Exam-2

Ayush Kumar

4/21/2021

1. We begin our analysis.

a. We have the following result:

```
wbca<-read.table('http://math.uttyler.edu/nathan/data/wbca.data',header=TRUE)
head(wbca)
```

```
##   Class Adhes BNucl Chrom Epith Mitos NNucl Thick UShap USize
## 1     1     1     1     3     2     1     1     5     1     1
## 2     1     5    10     3     7     1     2     5     4     4
## 3     1     1     2     3     2     1     1     3     1     1
## 4     1     1     4     3     3     1     7     6     8     8
## 5     1     3     1     3     2     1     1     4     1     1
## 6     0     8    10     9     7     1     7     8    10    10
```

```
m1<-glm(Class~.,data=wbca,family=binomial())
summary(m1)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial(), data = wbca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4828  -0.0118   0.0474   0.0968   3.0642
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.1668     1.4149    7.89    3e-15 ***
## Adhes        -0.3968     0.1338   -2.96   0.0030 **
## BNucl        -0.4148     0.1023   -4.05    5e-05 ***
## Chrom        -0.5646     0.1873   -3.01   0.0026 **
## Epith        -0.0644     0.1659   -0.39   0.6979
## Mitos        -0.6571     0.3676   -1.79   0.0739 .
## NNucl        -0.2866     0.1262   -2.27   0.0232 *
## Thick        -0.6268     0.1589   -3.94    8e-05 ***
## UShap        -0.2801     0.2523   -1.11   0.2670
## USize         0.0572     0.2327    0.25   0.8059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.464  on 671  degrees of freedom
## AIC: 109.5
##
## Number of Fisher Scoring iterations: 8
```

b. We have the following prediction

```
predict(m1,
        newdata=data.frame(Adhes=1,BNucl=1,Chrom=3,Epith=2,Mitos=1,NNucl=1,Thick=4,
                           UShap=1,USize=1),type="response")
```

```
##      1
## 0.9923
```

c. We present five-fold cross validation to check the error using p>0.5 as our criteria

```
set.seed(13)
the.shuff <- sample(1:681)

gp.1 <- the.shuff[1:136]
gp.2 <- the.shuff[137:272]
gp.3 <- the.shuff[273:408]
gp.4 <- the.shuff[409:544]
gp.5 <- the.shuff[545:681]

round.1.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.1)
round.1.p <- predict(round.1.mod, newdata=wbca[gp.1,-1],type="response")
round.1.mse <- sum((wbca[gp.1,1] - ifelse(round.1.p>0.5,1,0))^2)

round.2.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.2)
round.2.p <- predict(round.2.mod,newdata=wbca[gp.2,-1],type="response")
round.2.mse <- sum((wbca[gp.2,1] - ifelse(round.2.p>0.5,1,0))^2)

round.3.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.3)
round.3.p <- predict(round.3.mod,newdata=wbca[gp.3,-1],type="response")
round.3.mse <- sum((wbca[gp.3,1] - ifelse(round.3.p>0.5,1,0))^2)

round.4.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.4)
round.4.p <- predict(round.4.mod,newdata=wbca[gp.4,-1],type="response")
round.4.mse <- sum((wbca[gp.4,1]- ifelse(round.4.p>0.5,1,0))^2)

round.5.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.5)
round.5.p <- predict(round.5.mod,newdata=wbca[gp.5,-1],type="response")
round.5.mse <- sum((wbca[gp.5,1]- ifelse(round.5.p>0.5,1,0))^2)

cv.mse.1 <- (round.1.mse+round.2.mse+round.3.mse+round.4.mse+round.5.mse)/5

cv.mse.1
```

```
## [1] 4
```

d. We present five-fold cross validation to check the error using p>0.9 as our criteria

```
set.seed(14)
the.shuff <- sample(1:681)

gp.1 <- the.shuff[1:136]
gp.2 <- the.shuff[137:272]
gp.3 <- the.shuff[273:408]
gp.4 <- the.shuff[409:544]
gp.5 <- the.shuff[545:681]

round.1.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.1)
round.1.p <- predict(round.1.mod, newdata=wbca[gp.1,-1],type="response")
round.1.mse <- sum((wbca[gp.1,1] - ifelse(round.1.p>0.9,1,0))^2)

round.2.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.2)
round.2.p <- predict(round.2.mod,newdata=wbca[gp.2,-1],type="response")
round.2.mse <- sum((wbca[gp.2,1] - ifelse(round.2.p>0.9,1,0))^2)

round.3.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.3)
round.3.p <- predict(round.3.mod,newdata=wbca[gp.3,-1],type="response")
round.3.mse <- sum((wbca[gp.3,1] - ifelse(round.3.p>0.9,1,0))^2)

round.4.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.4)
round.4.p <- predict(round.4.mod,newdata=wbca[gp.4,-1],type="response")
round.4.mse <- sum((wbca[gp.4,1]- ifelse(round.4.p>0.9,1,0))^2)

round.5.mod <- glm(Class~.,data=wbca,family=binomial(),subset=-gp.5)
round.5.p <- predict(round.5.mod,newdata=wbca[gp.5,-1],type="response")
round.5.mse <- sum((wbca[gp.5,1]- ifelse(round.5.p>0.9,1,0))^2)

cv.mse.2 <- (round.1.mse+round.2.mse+round.3.mse+round.4.mse+round.5.mse)/5

cv.mse.2
```

```
## [1] 4.2
```

2. We continue the use of same data,

a. We fit a linear discriminant analysis model, we do not scale as the standard deviation appears to be similar for all predictor variables.

```
m2<-lda(Class~.,data=wbca)
```

b. We present the prediction for the same newdata.

```
predict(m2,newdata=data.frame(Adhes=1,BNucl=1,Chrom=3,Epith=2,Mitos=1,
                              NNucl=1,Thick=4,UShap=1,USize=1))$posterior[,2]
```

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

c. We present five-fold cross validation to check the error using p>0.5 as our criteria.

```
set.seed(23)
the.shuff <- sample(1:681)

gp.1 <- the.shuff[1:136]
gp.2 <- the.shuff[137:272]
gp.3 <- the.shuff[273:408]
gp.4 <- the.shuff[409:544]
gp.5 <- the.shuff[545:681]


round.1.mod.ld <- lda(Class~.,data=wbca,subset=-gp.1)
round.1.p.ld <- predict(round.1.mod.ld,newdata=wbca[gp.1,])
round.1.mse.ld <- sum((wbca[gp.1,1] - ifelse(round.1.p.ld$posterior[,2]>0.5,1,0))^2)

round.2.mod.ld <- lda(Class~.,data=wbca,subset=-gp.2)
round.2.p.ld <- predict(round.2.mod.ld,newdata=wbca[gp.2,])
round.2.mse.ld <- sum((wbca[gp.2,1] - ifelse(round.2.p.ld$posterior[,2]>0.5,1,0))^2)

round.3.mod.ld <- lda(Class~.,data=wbca,subset=-gp.3)
round.3.p.ld <- predict(round.3.mod.ld,newdata=wbca[gp.3,])
round.3.mse.ld <- sum((wbca[gp.3,1] - ifelse(round.3.p.ld$posterior[,2]>0.5,1,0))^2)

round.4.mod.ld <- lda(Class~.,data=wbca,subset=-gp.4)
round.4.p.ld <- predict(round.4.mod.ld,newdata=wbca[gp.4,])
round.4.mse.ld <- sum((wbca[gp.4,1] - ifelse(round.4.p.ld$posterior[,2]>0.5,1,0))^2)

round.5.mod.ld <- lda(Class~.,data=wbca,subset=-gp.5)
round.5.p.ld <- predict(round.5.mod.ld,newdata=wbca[gp.5,])
round.5.mse.ld <- sum((wbca[gp.5,1] - ifelse(round.5.p.ld$posterior[,2]>0.5,1,0))^2)


cv.mse <- (round.1.mse.ld+round.2.mse.ld+round.3.mse.ld+round.4.mse.ld+round.5.mse.ld)/5

cv.mse
```

```
## [1] 5.2
```

3. Now we consider the data in college.data. Before we begin analysis of this dataset we first perform some data cleaning.

```
set.seed(3)
col<-read.table('http://math.uttyler.edu/nathan/data/college.data',header=TRUE)
col.use<-col[,-1]
col.use$private<-ifelse(col.use$private=="Yes",1,0)
testers <- sample(1:777,200)
```

a. Next we make the model with all predictors for apps response variable.

```
m3<-lm(apps~.,data=col.use[-testers,])
```

b. We use the step command to identify a smaller linear model, and then estimante the error.

```
step(m3)
```

```
## Start:  AIC=8000
## apps ~ private + accept + enroll + top10perc + top25perc + f.ugrad +
##     p.ugrad + out.st + rm.bd + books + personal + phd + terminal +
##     s.f.ratio + perc.alumni + expend + grad.rate
##
##               Df Sum of Sq      RSS  AIC
## - terminal     1  5.72e+04 5.69e+08 7998
## - books        1  1.59e+05 5.69e+08 7998
## - personal     1  6.37e+05 5.70e+08 7998
## - perc.alumni  1  1.51e+06 5.71e+08 7999
## <none>                    5.69e+08 8000
## - private      1  2.02e+06 5.71e+08 8000
## - p.ugrad      1  2.97e+06 5.72e+08 8001
## - phd          1  3.22e+06 5.72e+08 8001
## - grad.rate    1  3.27e+06 5.73e+08 8001
## - s.f.ratio    1  4.33e+06 5.74e+08 8002
## - top25perc    1  4.55e+06 5.74e+08 8002
## - f.ugrad      1  7.87e+06 5.77e+08 8006
## - rm.bd        1  8.70e+06 5.78e+08 8007
## - out.st       1  2.16e+07 5.91e+08 8019
## - enroll       1  3.15e+07 6.01e+08 8029
## - expend       1  4.08e+07 6.10e+08 8038
## - top10perc    1  4.82e+07 6.17e+08 8045
## - accept       1  1.37e+09 1.94e+09 8704
##
## Step:  AIC=7998
## apps ~ private + accept + enroll + top10perc + top25perc + f.ugrad +
##     p.ugrad + out.st + rm.bd + books + personal + phd + s.f.ratio +
##     perc.alumni + expend + grad.rate
##
##               Df Sum of Sq      RSS  AIC
## - books        1  1.68e+05 5.69e+08 7996
## - personal     1  6.43e+05 5.70e+08 7996
## - perc.alumni  1  1.47e+06 5.71e+08 7997
## - private      1  1.97e+06 5.71e+08 7998
## <none>                    5.69e+08 7998
## - p.ugrad      1  2.97e+06 5.72e+08 7999
## - grad.rate    1  3.35e+06 5.73e+08 7999
## - s.f.ratio    1  4.33e+06 5.74e+08 8000
## - top25perc    1  4.85e+06 5.74e+08 8001
## - f.ugrad      1  7.83e+06 5.77e+08 8004
## - rm.bd        1  8.66e+06 5.78e+08 8005
## - phd          1  8.83e+06 5.78e+08 8005
## - out.st       1  2.21e+07 5.91e+08 8018
## - enroll       1  3.15e+07 6.01e+08 8027
## - expend       1  4.07e+07 6.10e+08 8036
## - top10perc    1  4.95e+07 6.19e+08 8044
## - accept       1  1.37e+09 1.94e+09 8703
##
## Step:  AIC=7996
```

```
## apps ~ private + accept + enroll + top10perc + top25perc + f.ugrad +
##     p.ugrad + out.st + rm.bd + personal + phd + s.f.ratio + perc.alumni +
##     expend + grad.rate
##
##               Df Sum of Sq      RSS  AIC
## - personal     1  5.30e+05 5.70e+08 7995
## - perc.alumni  1  1.46e+06 5.71e+08 7995
## <none>                     5.69e+08 7996
## - private      1  1.99e+06 5.71e+08 7996
## - p.ugrad      1  2.92e+06 5.72e+08 7997
## - grad.rate    1  3.40e+06 5.73e+08 7997
## - s.f.ratio    1  4.24e+06 5.74e+08 7998
## - top25perc    1  4.94e+06 5.74e+08 7999
## - f.ugrad      1  7.88e+06 5.77e+08 8002
## - rm.bd        1  8.52e+06 5.78e+08 8003
## - phd          1  8.92e+06 5.78e+08 8003
## - out.st       1  2.20e+07 5.91e+08 8016
## - enroll       1  3.15e+07 6.01e+08 8025
## - expend       1  4.07e+07 6.10e+08 8034
## - top10perc    1  4.93e+07 6.19e+08 8042
## - accept       1  1.37e+09 1.94e+09 8701
##
## Step:  AIC=7995
## apps ~ private + accept + enroll + top10perc + top25perc + f.ugrad +
##     p.ugrad + out.st + rm.bd + phd + s.f.ratio + perc.alumni +
##     expend + grad.rate
##
##               Df Sum of Sq      RSS  AIC
## - perc.alumni  1  1.35e+06 5.71e+08 7994
## - private      1  1.95e+06 5.72e+08 7994
## <none>                     5.70e+08 7995
## - grad.rate    1  3.10e+06 5.73e+08 7996
## - p.ugrad      1  3.20e+06 5.73e+08 7996
## - s.f.ratio    1  3.97e+06 5.74e+08 7997
## - top25perc    1  5.00e+06 5.75e+08 7998
## - rm.bd        1  8.41e+06 5.78e+08 8001
## - f.ugrad      1  8.55e+06 5.79e+08 8001
## - phd          1  9.02e+06 5.79e+08 8002
## - out.st       1  2.31e+07 5.93e+08 8015
## - enroll       1  3.19e+07 6.02e+08 8024
## - expend       1  4.13e+07 6.11e+08 8033
## - top10perc    1  5.00e+07 6.20e+08 8041
## - accept       1  1.37e+09 1.94e+09 8699
##
## Step:  AIC=7994
## apps ~ private + accept + enroll + top10perc + top25perc + f.ugrad +
##     p.ugrad + out.st + rm.bd + phd + s.f.ratio + expend + grad.rate
##
##             Df Sum of Sq      RSS  AIC
## - private    1  1.88e+06 5.73e+08 7994
## <none>                   5.71e+08 7994
## - p.ugrad    1  3.12e+06 5.74e+08 7995
## - s.f.ratio  1  3.64e+06 5.75e+08 7996
## - grad.rate  1  4.37e+06 5.76e+08 7996
```

```
## - top25perc  1  4.59e+06 5.76e+08 7996
## - rm.bd      1  7.52e+06 5.79e+08 7999
## - f.ugrad    1  8.08e+06 5.79e+08 8000
## - phd        1  8.77e+06 5.80e+08 8001
## - out.st     1  2.18e+07 5.93e+08 8013
## - enroll     1  3.09e+07 6.02e+08 8022
## - expend     1  4.18e+07 6.13e+08 8033
## - top10perc  1  5.04e+07 6.22e+08 8041
## - accept     1  1.38e+09 1.95e+09 8701
##
## Step:  AIC=7994
## apps ~ accept + enroll + top10perc + top25perc + f.ugrad + p.ugrad +
##     out.st + rm.bd + phd + s.f.ratio + expend + grad.rate
##
##             Df Sum of Sq      RSS  AIC
## <none>                   5.73e+08 7994
## - p.ugrad    1  3.31e+06 5.77e+08 7995
## - grad.rate  1  4.02e+06 5.77e+08 7996
## - top25perc  1  4.60e+06 5.78e+08 7996
## - s.f.ratio  1  4.65e+06 5.78e+08 7996
## - phd        1  7.12e+06 5.80e+08 7999
## - rm.bd      1  7.14e+06 5.80e+08 7999
## - f.ugrad    1  1.02e+07 5.83e+08 8002
## - enroll     1  3.19e+07 6.05e+08 8023
## - out.st     1  3.24e+07 6.06e+08 8024
## - expend     1  4.43e+07 6.17e+08 8035
## - top10perc  1  4.97e+07 6.23e+08 8040
## - accept     1  1.39e+09 1.96e+09 8701
```

```
##
## Call:
## lm(formula = apps ~ accept + enroll + top10perc + top25perc +
##     f.ugrad + p.ugrad + out.st + rm.bd + phd + s.f.ratio + expend +
##     grad.rate, data = col.use[-testers, ])
##
## Coefficients:
## (Intercept)       accept       enroll    top10perc    top25perc      f.ugrad
##   -971.1931       1.6340      -1.1906      41.9929     -10.3192       0.1135
##     p.ugrad       out.st        rm.bd          phd    s.f.ratio       expend
##      0.0625      -0.1066       0.1379      -8.7252      30.5725       0.0862
##   grad.rate
##      6.2975
```

```
m.use<-lm(formula = apps ~ accept + enroll + top10perc + top25perc +
            f.ugrad + p.ugrad + out.st + rm.bd + phd + s.f.ratio + expend +
            grad.rate, data = col.use[-testers, ])
mse.1<-sqrt(sum((predict(m.use,newdata=col.use[testers,])-col.use[testers,2])^2)/200)
mse.1
```

```
## [1] 1172
```

Before moving on to the next steps we perform some more data manipulation.

```
build.data.t <-scale(col.use[-testers,-c(1,2)],center=TRUE,scale=TRUE)
build.data<-cbind(col.use[-testers,c(1,2)],build.data.t)
apply(build.data,2,mean)
```

```
##     private        apps      accept      enroll   top10perc   top25perc
##   7.400e-01   2.889e+03  -1.173e-17   5.631e-17  -1.197e-17  -1.362e-16
##     f.ugrad     p.ugrad      out.st       rm.bd       books    personal
##   2.283e-17   9.436e-18   5.270e-17  -4.454e-17  -1.035e-16  -7.766e-17
##         phd    terminal   s.f.ratio perc.alumni      expend   grad.rate
##   2.874e-16   3.685e-16   5.423e-17  -1.210e-16  -5.116e-17  -7.014e-19
```

```
apply(build.data,2,sd)
```

```
##     private        apps      accept      enroll   top10perc   top25perc
##       0.439    3956.548       1.000       1.000       1.000       1.000
##     f.ugrad     p.ugrad      out.st       rm.bd       books    personal
##       1.000       1.000       1.000       1.000       1.000       1.000
##         phd    terminal   s.f.ratio perc.alumni      expend   grad.rate
##       1.000       1.000       1.000       1.000       1.000       1.000
```

```
test.data.t <- sweep(col.use[testers,-c(1,2)],2,attr(build.data.t,"scaled:center"))
test.data.t.2 <- sweep(test.data.t,2,attr(build.data.t,"scaled:scale"),FUN='/')
test.data<-cbind(col.use[testers,c(1,2)],test.data.t.2)
```

c. We fit a ridge regression model with λ chosen by cross validation and then report the test error obtained. We decided to incorporate the private variable and found that the predictions were worse, and hence we dropped it. To view the actual compuation consider the R file attached.

```
m.ridge <- glmnet(build.data[,-c(1,2)],build.data[,2],alpha=0,standardize=FALSE
                  ,lambda=seq(.001,1,length=100))

m.ridg1 <- glmnet(build.data[,-c(1,2)],build.data[,2],standardize=FALSE
                  ,alpha=0,lambda=cv.glmnet(as.matrix(build.data[,-c(1,2)])
                                            ,build.data[,2],alpha=0,folds=5)$lambda.min)

ridg.hat <- as.matrix(test.data[,-c(1,2)]) %*% m.ridg1$beta

mse.ridge<-sqrt(sum((test.data[,2]-ridg.hat)^2)/200)

mse.ridge
```

```
## [1] 3155
```

d. We fit a lasso regression model with λ chosen by cross validation and then report the test error obtained. Additionally we compare the zero coefficients in this model to the one obtained by step command. When we made a model with the private variable incorporated it was set to zero, for the actual code consider the R file.

```
m.lass0 <- glmnet(build.data[,-c(1,2)],build.data[,2],alpha=1,standardize=FALSE,
                  lambda=seq(.001,1,length=100))

m.lass1 <- glmnet(build.data[,-c(1,2)],build.data[,2],standardize=FALSE,
                  alpha=1,lambda=cv.glmnet(as.matrix(build.data[,-c(1,2)]),
                                           build.data[,2],alpha=1,folds=5)$lambda.min)

las.hat.1 <- as.matrix(test.data[,-c(1,2)]) %*% m.lass1$beta

mse.lasso<-sqrt(sum((test.data[,2]-las.hat.1)^2)/200)

mse.lasso
```

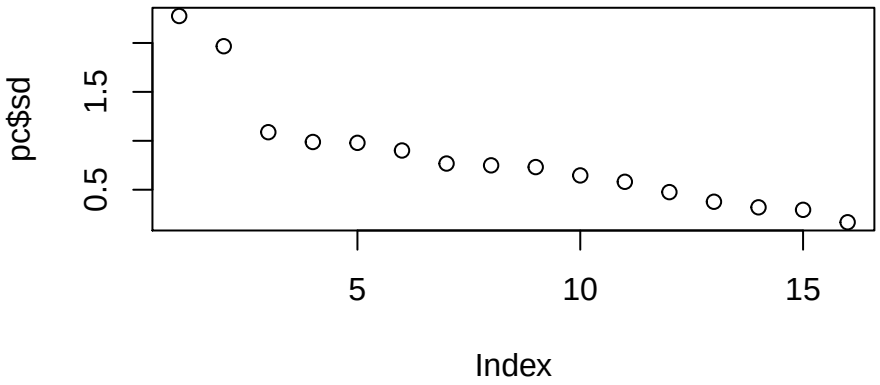
```
## [1] 3160
```

```
m.lass1$beta
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                  s0
## accept      3734.16
## enroll      -183.75
## top10perc    484.90
## top25perc      .
## f.ugrad        .
## p.ugrad       51.55
## out.st      -292.10
## rm.bd        124.12
## books          .
## personal      17.40
## phd          -85.95
## terminal      -2.43
## s.f.ratio     76.77
## perc.alumni    .
## expend       388.16
## grad.rate     43.73
```

We note that the variables dropped by step function and lasso are different, in particular we note that while predictors top25perc and f.ugrad were dropped by lasso and not by step, step dropped personal and terminal which was non-zero in lasso model. e. Now we perform principle component analysis. Here notice that the private variable was dropped again, as the model was worse.

```
pc<-prcomp(build.data.t)
plot(pc$sd)
```




```
summary(pc)
```

```
## Importance of components:
##                           PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
## Standard deviation     2.275 1.967 1.088 0.988 0.9791 0.9009 0.7676 0.7490
## Proportion of Variance 0.324 0.242 0.074 0.061 0.0599 0.0507 0.0368 0.0351
## Cumulative Proportion  0.324 0.565 0.639 0.700 0.7601 0.8109 0.8477 0.8828
##                           PC9   PC10   PC11   PC12    PC13    PC14    PC15
## Standard deviation     0.7315 0.6461 0.5809 0.4756 0.37710 0.32013 0.29465
## Proportion of Variance 0.0334 0.0261 0.0211 0.0141 0.00889 0.00641 0.00543
## Cumulative Proportion  0.9162 0.9423 0.9634 0.9775 0.98641 0.99282 0.99824
##                           PC16
## Standard deviation     0.16761
## Proportion of Variance 0.00176
## Cumulative Proportion  1.00000
```

```
m.pc<-lm(build.data$apps ~ pc$x[,1:11])
m.pc.1 <- lm(build.data$apps ~ pc$x[,1:11]+build.data$private)
anova(m.pc,m.pc.1) #Notice build.data$private is not significant
```

```
## Analysis of Variance Table
##
## Model 1: build.data$apps ~ pc$x[, 1:11]
## Model 2: build.data$apps ~ pc$x[, 1:11] + build.data$private
##   Res.Df      RSS Df Sum of Sq    F Pr(>F)
## 1    565 1.24e+09
## 2    564 1.24e+09  1     2e+06 0.91   0.34
```

```
summary(m.pc)
```

```
##
## Call:
## lm(formula = build.data$apps ~ pc$x[, 1:11])
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6810   -516    -78    408  22592
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2889.3       61.6   46.91  < 2e-16 ***
## pc$x[, 1:11]PC1     648.6       27.1   23.94  < 2e-16 ***
## pc$x[, 1:11]PC2   -1555.7       31.4  -49.62  < 2e-16 ***
## pc$x[, 1:11]PC3     539.0       56.7    9.51  < 2e-16 ***
## pc$x[, 1:11]PC4   -1045.1       62.4  -16.74  < 2e-16 ***
## pc$x[, 1:11]PC5    -150.5       63.0   -2.39  0.01719 *
## pc$x[, 1:11]PC6     445.4       68.4    6.51  1.7e-10 ***
## pc$x[, 1:11]PC7    -484.6       80.3   -6.03  2.9e-09 ***
## pc$x[, 1:11]PC8      11.2       82.3    0.14  0.89229
## pc$x[, 1:11]PC9    -579.8       84.3   -6.88  1.6e-11 ***
## pc$x[, 1:11]PC10    368.1       95.4    3.86  0.00013 ***
## pc$x[, 1:11]PC11    -79.6      106.1   -0.75  0.45335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1480 on 565 degrees of freedom
## Multiple R-squared:  0.863,  Adjusted R-squared:  0.86
## F-statistic:  323 on 11 and 565 DF,  p-value: <2e-16
```

```
test.3 <- as.matrix(test.data[,-c(2,1)]) %*% pc$rot[,1:11]
test.4 <- cbind(rep(1,200),test.3)

test.yhat <- test.4 %*% m.pc$coeff
test.y <- col.use$apps[testers]

mse.pc<-sqrt(sum((as.data.frame(test.y) - as.data.frame(test.yhat))^2 )/200)
mse.pc
```

```
## [1] 1471
```

Now comparing the models we have thus, the step models is the best predictor in general.

```
mse.1
```

```
## [1] 1172
```

```
mse.ridge
```

```
## [1] 3155
```

```
mse.lasso
```

```
## [1] 3160
```

```
mse.pc
```

```
## [1] 1471
```