Assignment #5: Advanced Regression

SANDEEP DASARI **UIN - 829002252**

Problem 1

In this question, we will predict the number of applications received (Apps) using the other variables in the College data set (ISLR package).

(a) Perform best subset selection to the data. What is the best model obtained according to C_p, BIC and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model.

```
> library(ISLR)
>attach(College)
>library(glmnet)
>library(leaps)

> regfit.full = regsubsets(Apps ~.,data = College, nvmax = 17)  #to return as many predictors as specified
> reg.summary = summary(regfit.full)
> par(mfrow = c(2,2))
> plot(reg.summary$rss, xlab = "No. of Predictors", ylab = "RSS")
> plot(reg.summary$adjr2, xlab = "No. of Predictors", ylab = "Adjusted Rsq")
> which.max(reg.summary$adjr2)
[1] 13
> points(13,reg.summary$adjr2[13], col="red",cex=2,pch=20)
> plot(reg.summary$cp, xlab = "No. of Predictors", ylab = "Cp")
> which.min(reg.summary$cp)
[1] 12
> points(12,reg.summary$cp[12],col="red",cex=2,pch=20)
> plot(reg.summary$bic,xlab="No. of predictors",ylab="BIC")
> which.min(reg.summary$bic)
[1] 10
> points(10,reg.summary$bic[10],col="red",cex=2,pch=20)
```

- From the data the best model obtained for C_p has 12 predictors, BIC gave 10 and adjusted R^2 gave 13 predictors.

- The coefficients for the best models according to to C_p, BIC and adjusted R^2 are given below:

Adjusted Rsquare

```
> coef(regfit.full,13)
```

(Intercept) PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board PhD

0.09108187 -0.12525772 1.00409566 -0.21085268 0.22978966 -0.07489947 0.07222326 0.01826143 -0.09171700 0.04164454 -0.04606261

S.F.Ratio Expend Grad.Rate

0.01549991 0.10505631 0.03810762

CP

> coef(regfit.full,12) #Cp

(Intercept) PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board PhD

0.09615760 -0.13223797 1.00504039 -0.21191169 0.22977460 -0.07546585 0.07451314 0.01806788 -0.09373705 0.04187231 -0.04516389

Expend Grad.Rate

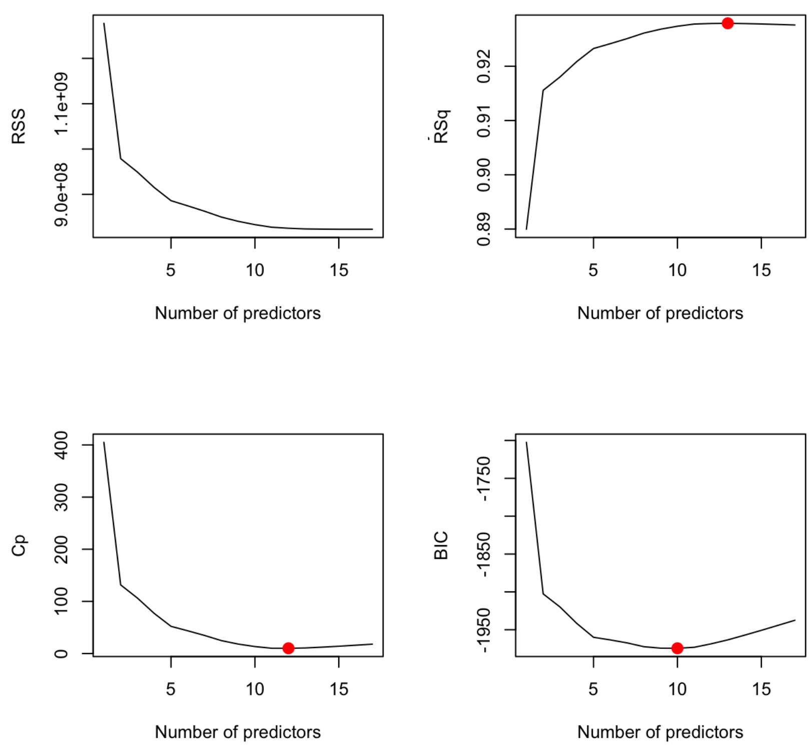
0.09777360 0.03834651

BIC

> coef(regfit.full,10)

(Intercept) PrivateYes Accept Enroll Top10perc Top25perc Outstate Room.Board PhD Expend Grad.Rate

0.10804758 -0.14858932 1.00333304 -0.13497766 0.22397579 -0.07095222 -0.09840240 0.04639797 -0.04225729 0.09813952 0.03254580



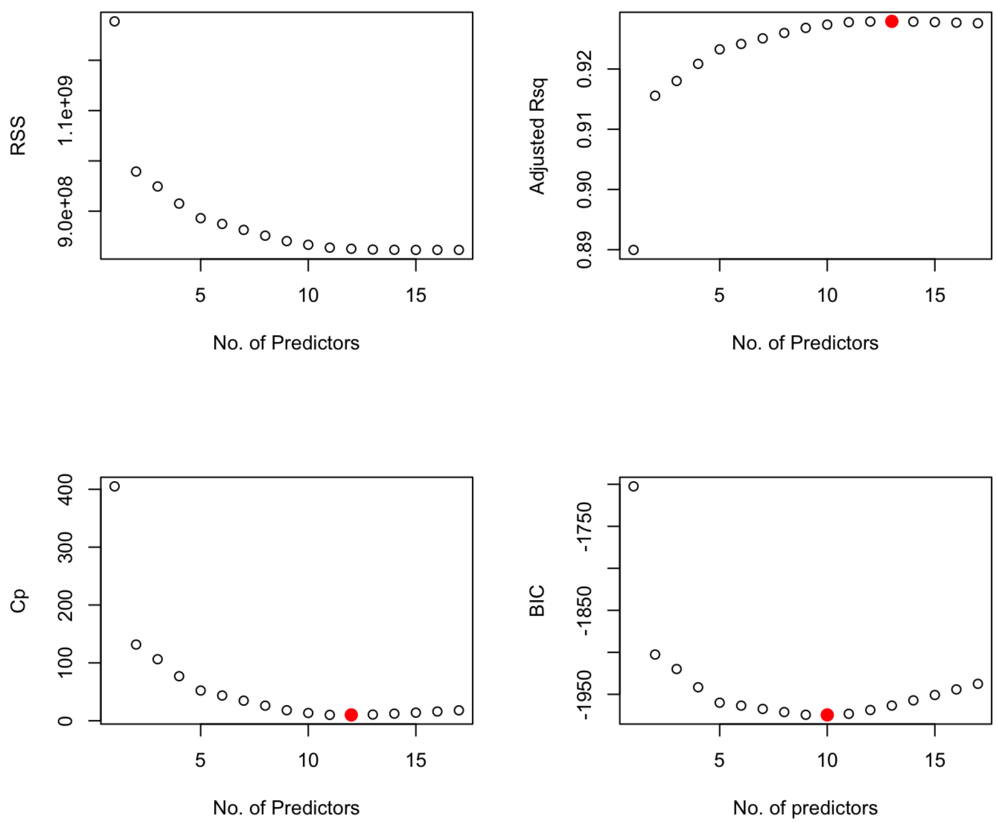
(b) Repeat (a) using forward stepwise selection and backwards stepwise selection. How does your answer compare to the results in (a)?

Forward Selection

```
> regfit.fwd=regsubsets(Apps~.,data=College,nvmax=17, method="forward")
> fwd = summary(regfit.fwd)
> par(mfrow = c(2,2))
> plot(fwd$rss, xlab = "No. of Predictors", ylab = "RSS")
> plot(fwd$adjr2, xlab = "No. of Predictors", ylab = "Adjusted Rsq")
> which.max(fwd$adjr2)
[1] 13
>  points(13,fwd$adjr2[13], col="red",cex=2,pch=20)
> plot(fwd$cp, xlab = "No. of Predictors", ylab = "Cp")
> which.min(fwd$cp)
[1] 12
> points(12,fwd$cp[12],col="red",cex=2,pch=20)
> plot(fwd$bic,xlab="No. of predictors",ylab="BIC")
> which.min(fwd$bic)
[1] 10
> points(10,fwd$bic[10],col="red",cex=2,pch=20)
> coef(regfit.fwd,13)
(Intercept)   PrivateYes        Accept        Enroll     Top10perc     Top25perc   F.Undergrad
P.Undergrad   Outstate  Room.Board        PhD
 0.09108187 -0.12525772  1.00409566 -0.21085268  0.22978966 -0.07489947  0.07222326  0.01826143 -
0.09171700  0.04164454 -0.04606261
 S.F.Ratio     Expend   Grad.Rate
 0.01549991  0.10505631  0.03810762

> coef(regfit.fwd,12)
(Intercept)   PrivateYes        Accept        Enroll     Top10perc     Top25perc   F.Undergrad
P.Undergrad   Outstate  Room.Board        PhD
 0.09615760 -0.13223797  1.00504039 -0.21191169  0.22977460 -0.07546585  0.07451314  0.01806788 -
0.09373705  0.04187231 -0.04516389
    Expend   Grad.Rate
 0.09777360  0.03834651

> coef(regfit.fwd,10)
(Intercept)  PrivateYes     Accept     Enroll   Top10perc   Top25perc    Outstate  Room.Board        PhD
 Expend   Grad.Rate
 0.10804758 -0.14858932  1.00333304 -0.13497766  0.22397579 -0.07095222 -0.09840240  0.04639797 -
0.04225729  0.09813952  0.03254580
```



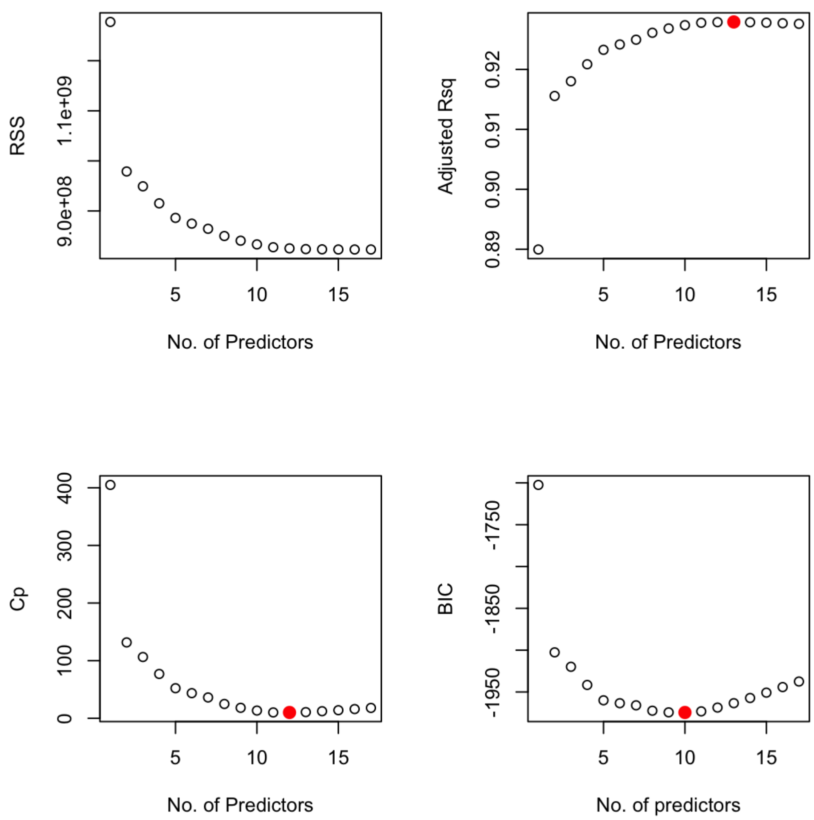
<u>Backward Selection</u>

```
> regfit.back=regsubsets(Apps~.,data=College,nvmax=17, method="backward")
> back = summary(regfit.back)
> par(mfrow = c(2,2))
> plot(back$rss, xlab = "No. of Predictors", ylab = "RSS")
> plot(back$adjr2, xlab = "No. of Predictors", ylab = "Adjusted Rsq")
> which.max(back$adjr2)
[1] 13
> points(13,back$adjr2[13], col="red",cex=2,pch=20)
> plot(back$cp, xlab = "No. of Predictors", ylab = "Cp")
> which.min(back$cp)
[1] 12
> points(12,back$cp[12],col="red",cex=2,pch=20)
> plot(back$bic,xlab="No. of predictors",ylab="BIC")
> which.min(back$bic)
[1] 10
> points(10,back$bic[10],col="red",cex=2,pch=20)
```

```
> coef(regfit.bwd,13)
(Intercept)     PrivateYes          Accept          Enroll       Top10perc       Top25perc    F.Undergrad
P.Undergrad    Outstate  Room.Board        PhD
 0.09108187 -0.12525772  1.00409566 -0.21085268  0.22978966 -0.07489947  0.07222326  0.01826143 -
0.09171700  0.04164454 -0.04606261
  S.F.Ratio      Expend   Grad.Rate
 0.01549991  0.10505631  0.03810762
```

```
> coef(regfit.bwd,12)
(Intercept)     PrivateYes          Accept          Enroll       Top10perc       Top25perc    F.Undergrad
P.Undergrad    Outstate  Room.Board        PhD
 0.09615760 -0.13223797  1.00504039 -0.21191169  0.22977460 -0.07546585  0.07451314  0.01806788 -
0.09373705  0.04187231 -0.04516389
     Expend   Grad.Rate
 0.09777360  0.03834651
```

```
> coef(regfit.bwd,10)
(Intercept)  PrivateYes      Accept      Enroll   Top10perc   Top25perc    Outstate  Room.Board         PhD
  Expend   Grad.Rate
 0.10804758 -0.14858932  1.00333304 -0.13497766  0.22397579 -0.07095222 -0.09840240  0.04639797 -
0.04225729  0.09813952  0.03254580
```



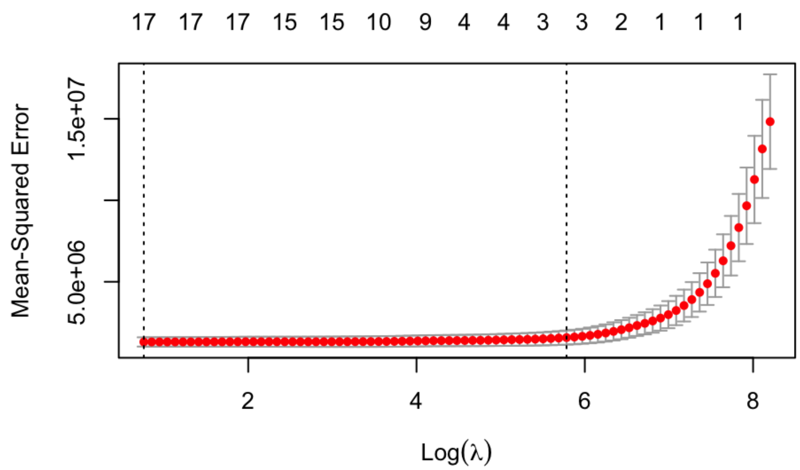
- After solving for forward and backward selection we can see that all the three results are same.
- With C_p having 12 predictors, BIC 10 and adjusted R^2 13 predictors.

(c) Fit a lasso model on the data. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates.

```
> x=model.matrix(Apps~.,College)[,-1]
> y=College$Apps
> grid=10^seq(10,-2,length=100)
> lasso.mod=glmnet(x,y,alpha=1,lambda=grid)
> dim(coef(lasso.mod))
[1]  18 100
> set.seed(1)
> cv.out=cv.glmnet(x,y,alpha=1)
> plot(cv.out)
> bestlam=cv.out$lambda.min
> bestlam
[1] 2.137223
> cv.out
Call:  cv.glmnet(x = x, y = y, alpha = 1)
Measure: Mean-Squared Error
    Lambda Index Measure     SE Nonzero
min    2.1    81 1306653 284264      17
1se  324.8    27 1574599 434651       3
```

- The coefficients are reported below:

```
> coef(lasso.mod)[,81]
  (Intercept)    PrivateYes        Accept        Enroll     Top10perc     Top25perc   F.Undergrad
-469.24652351 -491.43250918    1.57096364   -0.76691866   48.23081240  -12.92684210    0.04271608
  P.Undergrad      Outstate    Room.Board         Books      Personal           PhD      Terminal
   0.04406319   -0.08336740    0.14960048    0.01540876    0.02911802   -8.42093118   -3.26538539
    S.F.Ratio   perc.alumni        Expend     Grad.Rate
  14.59619823   -0.02813289    0.07716952    8.30956380
```



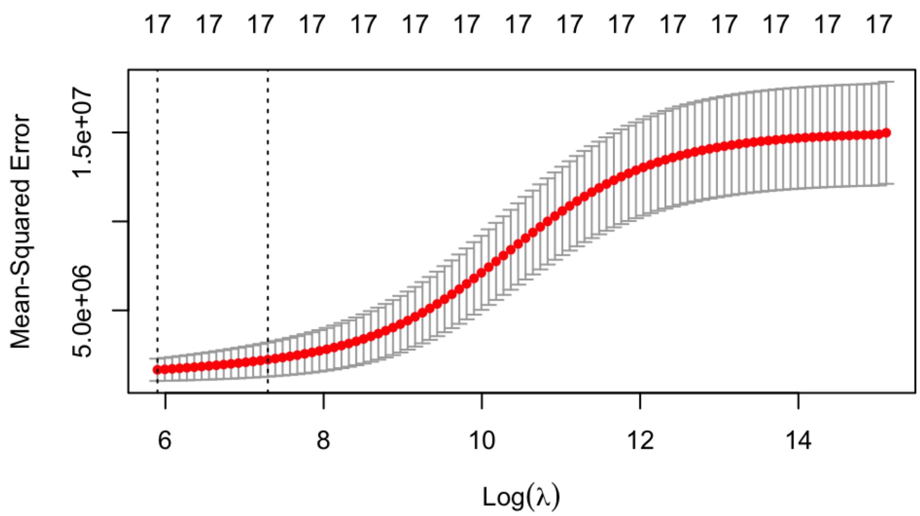
(d) Fit a ridge regression model on the data. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates.

```
> x=model.matrix(Apps~.,College)[,-1]
> y=College$Apps
> grid=10^seq(10,-2,length=100)
> ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
> dim(coef(ridge.mod))
[1]  18 100
> set.seed(1)
> cv.out=cv.glmnet(x,y,alpha=0)
> plot(cv.out)
> bestlam=cv.out$lambda.min
> bestlam
[1] 364.8993
> cv.out
Call:  cv.glmnet(x = x, y = y, alpha = 0)
Measure: Mean-Squared Error
    Lambda Index Measure     SE Nonzero
min  364.9   100 1658142 625944      17
1se 1473.1    85 2233273 957497      17
```

- <u>The coefficients are reported below:</u>

```
> coef(ridge.mod)[,100]
```

```
(Intercept)   PrivateYes      Accept      Enroll    Top10perc     Top25perc   F.Undergrad
-446.11999448  -494.19910151      1.58524428      -0.87781541      49.90206822     -14.21860143
0.05710387
 P.Undergrad      Outstate    Room.Board       Books      Personal          PhD      Terminal
  0.04445340  -0.08584251   0.15112815   0.02092752   0.03106553  -8.67676502  -3.33139275
   S.F.Ratio   perc.alumni       Expend     Grad.Rate
 15.39244418  0.16868856      0.07790702  8.66969506
```



(e) Now split the data set into a training set and a test set.

i. Fit the best models obtained in the best subset selection (according to C_p, BIC or adjusted R^2) to the training set, and report the test error obtained.

```
> College[, -1] = apply(College[, -1], 2, scale)
> train.size =  dim(College)[1] / 2
> train =  sample(1:dim(College)[1], train.size)
> test =  -train
> College.train =  College[train, ]
> College.test =  College[test, ]
>  set.seed(1)
>  train=sample(c(TRUE,FALSE), nrow(College), rep=TRUE)
>  test=(!train)
> regfit.best=regsubsets(Apps~.,data=College[train,],nvmax=17)
> test.mat=model.matrix(Apps~.,data=College[test,])
> val.errors=rep(NA,17)
> for(i in 1:17){
+  coefi=coef(regfit.best,id=i)
```

```
+ pred=test.mat[,names(coefi)]%*%coefi
+ val.errors[i]=mean((College$Apps[test]-pred)^2)
+  }
>  val.errors
 [1] 0.09484810 0.07290081 0.07481260 0.07342112 0.06616749 0.06711697
 [7] 0.06776058 0.06596368 0.06613753 0.06470319 0.06643225 0.06650641
[13] 0.06584790 0.06603828 0.06558849 0.06572693 0.06574395
> which.min(val.errors)
[1] 10
> min(val.errors)
[1] 0.06470319
> coef(regfit.best,10)
(Intercept)  PrivateYes      Accept      Enroll   Top10perc   Top25perc
 0.13274780 -0.17746317  1.06253302 -0.16191898  0.26765101 -0.10992429
   Outstate  Room.Board         PhD      Expend   Grad.Rate
-0.08470742  0.03145215 -0.05490531  0.10461991  0.04514569
```

- The test error (MSE) is 0.06470319.

  ii. Fit a lasso model to the training set, with λ chosen by cross validation. Report the test error obtained.

```
> train.mat =  model.matrix(Apps ~ . -1 , data = College.train)
>  test.mat =  model.matrix(Apps ~ . -1, data = College.test)
>  grid =  10 ^ seq(4, -2, length = 100)
>  mod.lasso =  cv.glmnet(train.mat, College.train[, "Apps"], alpha = 1, lambda = grid, thresh = 1e-12)
>  lambda1 =  mod.lasso$lambda.min
>  lambda1
[1] 0.01
>  lasso.pred =  predict(mod.lasso, newx = test.mat, s = lambda1
>  mean((College.test[, "Apps"] - lasso.pred)^2)
[1] 0.06568935
>  mod.lasso =  glmnet(model.matrix(Apps ~ . -1, data = College),College[, "Apps"], alpha = 1)
>  predict(mod.lasso, s = lambda1 type = "coefficients")
19 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept) -2.483323e-02
PrivateNo    9.101612e-02
```

```
PrivateYes  -1.017338e-13
Accept       8.827830e-01
Enroll       .
Top10perc    1.285778e-01
Top25perc    .
F.Undergrad  .
P.Undergrad  .
Outstate    -3.693941e-02
Room.Board   2.682937e-02
Books        .
Personal     .
PhD         -1.307949e-02
Terminal    -1.016626e-02
S.F.Ratio    .
perc.alumni -1.794075e-03
Expend       8.228831e-02
Grad.Rate    1.271356e-02
```

- The test error (MSE) is 0.065 and the lambda is 0.01.

iii. Fit a ridge regression model to the training set, with λ chosen by cross validation. Report the test error obtained.

```
>  mod.ridge =  cv.glmnet(train.mat, College.train[, "Apps"], alpha = 0, lambda = grid, thresh = 1e-12)
>  lambda2 =  mod.ridge$lambda.min
>  lambda2
[1] 0.01
>  ridge.pred =  predict(mod.ridge, newx = test.mat, s = lambda2)
>  mean((College.test[, "Apps"] - ridge.pred)^2)
[1] 0.645922
>  mod.ridge =  glmnet(model.matrix(Apps ~ . -1, data = College),College[, "Apps"], alpha = 0)
>  predict(mod.ridge, s = lambda2, type = "coefficients")
19 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)  0.034643113
PrivateNo    0.074968069
PrivateYes  -0.075771556
```

Accept	0.635847145
Enroll	0.102959802
Top10perc	0.117721348
Top25perc	0.002865456
F.Undergrad	0.089667745
P.Undergrad	0.009164289
Outstate	-0.022853483
Room.Board	0.056918260
Books	0.005610032
Personal	-0.001512521
PhD	-0.017809249
Terminal	-0.019072607
S.F.Ratio	0.012463665
perc.alumni	-0.026810858
Expend	0.102090055
Grad.Rate	0.050408375

- The test error (MSE) is 0.06459 and the lambda is 0.01.

iv. Compare the test errors obtained in the above analysis (i-iii) and determine the optimal model.

- From the above models we can see that,

Test errors for best subset, lasso and ridge regression are 0.0647, 0.0656 and 0.0645 respectively.

- Actually all the test errors are very similar so any one of the model can be selected, in order to make a decision ridge regression can be selected.