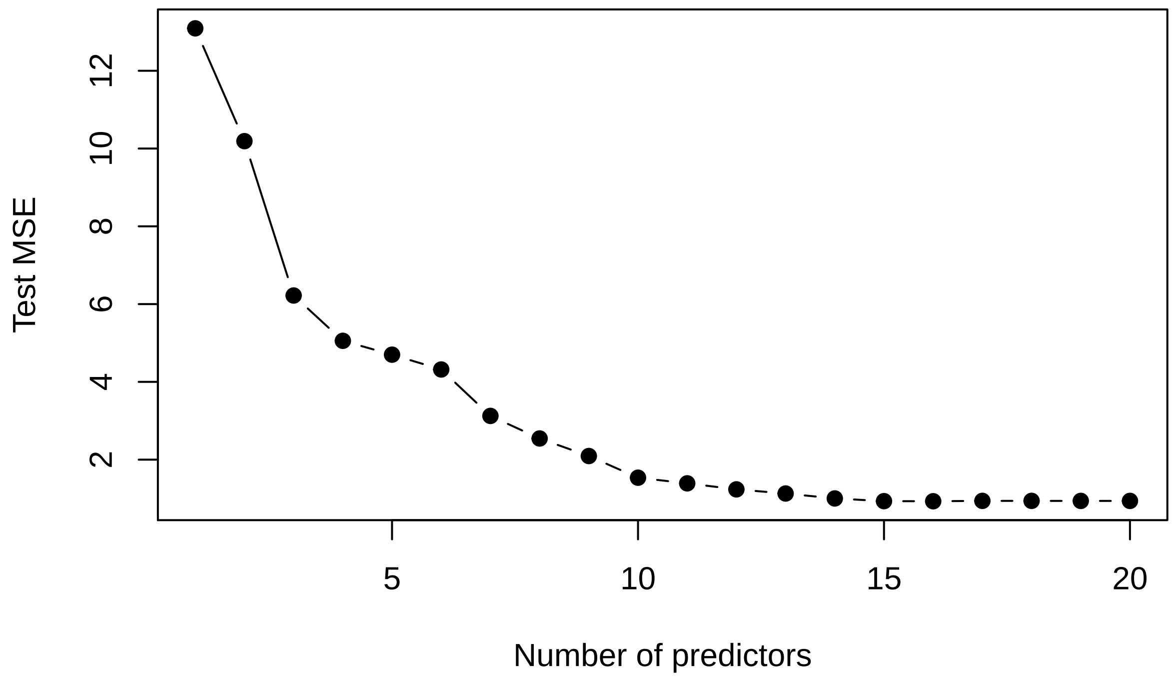
```
regfit <- regsubsets(y ~ ., data = train, nvmax = 20)
train_fit <- model.matrix(y ~ ., data = train, nvmax = 20)
val_errors_train <- rep(NA, 20)
for (i in 1:20) {
    coeff <- coef(regfit, id = i)
```

```
    pred <- train_fit[, names(coeff)] %*% coeff
    val_errors_train[i] <- mean((pred - train$y)^2)
}
plot(val_errors_train, xlab = "Number of predictors", ylab = "Train MSE", pch = 19, type = "b")
```



D

```
test_fit <- model.matrix(y ~ ., data = test, nvmax = 20)
val_errors_test <- rep(NA, 20)
for (i in 1:20) {
    coefi <- coef(regfit, id = i)
    pred <- test_fit[, names(coefi)] %*% coefi
    val_errors_test[i] <- mean((pred - test$y)^2)
}
plot(val_errors_test, xlab = "Number of predictors", ylab = "Test MSE", pch = 19, type = "b")
```



E

```r
which.min(val_errors_test)
```

```
## [1] 16
```

MSE takes its minimal value after 16 predictors.

F

```r
coef(regfit, id =16)
```

```
## (Intercept)          X2          X3          X5          X7          X8
##  0.03279512  0.23344922 -0.66677676  0.99101882 -1.44677632  0.69851493
##          X9         X10         X11         X12         X13         X14
##  2.00337798  0.03044473  0.86107343  0.57564899 -0.26347564 -0.65675222
##         X15         X16         X17         X18         X19
## -0.73034737 -0.29061810  0.32804622  1.63298780  0.90517682
```

This model was able to catch the most zeroed out coefficients with only 3 leaving the range of zero.

G

```
#val_errors_g <- rep(NA, 20)
#x_cols = colnames(x, do.NULL = FALSE)
#for (i in 1:20) {
#    coefi <- coef(regfit, id = i)
#    val_errors_g[i] <- sqrt(sum((beta[x_cols %in% names(coefi)] - coefi[names(coefi) %in% x_cols])^2)
#}
```