Problem Set 8

Noah Estrada-Rand

10/20/2019

a) Reading In Data

```
Bikes_df <-  read.csv("day.csv")
```

#b) Factorizing All Necessary Variables

```
Bikes_df[,3:9] <- lapply(Bikes_df[,3:9],factor)
```

c) Ensuring Factorization

```
sapply(Bikes_df,is.factor)
```

```
##    instant     dteday     season         yr       mnth    holiday
##      FALSE       TRUE       TRUE       TRUE       TRUE       TRUE
##    weekday workingday weathersit       temp      atemp        hum
##       TRUE       TRUE       TRUE      FALSE      FALSE      FALSE
##  windspeed     casual registered        cnt
##      FALSE      FALSE      FALSE      FALSE
```

d) Feature Transformations

In the code below, the index column is removed and squared terms are added for casual and registered values.

```
Bikes_df <- Bikes_df[,2:ncol(Bikes_df)]
Bikes_df$casual_sq <- Bikes_df$casual^2
Bikes_df$registered_sq <- Bikes_df$registered^2
```

e) Splitting Data Into Test and Training Sets

```
set.seed(2019)
train_index <- sample(1:nrow(Bikes_df),(.7*nrow(Bikes_df)),replace = FALSE)
train_bikes <- Bikes_df[train_index,]
test_bikes <- Bikes_df[-train_index,]
```

f) Fit Forward Stepwise Linear Model

```
fwd_fit <- regsubsets(cnt ~ season + holiday + mnth + workingday +
                              weathersit + temp + hum +windspeed,
                      data = train_bikes, nvmax = 7,
                      method = "forward")
summary(fwd_fit)
```

```
## Subset selection object
```

```
## Call: regsubsets.formula(cnt ~ season + holiday + mnth + workingday +
##     weathersit + temp + hum + windspeed, data = train_bikes,
##     nvmax = 7, method = "forward")
## 21 Variables  (and intercept)
##             Forced in Forced out
## season2         FALSE      FALSE
## season3         FALSE      FALSE
## season4         FALSE      FALSE
## holiday1        FALSE      FALSE
## mnth2           FALSE      FALSE
## mnth3           FALSE      FALSE
## mnth4           FALSE      FALSE
## mnth5           FALSE      FALSE
## mnth6           FALSE      FALSE
## mnth7           FALSE      FALSE
## mnth8           FALSE      FALSE
## mnth9           FALSE      FALSE
## mnth10          FALSE      FALSE
## mnth11          FALSE      FALSE
## mnth12          FALSE      FALSE
## workingday1     FALSE      FALSE
## weathersit2     FALSE      FALSE
## weathersit3     FALSE      FALSE
## temp            FALSE      FALSE
## hum             FALSE      FALSE
## windspeed       FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: forward
##          season2 season3 season4 holiday1 mnth2 mnth3 mnth4 mnth5 mnth6
## 1  ( 1 ) " "     " "     " "     " "      " "   " "   " "   " "   " "
## 2  ( 1 ) " "     " "     "*"     " "      " "   " "   " "   " "   " "
## 3  ( 1 ) " "     " "     "*"     " "      " "   " "   " "   " "   " "
## 4  ( 1 ) " "     " "     "*"     " "      " "   " "   " "   " "   " "
## 5  ( 1 ) "*"     " "     "*"     " "      " "   " "   " "   " "   " "
## 6  ( 1 ) "*"     " "     "*"     " "      " "   " "   " "   " "   " "
## 7  ( 1 ) "*"     " "     "*"     " "      " "   " "   " "   " "   " "
##          mnth7 mnth8 mnth9 mnth10 mnth11 mnth12 workingday1 weathersit2
## 1  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 2  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 3  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 4  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 5  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 6  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 7  ( 1 ) " "   " "   "*"   " "    " "    " "    " "         " "
##          weathersit3 temp hum windspeed
## 1  ( 1 ) " "         "*"  " " " "
## 2  ( 1 ) " "         "*"  " " " "
## 3  ( 1 ) "*"         "*"  " " " "
## 4  ( 1 ) "*"         "*"  "*" " "
## 5  ( 1 ) "*"         "*"  "*" " "
## 6  ( 1 ) "*"         "*"  "*" "*"
## 7  ( 1 ) "*"         "*"  "*" "*"
```

Based on the summary shown above, the first five variables selected are temp, season4, weathersit3, humidity, season2. This roughly translates to tempurature, the season of fall, light snow/rain, humidity, and the season of spring.

g) Fit Backwards Stepwise Linear Model

```
bkwd_fit <- regsubsets(cnt ~ season + holiday + mnth + workingday +
                         weathersit + temp + hum +windspeed,
                       data = train_bikes, nvmax = 7,
                       method = "backward")
summary(bkwd_fit)
```

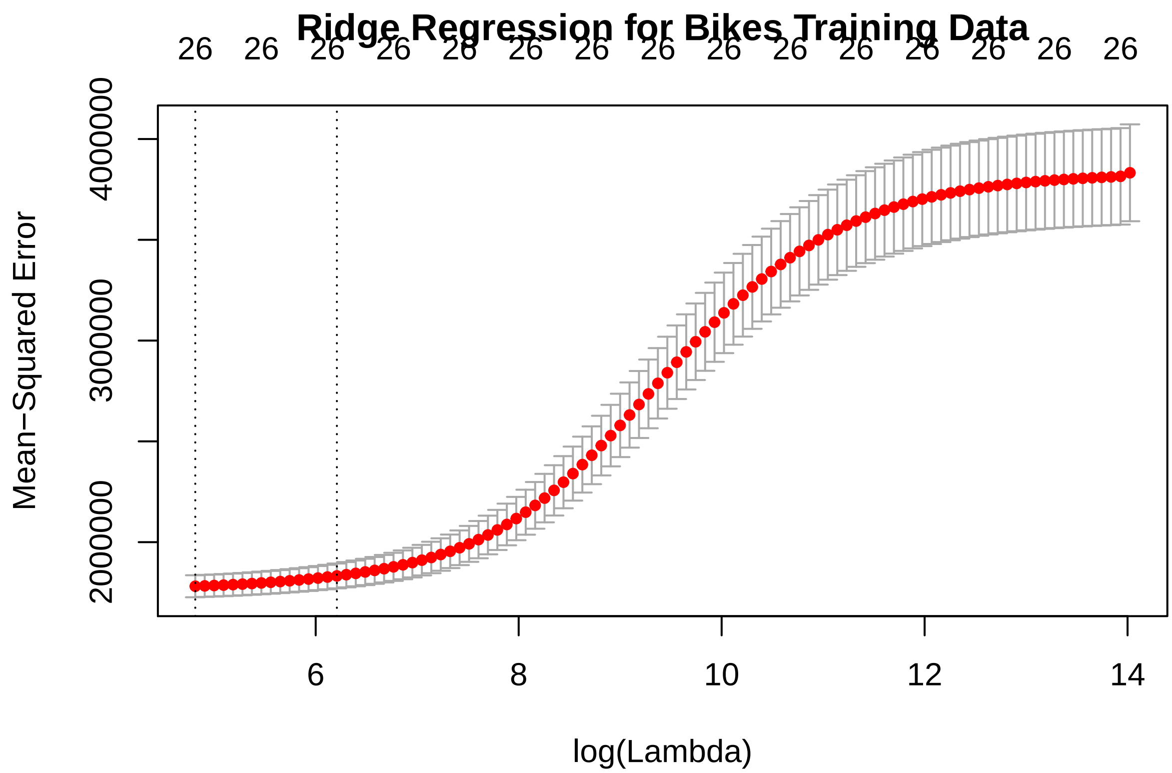
```
## Subset selection object
## Call: regsubsets.formula(cnt ~ season + holiday + mnth + workingday +
##     weathersit + temp + hum + windspeed, data = train_bikes,
##     nvmax = 7, method = "backward")
## 21 Variables  (and intercept)
##              Forced in Forced out
## season2          FALSE      FALSE
## season3          FALSE      FALSE
## season4          FALSE      FALSE
## holiday1         FALSE      FALSE
## mnth2            FALSE      FALSE
## mnth3            FALSE      FALSE
## mnth4            FALSE      FALSE
## mnth5            FALSE      FALSE
## mnth6            FALSE      FALSE
## mnth7            FALSE      FALSE
## mnth8            FALSE      FALSE
## mnth9            FALSE      FALSE
## mnth10           FALSE      FALSE
## mnth11           FALSE      FALSE
## mnth12           FALSE      FALSE
## workingday1      FALSE      FALSE
## weathersit2      FALSE      FALSE
## weathersit3      FALSE      FALSE
## temp             FALSE      FALSE
## hum              FALSE      FALSE
## windspeed        FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: backward
##          season2 season3 season4 holiday1 mnth2 mnth3 mnth4 mnth5 mnth6
## 1  ( 1 ) " "     " "     " "     " "      " "   " "   " "   " "   " "
## 2  ( 1 ) " "     " "     "*"     " "      " "   " "   " "   " "   " "
## 3  ( 1 ) " "     " "     "*"     " "      " "   " "   " "   " "   " "
## 4  ( 1 ) " "     " "     "*"     " "      " "   " "   " "   " "   " "
## 5  ( 1 ) "*"     " "     "*"     " "      " "   " "   " "   " "   " "
## 6  ( 1 ) "*"     " "     "*"     " "      " "   " "   " "   " "   " "
## 7  ( 1 ) "*"     " "     "*"     " "      " "   " "   " "   " "   " "
##          mnth7 mnth8 mnth9 mnth10 mnth11 mnth12 workingday1 weathersit2
## 1  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 2  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 3  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
## 4  ( 1 ) " "   " "   " "   " "    " "    " "    " "         " "
```

```
## 5  ( 1 ) " "    " "    " "    " "     " "     " "     " "          " "
## 6  ( 1 ) " "    " "    " "    " "     " "     " "     " "          " "
## 7  ( 1 ) "*"    " "    " "    " "     " "     " "     " "          " "
##          weathersit3 temp hum windspeed
## 1  ( 1 ) " "         "*"  " " " "
## 2  ( 1 ) " "         "*"  " " " "
## 3  ( 1 ) " "         "*"  "*" " "
## 4  ( 1 ) " "         "*"  "*" "*"
## 5  ( 1 ) " "         "*"  "*" "*"
## 6  ( 1 ) "*"         "*"  "*" "*"
## 7  ( 1 ) "*"         "*"  "*" "*"
```

Based on the above summary, it is shown that the variables used in Model 5 are season2 ,season4, temp, humidity, and windspeed. These variables are not the same as the previously fitted forward stepwise linear model. This is because we can never guarantee the same variables to result from the two stepwise approximations. Results of this nature occur because of the fact that foward stepwise approximation starts with one variable and adds the best predictors to the model, moving through all variables. Backwards does the opposite, starting with a saturated model and removing variables that result in the most insignificant reduction in R^2 value. Thus the processes will not always converge to the same answer since forwards is based on maximization of increases of R^2 while backwards is focused on the minimization of decreases in R^2.

h) Ridge Regression Plot for Training Data

```
train_ridge <- cv.glmnet(cnt ~ season + holiday + mnth + workingday +
                           weathersit + temp + hum +windspeed,
                         data = train_bikes,alpha = 0)
plot(train_ridge, main = "Ridge Regression for Bikes Training Data")
```



i) Lambda Min and Lambda 1se Values

```
print(train_ridge$lambda.min)
```

```
## [1] 123.1331
```

```
print(train_ridge$lambda.1se)
```

```
## [1] 497.0906
```

The meaning of the min lambda mentioned above is that it is the lambda that produced the lowest amount of mean cross validated error. the 1se lambda on the other hand produced the most regularized model such that the mean squared error is one standard error away from the minimum error.

j)

Ridge Regression Coefficients for Lambda Min

```
print(as.matrix(coef(train_ridge,c = train_ridge$lambda.min)))
```

```
##                        1
## (Intercept)  4058.47999
## season1      -705.31090
## season2       269.55155
## season3        73.53166
## season4       393.37737
```

```
## holiday0      140.15765
## holiday1     -139.60622
## mnth1        -617.95728
## mnth2        -445.93075
## mnth3          15.56318
## mnth4         -31.42848
## mnth5         196.00876
## mnth6         167.16098
## mnth7        -101.67869
## mnth8         273.95633
## mnth9         717.57168
## mnth10        281.46879
## mnth11       -179.11801
## mnth12       -320.43758
## workingday0   -30.81464
## workingday1    31.07162
## weathersit1   340.18860
## weathersit2  -146.89042
## weathersit3 -1522.24185
## temp         3317.74376
## hum         -1445.51040
## windspeed   -2800.60025
```

Ridge Regression Coefficients for Lambda 1se

```r
print(as.matrix(coef(train_ridge,c = train_ridge$lambda.1se)))
```

```
##                       1
## (Intercept)  4058.47999
## season1      -705.31090
## season2       269.55155
## season3        73.53166
## season4       393.37737
## holiday0      140.15765
## holiday1     -139.60622
## mnth1        -617.95728
## mnth2        -445.93075
## mnth3          15.56318
## mnth4         -31.42848
## mnth5         196.00876
## mnth6         167.16098
## mnth7        -101.67869
## mnth8         273.95633
## mnth9         717.57168
## mnth10        281.46879
## mnth11       -179.11801
## mnth12       -320.43758
## workingday0   -30.81464
## workingday1    31.07162
## weathersit1   340.18860
## weathersit2  -146.89042
## weathersit3 -1522.24185
## temp         3317.74376
## hum         -1445.51040
## windspeed   -2800.60025
```

Looking at both sets of coefficients above, it becomes clear that the coefficients are generally smaller in the case of lambda 1se. This is most likely due to the fact that the lambda 1se is a stronger penalty to the coefficients, leading them all to be smaller overall.

k) Lasso Model Estimation

```
train_lasso <- cv.glmnet(cnt ~season + holiday + mnth + workingday +
                           weathersit + temp + hum +windspeed,
                         data = train_bikes,alpha = 1)
```

l)

Lasso Model Coefficients at Lambda Min

```
print(as.matrix(coef(train_lasso,s = train_lasso$lambda.min)))
```

```
##                         1
## (Intercept)  2.986064e+03
## season1     -5.502100e+02
## season2      4.168155e+02
## season3      0.000000e+00
## season4      7.283224e+02
## holiday0     2.708828e+02
## holiday1    -2.869791e-11
## mnth1        0.000000e+00
## mnth2        0.000000e+00
## mnth3        2.408162e+02
## mnth4        0.000000e+00
## mnth5        0.000000e+00
## mnth6       -2.248092e+02
## mnth7       -5.679190e+02
## mnth8        0.000000e+00
## mnth9        6.848623e+02
## mnth10       1.901107e+02
## mnth11       0.000000e+00
## mnth12       0.000000e+00
## workingday0  0.000000e+00
## workingday1  0.000000e+00
## weathersit1  3.359837e+02
## weathersit2  0.000000e+00
## weathersit3 -1.441185e+03
## temp         6.188115e+03
## hum         -2.378461e+03
## windspeed   -3.326535e+03
```

From the coefficients shown above, it is clear that the lasso model at lambda min selected 15 variables in total.

Lasso Model Coefficients at Lambda 1se

```
print(as.matrix(coef(train_lasso,s = train_lasso$lambda.1se)))
```

```
##                       1
## (Intercept)  3428.65535
## season1     -1021.74019
```

```
## season2          0.00000
## season3          0.00000
## season4        193.27464
## holiday0         0.00000
## holiday1         0.00000
## mnth1          -12.18058
## mnth2            0.00000
## mnth3            0.00000
## mnth4            0.00000
## mnth5            0.00000
## mnth6            0.00000
## mnth7         -284.00192
## mnth8            0.00000
## mnth9          343.53588
## mnth10          46.99140
## mnth11           0.00000
## mnth12           0.00000
## workingday0      0.00000
## workingday1      0.00000
## weathersit1    354.38138
## weathersit2      0.00000
## weathersit3 -1299.92540
## temp         4683.64202
## hum         -1292.26485
## windspeed   -2052.53166
```

From the coefficients above, it becomes clear that the lasso model at lambda 1se selected a total of 11 variables.

m) Arguments for both Lasso and Ridge Regression

Both lasso and ridge regression have their rightful place in selecting models with optimized predictors. However, neither one is preferred for all scenarios. Lasso is more useful when we know that not all variables play a role in predicting the outcome and as such we can afford to remove them from our models. Furthermore Lasso is a stronger selection when the data generating process is sparse. Ridge regression, on the other hand is more useful when each variable has a significant effect on the outcome, even a little. Moreover, in cases where accuracy is paramount ridge regression is the preferred approach to model selection.