EAS596, Homework_2

Abhishek Kumar, Class#1

9/29/2018

SOLUTION 1

The college data has 777 observations and the data is clean. Data has been divided into training and test data-set in the ratio of 3:1, with training set having 583 observations and test set having 194 observations.

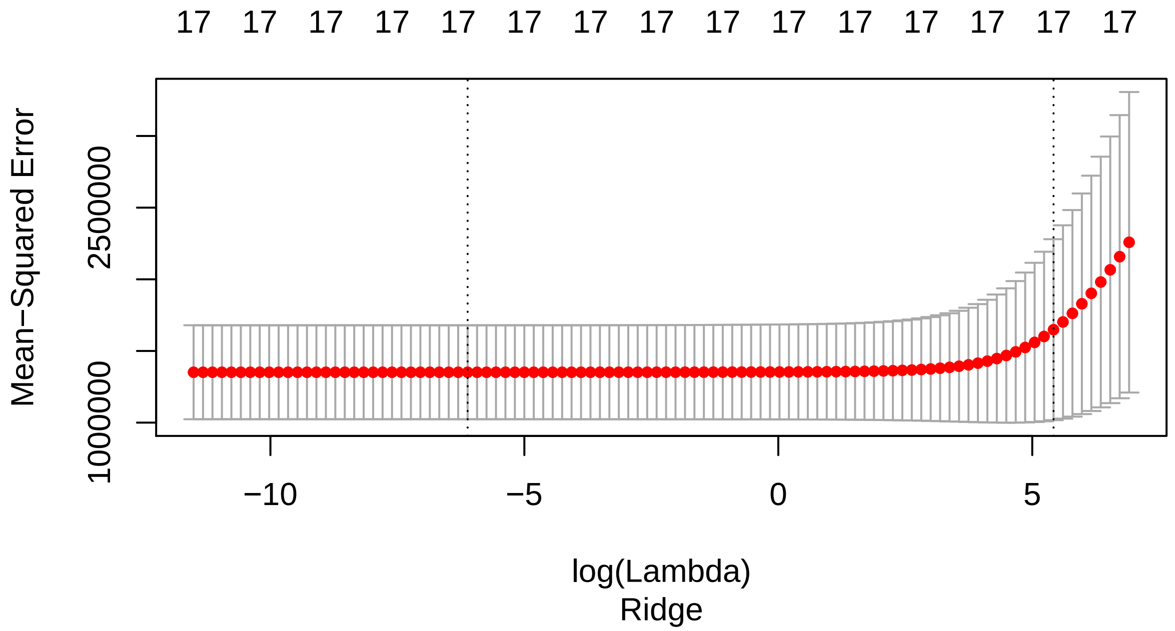
a) Using linear model we get an RMSE of 1044.6566 on the test set.

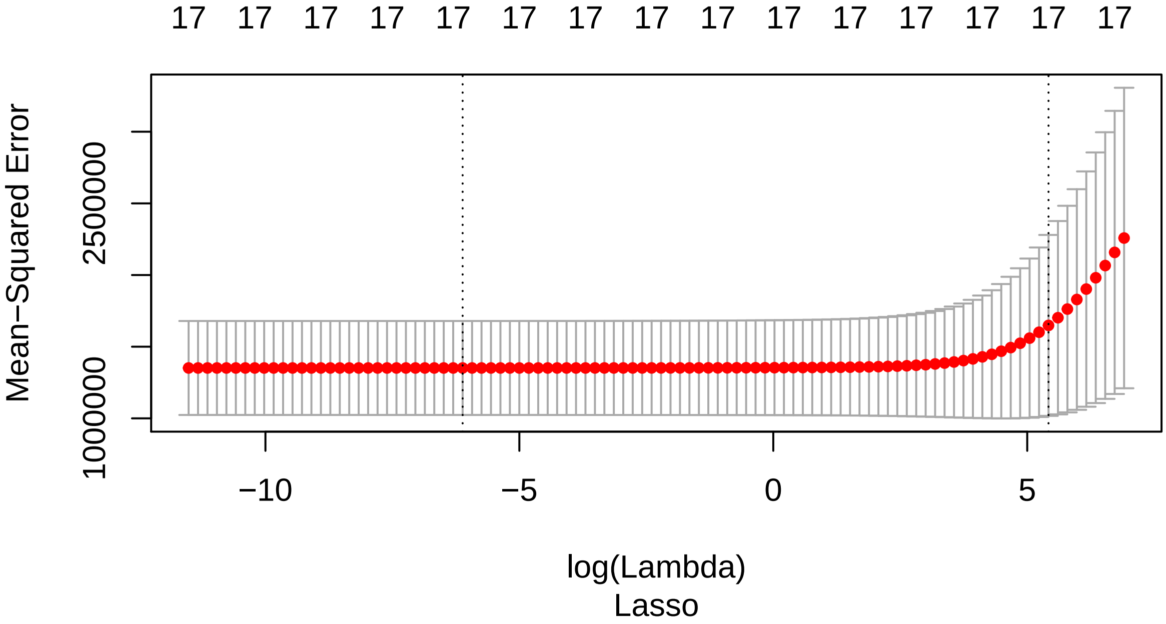
b) Now, we fit the ridge regression using cross-validation and calculate the test error to be 1044.654 at lambda = 0.000135. From the plot below we can also see that the MSE does not vary much with log(lambda) in range (-10, 3) and the test error increases if we further increase lambda beyond 1000.

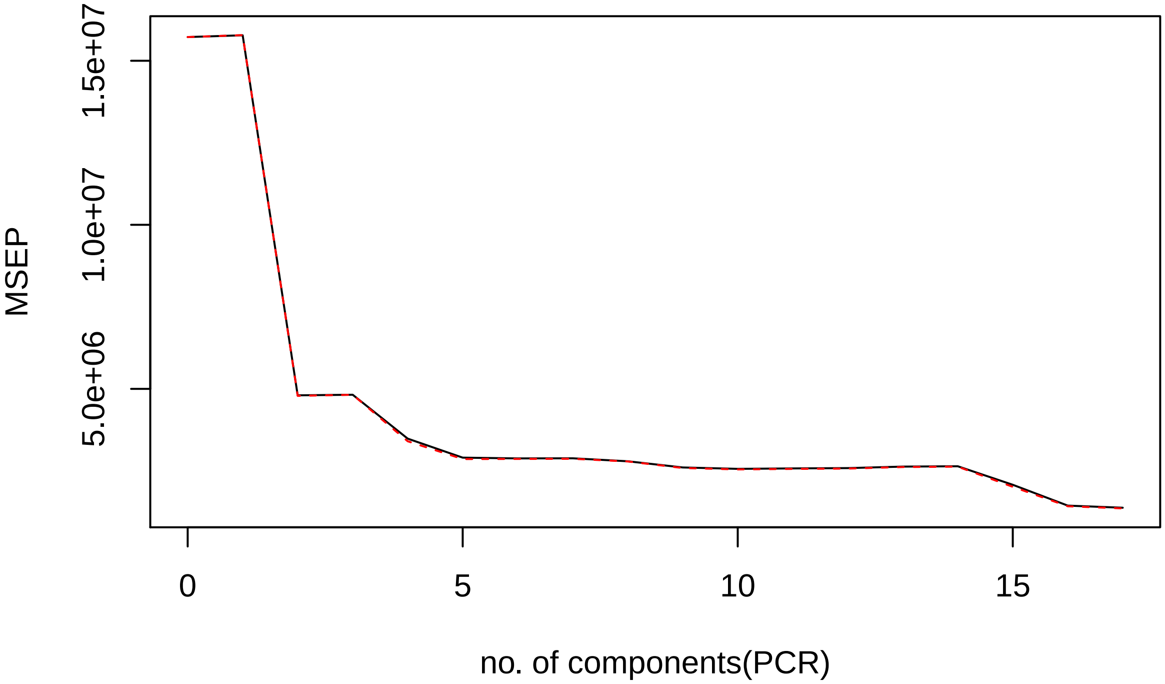
c) Here we have used lasso regression with cross-validation and get almost the same results as with ridge regression. The RMSE comes to be 1044.6562 at lambda = 10ˆ5. We also see the behaviour that the MSE does not vary much with lambda in the range of (10ˆ-10, 10ˆ3). The number of non-zero coefficients using lasso is 15, excluding the intercept.

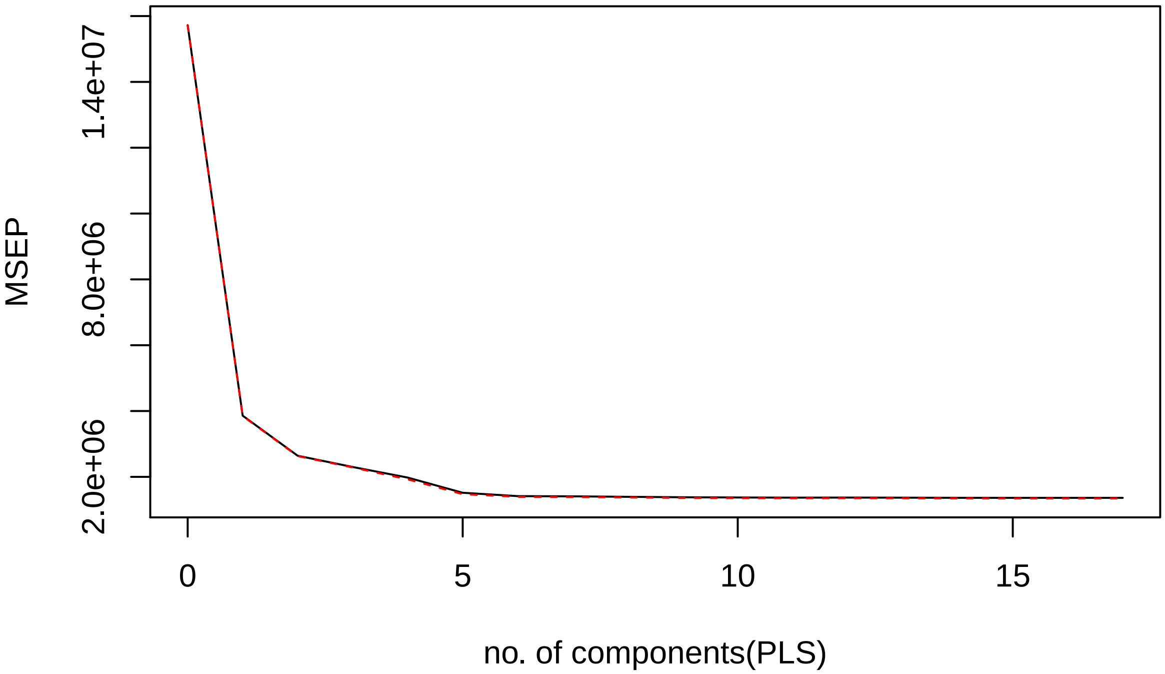
d) When we use Principal Component Regression to fit the model we see that most of the variance in the model is explained by the 15 variables(~91%). The RMSE when using 15 components comes to be around 1219.3785, while if we use all 17 components the RMSE comes to be 1044.6566. Finally we use k=15 for better interpretability.

e) Here we are implementing Partial Least Square to fit the model. In the plot below we see that most of the variance(~92.66%) is explained by only 5 variables and the RMSE on test data comes to be 1127.7290. We choose k=5 for better interpretability.




17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
Mean−Squared Error
1000000
2500000
−10
−5
0
5
log(Lambda)
Lasso


MSEP
5.0e+06
1.0e+07
1.5e+07
0
5
10
15
no. of components(PCR)



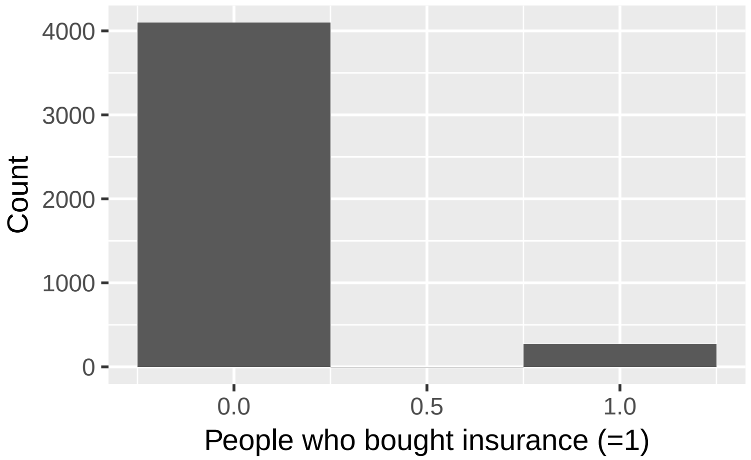
Model	RMSE
OLS	1044.66
Ridge	1044.64
Lasso	1044.66
PCR(k=15)	1219.37
PLS(k=5)	1127.73

g) As a summary, we can say that all of the methods give almost same RMSE= 1044.65 on the test data. But if we can tradeoff some error for more interpretability, partial least square gives an RMSE of just 1127.72 using just five of the variables against other methods who accuracy is good only when we include all the variables in the model.

SOLUTION 2

To predict people who will be interested in buying the caravan insurance policy, we have used ordinary least square, forward subset selection, backward subset selection, ridge regression and lasso regression. We get an error of ~5.95% on test data for OLS, forward, backward and an error of ~5.28% for ridge and lasso regression.

But when we plot the histogram of people who bought insurance we find that the classes are skewed, i.e number of people who buy insurance is 6.23%(training-set) and 5.95%(test-set). And our model's accuracy is ~5.95%. This implies that even if we say that none of the people in the test data took insurance, our accuracy will be 5.95%. Thus, even though our MSE is very low our model isn't working. This is due to the fact that our data-set is highly skewed and we cannot predict with confidance the response. We may try masking or oversampling/undersampling methods to remove the imbalance of the data-set and then perform one of the classification algorithm to predict the response.



Some extra analysis

Here I have used Random walk oversampling method to oversample the minority class to 10 times and then used linear discriminant analysis to fit the model. When observing the confusion matrix, we see that the model is not able to predict people who will buy insurance. This is almost the similar confusion matrix we get when we apply OLS without oversampling. We need to furthur deploy more sophisticated techniques to be able to predict the responses accurately.

Confusion Matrix

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

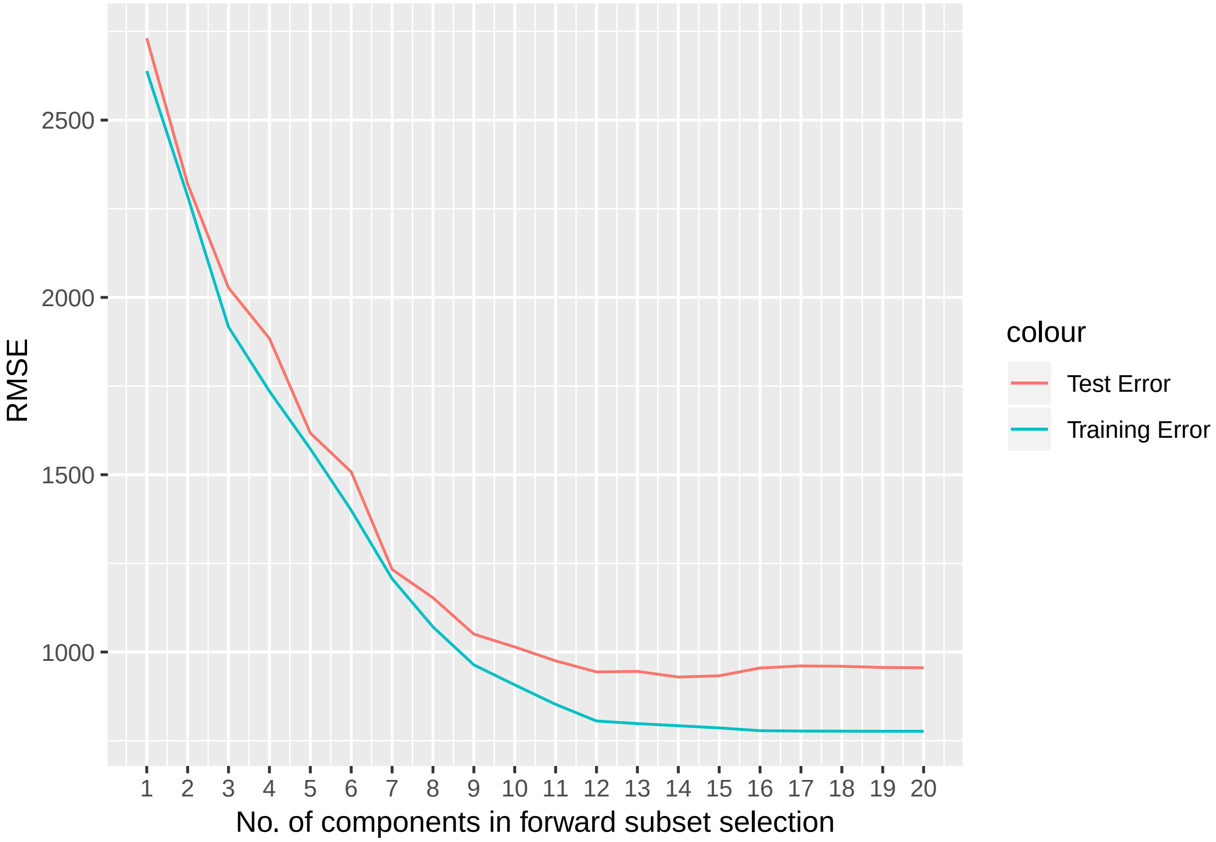
```
## The following object is masked from 'package:pls':
##
##     R2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3754  238
##          1    8    0
##
##                Accuracy : 0.9385
##                  95% CI : (0.9306, 0.9457)
##     No Information Rate : 0.9405
##     P-Value [Acc > NIR] : 0.7173
##
##                   Kappa : -0.0039
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9979
##             Specificity : 0.0000
##          Pos Pred Value : 0.9404
##          Neg Pred Value : 0.0000
##              Prevalence : 0.9405
##          Detection Rate : 0.9385
##    Detection Prevalence : 0.9980
##       Balanced Accuracy : 0.4989
##
##        'Positive' Class : 0
##
```

SOLUTION 3

For this problem, I have created 1000 observations with 20 predictors by randomly choosing integers in range 0-100 and similarly created beta by randomly choosing integers in range 0-50. we then add a Also some beta (index-3,6,12,13,16) were manually set to zero. We now calculate the response,Y, analyze and add a gaussian error to it with mean 0 and standard deviation of 1000. Then the data is divided into training and test data in the ratio of 1:9 and we perform best subset selection.

Now, we plot the RMSE between training set and test set and we see that as we increase the number of observations in our model the training error necessarily decreases but the test error does not decrease alwayss.



We can also see that that coefficients predicted from best subset selection is very close to the ones manually created. Also the model was able to discard the coefficients corresponding to the ones we have manually set to 0.

Manually created beta values:

```
## [1] 14
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]   17   48    0   32   15    0   11   22    4    38     9     0     0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## [1,]    26    50     0     5     2    10    30
```

calculated/predicted beta values:

```
## (Intercept)          X1          X2          X4          X5          X7
##  121.916055   16.350143   50.704652   31.368518   19.701855   11.057156
##          X8         X10         X11         X13         X14         X15
##   18.385426   37.184258   10.371724   -3.463529   30.272316   46.233812
##         X17         X18         X19         X20
##    4.359942    3.565863   14.599929   28.086692
```