HW 8

Charlie Marcou, Carrie Mecca, Jessie Bustin

2022-11-30

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.2.2
```

```
library(lars)
```

```
## Loaded lars 1.3
```

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 4.2.2
```

```
##
## Attaching package: 'pls'
##
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

1) Train Test Split

```
# Reading Transformed Data In From CS1
sub1_data<-read.csv("sub1_data")

# Convert Region to factor level
sub1_data <- sub1_data %>% mutate(region = as.factor(region))

# Setting Seed
set.seed(425)

# Train Test Split
sample <- sample(c(TRUE, FALSE), nrow(sub1_data), replace=TRUE, prob=c(0.7, 0.3))
train <- sub1_data[sample, ]
test <- sub1_data[!sample, ]

# Create Table for Results
results <- data.frame(matrix(ncol = 12, nrow = 5))
colnames(results) <- c("model", "trainRMSE", "testRMSE", "pop_18to24", "pop_over65", "poverty_rate", "u
```

2)

```
#Creating model using training data
cs1_model<-lm(log_physicians ~ ., data=train)

#Creating function to calculate RMSE
rmse<-function(x,y) sqrt(mean((x-y)^2))

#Train MSE
rmse(fitted(cs1_model), train$log_physicians)
```

```
## [1] 0.2743425
```

```
#Test MSE
rmse(predict(cs1_model, test), test$log_physicians)
```

```
## [1] 0.2521535
```

```
results[1,1] <- "Full Model"
results[1,2] <- rmse(fitted(cs1_model), train$log_physicians)
results[1,3] <- rmse(predict(cs1_model, test), test$log_physicians)
results[1,4:12] <- TRUE
```

3)

```r
regsubsets_selection=regsubsets(log_physicians~., data = train)
rs = summary(regsubsets_selection)

# Adjusted-R2, 8th is best
rs$adjr2
```

```
## [1] 0.8030435 0.8768155 0.9297244 0.9321684 0.9366515 0.9385849 0.9392084
## [8] 0.9396167
```

```r
# BIC, 2nd is best
# Note that this is not the same as our calculated BIC
rs$bic
```

```
## [1] -478.6473 -615.2078 -779.4513 -785.4135 -801.3066 -805.9513 -804.3412
## [8] -801.6914
```

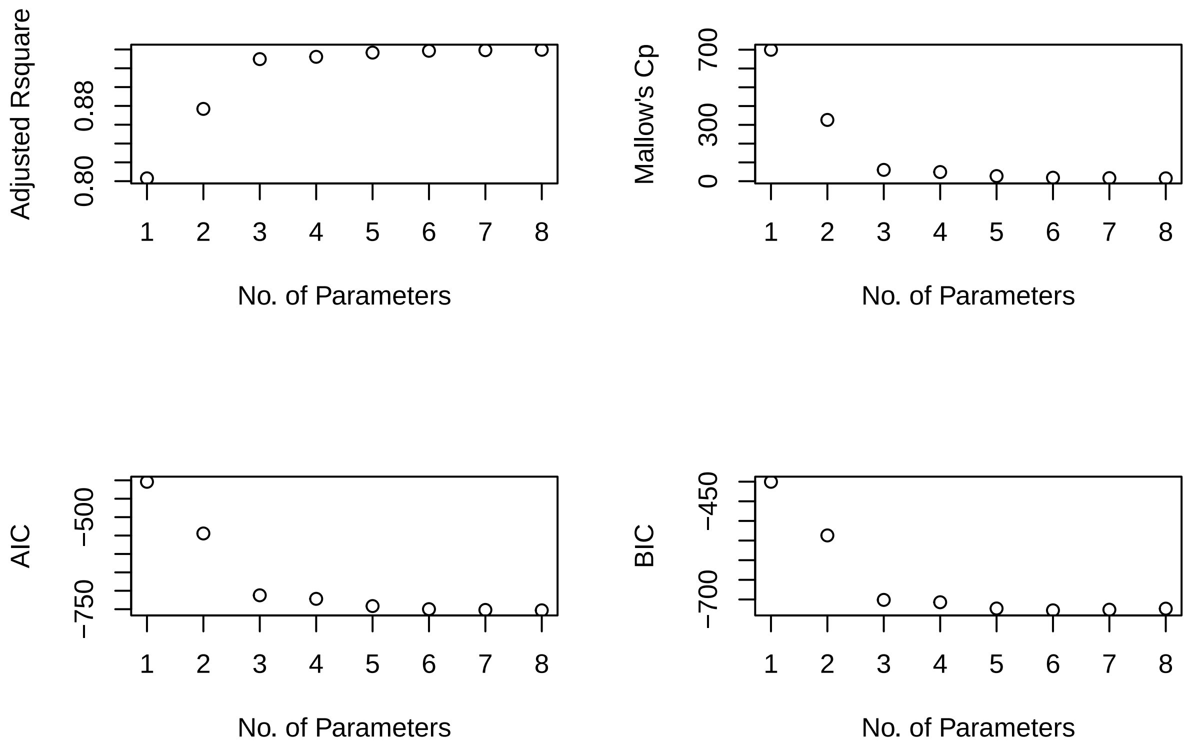
```r
#We will compute AIC and BIC by hand
n=dim(train)[1]
msize = 1:8

AIC = n*log(rs$rss/n) + 2*msize;
which.min(AIC) #8 is best
```

```
## [1] 8
```

```r
BIC = n*log(rs$rss/n) + msize*log(n);
which.min(BIC) #6 is best
```

```
## [1] 6
```

```r
par(mfrow=c(2,2))
plot(msize, rs$adjr2, xlab="No. of Parameters", ylab = "Adjusted Rsquare");
plot(msize, rs$cp, xlab="No. of Parameters", ylab = "Mallow's Cp");
plot(msize, AIC, xlab="No. of Parameters", ylab = "AIC");
plot(msize, BIC, xlab="No. of Parameters", ylab = "BIC");
```




```r
#Determining which variables to keep based
#Because both Adjusted R2 and AIC suggested 8 variables, we will choose 8 variables
rs$which[8,]
```

```
##       (Intercept)         pop_18to24         pop_over65       poverty_rate
##              TRUE              FALSE               TRUE               TRUE
## unemployment_rate            region2            region3            region4
##             FALSE               TRUE              FALSE               TRUE
##           log_pop     log_bachelors log_percap_income log_hospital_beds
##              TRUE               TRUE               TRUE               TRUE
```

```r
select.var = colnames(rs$which)[rs$which[8,]]

select.var = select.var[-1]

#fitting model
criteria_fit <- lm(log_physicians ~ . , data=train[, c("pop_over65", "poverty_rate", "region", "log_pop

#Using RMSE function from earlier we will calculate errors
#Train RMSE
rmse(fitted(criteria_fit), train$log_physicians)
```

```
## [1] 0.27868
```

```
#Test RMSE
rmse(predict(criteria_fit, test), test$log_physicians)
```

```
## [1] 0.2623562
```

```
results[2,1] <- "Citerion Selected Model"
results[2,2] <- rmse(fitted(criteria_fit), train$log_physicians)
results[2,3] <- rmse(predict(criteria_fit, test), test$log_physicians)
results[2, c("pop_over65", "poverty_rate", "region", "log_pop", "log_bachelors", "log_percap_income", "l
results[2,c(4,7)] <- FALSE
```

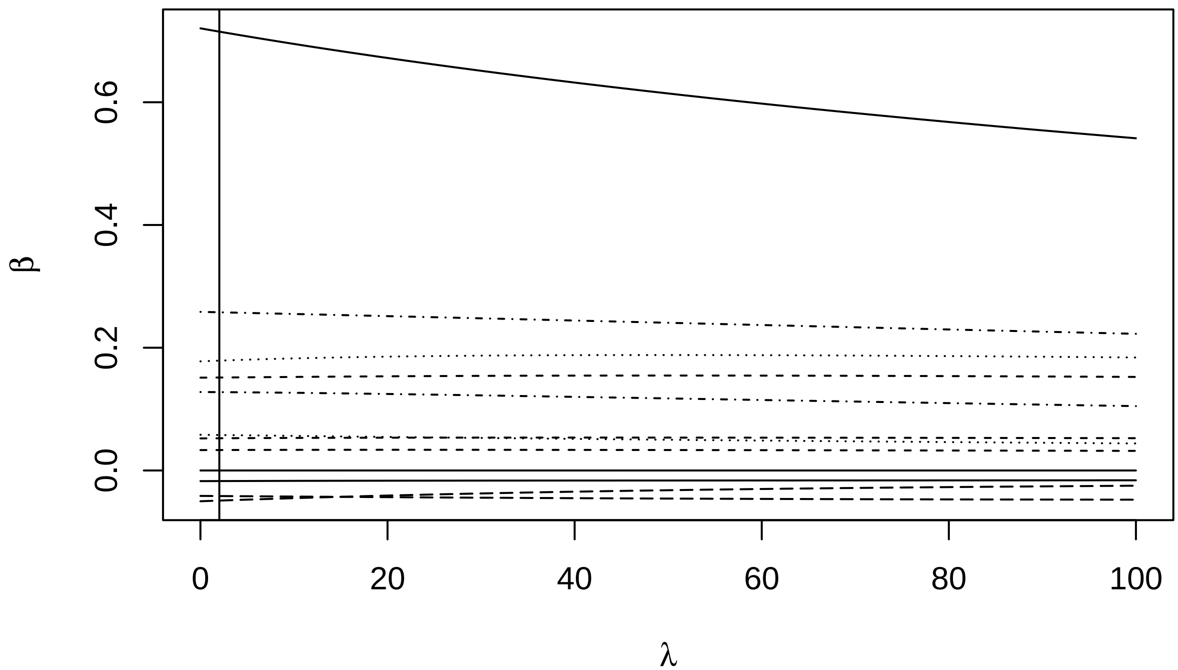
```
model_train <- train
model_test <- test

model_train$region2 <- ifelse(train$region == 2, 1, 0)
model_test$region2 <- ifelse(test$region == 2, 1, 0)
model_train$region3 <- ifelse(train$region == 3, 1, 0)
model_test$region3 <- ifelse(test$region == 3, 1, 0)
model_train$region4 <- ifelse(train$region == 4, 1, 0)
model_test$region4 <- ifelse(test$region == 4, 1, 0)

model_train <- data.frame(model_train %>%
  dplyr::select(-region))
model_test <- data.frame(model_test %>%
  dplyr::select(-region))
```

##4) Ridge

```
#standardize df
phys_train <- model_train %>% mutate_all(~(scale(.) %>% as.vector))
phys_test <- model_test %>% mutate_all(~(scale(.) %>% as.vector))

phys.ridge <- lm.ridge(log_physicians~., phys_train, lambda=seq(0, 100, len=100))
which.min(phys.ridge$GCV)
```

```
##   3.030303
##          4
```

```
matplot(phys.ridge$lambda, coef(phys.ridge), type="l", xlab=expression(lambda), ylab=expression(hat(beta
abline(v=2.020202)
```




```
phys.pred.train <- cbind(1, as.matrix(phys_train[,-5]))%*% coef(phys.ridge)[8,]

rmse(phys.pred.train, phys_train$log_physicians)
```

```
## [1] 0.2388474
```

```
phys.pred.test <- cbind(1, as.matrix(phys_test[,-5]))%*% coef(phys.ridge)[8,]

rmse(phys.pred.test, phys_test$log_physicians)
```

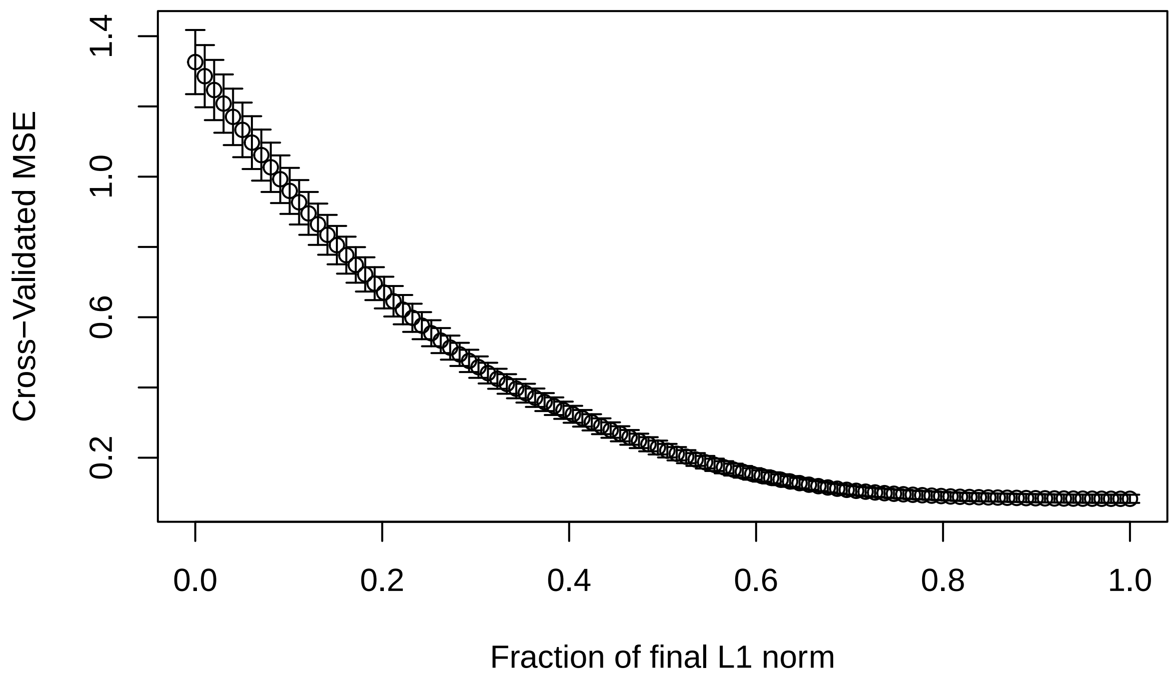
```
## [1] 0.2283761
```

```
results[3,1] <- "Ridge Model"
results[3,2] <- rmse(phys.pred.train, phys_train$log_physicians)
results[3,3] <- rmse(phys.pred.test, phys_test$log_physicians)
results[3,4:12] <- TRUE
```

##5) LASSO

```
train.y<-model_train$log_physicians
train.x<-as.matrix(model_train[,-5])

test.x<-as.matrix(model_test[,-5])
```

```
physlasso<-lars(train.x,train.y)
cv.ml<-cv.lars(train.x,train.y)
```




```
which.min(cv.ml$cv)
```

```
## [1] 100
```

```
svm<-cv.ml$index[which.min(cv.ml$cv)]
svm
```

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

```
predlasso_train <- predict(physlasso, train.x, s = svm, mode = "fraction")
rmse(model_train$log_physicians, predlasso_train$fit)
```

```
## [1] 0.2743425
```

```
predlasso_test<-predict(physlasso, test.x, s=svm, mode="fraction")
rmse(predlasso_test$fit, model_test$log_physicians)
```

```
## [1] 0.2521535
```

```r
coef(physlasso, s=svm, mode="fraction")
```

```
##       pop_18to24        pop_over65      poverty_rate unemployment_rate
##       0.01445188        0.01658768        0.03488714       -0.02551793
##          log_pop     log_bachelors log_percap_income log_hospital_beds
##       1.05723858        0.49881945        1.02822446        0.54055102
##          region2           region3           region4
##      -0.11060879       -0.04138433        0.10185466
```

```r
results[4,1] <- "LASSO Model"
results[4,2] <- rmse(model_train$log_physicians, predlasso_train$fit)
results[4,3] <- rmse(predlasso_test$fit, model_test$log_physicians)
results[4,4:12] <- TRUE
```

##6) PCR

```r
phys.pcr<-pcr(log_physicians ~ ., scale=TRUE, data=model_train,ncomp=11)
summary(phys.pcr)
```

```
## Data:    X dimension: 301 11
##  Y dimension: 301 1
## Fit method: svdpc
## Number of components considered: 11
## TRAINING: % variance explained
##                 1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X                 26.12    43.08    56.54    69.66    81.46    88.28    92.40
## log_physicians    15.48    15.48    53.31    64.26    89.97    91.89    93.66
##                 8 comps  9 comps  10 comps  11 comps
## X                 95.34    97.48      99.1    100.00
## log_physicians    94.29    94.30      94.3     94.31
```

```r
#Based on the summary 6 components seems reasonable as it brings us to over 85% of the variation explai

rmse(predict(phys.pcr, ncomp=6), model_train$log_physicians)
```

```
## [1] 0.3273691
```

```r
rmse(predict(phys.pcr, model_test, ncomp=6), model_test$log_physicians)
```

```
## [1] 0.3449859
```

```r
results[5,1] <- "Principal Componant Regression"
results[5,2] <- rmse(predict(phys.pcr, ncomp=6), model_train$log_physicians)
results[5,3] <- rmse(predict(phys.pcr, model_test, ncomp=6), model_test$log_physicians)
results
```

```
##                            model trainRMSE  testRMSE pop_18to24 pop_over65
## 1                     Full Model 0.2743425 0.2521535       TRUE       TRUE
## 2       Citerion Selected Model 0.2786800 0.2623562      FALSE       TRUE
## 3                    Ridge Model 0.2388474 0.2283761       TRUE       TRUE
```

```
## 4                   LASSO Model 0.2743425 0.2521535       TRUE       TRUE
## 5 Principal Componant Regression 0.3273691 0.3449859         NA         NA
##   poverty_rate unemployment_rate region log_pop log_bachelors log_percap_income
## 1         TRUE              TRUE   TRUE    TRUE          TRUE              TRUE
## 2         TRUE             FALSE   TRUE    TRUE          TRUE              TRUE
## 3         TRUE              TRUE   TRUE    TRUE          TRUE              TRUE
## 4         TRUE              TRUE   TRUE    TRUE          TRUE              TRUE
## 5           NA                NA     NA      NA            NA                NA
##   log_hospital_beds
## 1              TRUE
## 2              TRUE
## 3              TRUE
## 4              TRUE
## 5                NA
```

Above is a table with the summary of our analysis. Our original model performed relatively well. We attribute this to the feature engineering and selected we completed prior to completing case study 1. We either transformed or removed highly correlated variables. The criteria selected model using leaps and bounds removed 2 variables and the RMSE increased slightly. Chosing between these 2 models would come down to weighing model complexity over performance. For the penalized regression models, ridge performed exceptionally well. Overall, we would select the Ridge model as our ideal model due to the low train and test RMSE and that there isn't not a large difference in train verses test RMSE. We were not surprised that LASSO returned the full model after tuning lambda. Finally, the PCR model was the worst of the group. We feel that it might have performed better if we had not done the feature selection and engineering in our EDA for case study 1. In conclusion, we were satisfied with our full model but prefer the Ridge model as the best model.