EAS596, Homework\_2

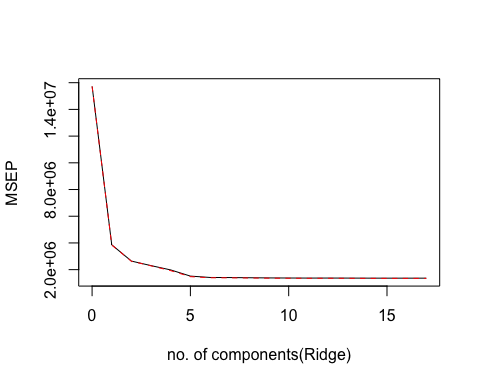
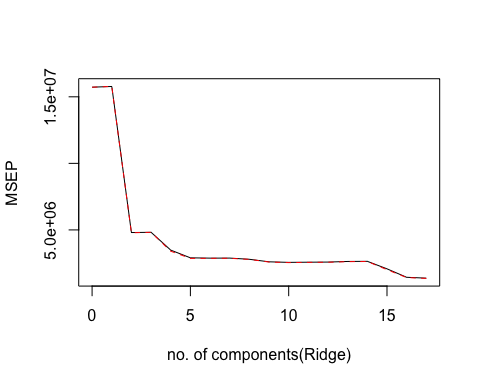
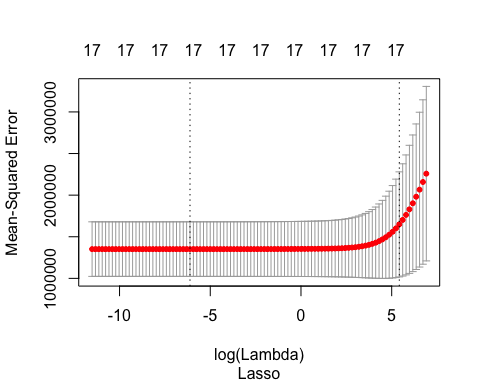
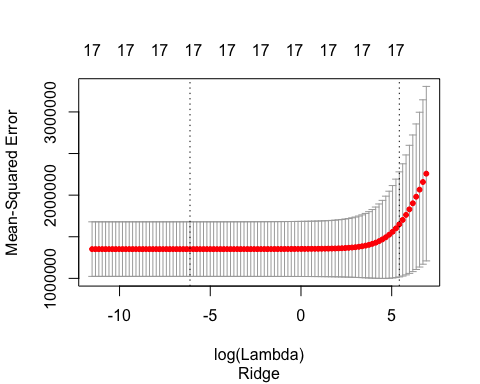
Abhishek Kumar, Class#1

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.

1. Using linear model we get an RMSE of 1044.6566 on the test set.
2. 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.
3. 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 = 10^5. We also see the behaviour that the MSE does not vary much with lambda in the range of (10^-10, 10^3).
4. 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
5. 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.



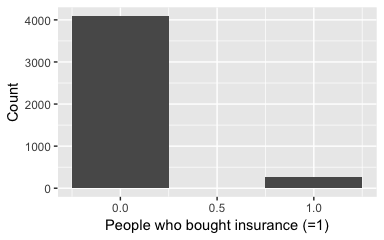
|  |  |
| --- | --- |
| Model | RMSE |
| OLS | 1044.66 |
| Ridge | 1044.64 |
| Lasso | 1044.66 |
| PCR(k=15) | 1219.37 |
| PLS(k=5) | 1127.73 |

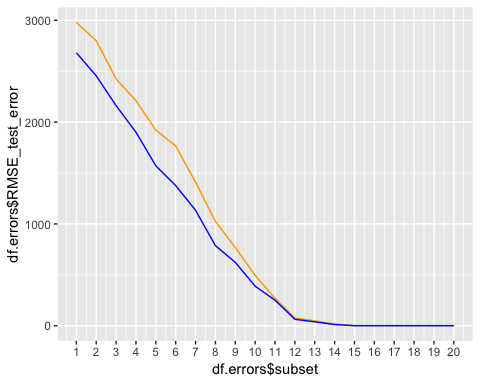
##### As a summary, we can say that all of the methods give almost same RMSE= 1044.65 on the test data. But if we can tradeoff some error for more interpretability, partial least square gives an RMSE of just 1127.72 using just five of the variables against other methods who accuracy is good only when 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 plot the histogram of the number of people who bought insurance we find that the classes are skewed, i.e we find that in training set we have only 6.23% people who took insuarance and in test set it is 5.95%. 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%. This show that 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 these algorithms to predict the response.

#LASSO REGRESSION  
library(ggplot2)  
grid =10^seq (10,-2, length = 100)  
caravan\_train\_matrix <- model.matrix(V86 ~ ., data = caravan\_train)  
caravan\_train\_y <- caravan\_train[,86]  
caravan\_cv\_matrix <- model.matrix(V86 ~ ., data = caravan\_cv)  
caravan\_cv\_y <- caravan\_cv[,86]  
  
caravan.cv.model.lasso <- cv.glmnet(caravan\_train\_matrix, caravan\_train\_y, alpha = 1)  
caravan\_bestlam\_lasso <- caravan.cv.model.lasso$lambda.min  
caravan.cv.pred.lasso <- predict(caravan.cv.model.lasso, s = caravan\_bestlam\_lasso, newx = caravan\_cv\_matrix, type = "response")  
caravan\_testMSE\_lasso <- mean((caravan.cv.pred.lasso - caravan\_cv\_y)^2) #MSE = mean\_test\_error  
  
  
ggplot(caravan\_train, aes(x = caravan\_train$V86))+ geom\_histogram(binwidth = 0.5) + labs(x = "People who bought insurance (=1)", y = "Count")



 ####Beta manually set:

## [1] 15

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]  
## [1,] 60 37 0 89 43 0 3 71 77 18 81 0 0  
## [,14] [,15] [,16] [,17] [,18] [,19] [,20]  
## [1,] 73 25 0 97 63 4 1

#### Beta(coeficients) calculated/predicted:

## (Intercept) X1 X2 X4 X5 X7   
## 1.015870 59.978484 36.999113 88.989247 42.995231 3.004158   
## X8 X9 X10 X11 X14 X15   
## 71.000099 76.987345 17.995677 81.009426 73.007207 25.014547   
## X17 X18 X19 X20   
## 96.996581 62.993056 3.992921 1.012402