Homework 1

Yiying Wu (yw3996)

R packages

```
library(tidyverse)
library(caret)
library(tidymodels)
```

Input dataset

```
housing_train<-read_csv("./data/housing_training.csv")
housing_train <- na.omit(housing_train)
housing_test<-read_csv("./data/housing_test.csv")
housing_test <- na.omit(housing_test)
```

Response: Sale price

(a) Fit a lasso model on the training data. Report the selected tuning parameter and the test error. When the 1SE rule is applied, how many predictors are included in the model?

```
ctrl1 <- trainControl(method = "repeatedcv",
                      number = 10,
                      repeats = 5,
                      selectionFunction = "oneSE")

ctrl2 <- trainControl(method = "repeatedcv",
                      number = 10,
                      repeats = 5,
                      selectionFunction = "best")
```

Using the minimal MSE rule

```
set.seed(8106)
lasso.fit2 <- train(Sale_Price ~ .,
                    data = housing_train,
                    method = "glmnet",
                    tuneGrid = expand.grid(alpha = 1,
                                           lambda = exp(seq(10, 0, length = 200))),
                    trControl = ctrl2)
# plot(lasso.fit1, xTrans = log)
```

Here's the selected tuning parameter when the minimal MSE rule is applied

```
lasso.fit2$bestTune
```

```
##    alpha   lambda
## 80     1 52.97654
```

The best tuning parameter is 52.977

And the test error is

```
lasso.pred2 <- predict(lasso.fit2, newdata = housing_test)
# test error
mean((lasso.pred2 - housing_test$Sale_Price)^2)
```

```
## [1] 441223469
```

MSE=4.4122347×10^8

Using 1SE rule

```
set.seed(8106)
lasso.fit1 <- train(Sale_Price ~ .,
                    data = housing_train,
                    method = "glmnet",
                    tuneGrid = expand.grid(alpha = 1,
                                           lambda = exp(seq(10, 0, length = 200))),
                    trControl = ctrl1)
# plot(lasso.fit1, xTrans = log)
```

Here's the selected tuning parameter when 1SE rule is applied

```
lasso.fit1$bestTune
```

```
##     alpha   lambda
## 123     1 459.7364
```

The best tuning parameter is 459.736

And the test error is

```
lasso.pred1 <- predict(lasso.fit1, newdata = housing_test)
# test error
mean((lasso.pred1 - housing_test$Sale_Price)^2)
```

```
## [1] 419720235
```

MSE=4.1972023×10^8

coefficients in the final model are

```r
# coefficients in the final model
coef(lasso.fit1$finalModel, lasso.fit1$bestTune$lambda)
```

```
## 40 x 1 sparse Matrix of class "dgCMatrix"
##                                       s1
## (Intercept)                -3.752039e+06
## Gr_Liv_Area                 6.035509e+01
## First_Flr_SF                9.726397e-01
## Second_Flr_SF               .           
## Total_Bsmt_SF               3.640038e+01
## Low_Qual_Fin_SF            -3.380912e+01
## Wood_Deck_SF                9.720812e+00
## Open_Porch_SF               1.140064e+01
## Bsmt_Unf_SF                -2.051679e+01
## Mas_Vnr_Area                1.323664e+01
## Garage_Cars                 3.399909e+03
## Garage_Area                 9.941195e+00
## Year_Built                  3.135644e+02
## TotRms_AbvGrd              -2.354873e+03
## Full_Bath                  -1.036505e+03
## Overall_QualAverage        -3.856229e+03
## Overall_QualBelow_Average  -1.058701e+04
## Overall_QualExcellent       8.846325e+04
## Overall_QualFair           -8.420850e+03
## Overall_QualGood            1.095588e+04
## Overall_QualVery_Excellent  1.576750e+05
## Overall_QualVery_Good       3.724160e+04
## Kitchen_QualFair           -1.199318e+04
## Kitchen_QualGood           -5.539188e+03
## Kitchen_QualTypical        -1.452088e+04
## Fireplaces                  7.750535e+03
## Fireplace_QuFair           -2.866628e+03
## Fireplace_QuGood            2.811236e+03
## Fireplace_QuNo_Fireplace    .           
## Fireplace_QuPoor           -5.028053e+02
## Fireplace_QuTypical        -3.325362e+03
## Exter_QualFair             -1.664540e+04
## Exter_QualGood              .           
## Exter_QualTypical          -4.781746e+03
## Lot_Frontage                8.441572e+01
## Lot_Area                    5.885854e-01
## Longitude                  -2.077860e+04
## Latitude                    3.485898e+04
## Misc_Val                    2.093642e-01
## Year_Sold                  -1.110221e+02
```

Therefore, there are 29 predictors included in the model.

(b) Fit an elastic net model on the training data. Report the selected tuning parameters and the test error. Is it possible to apply the 1SE rule to select the tuning parameters for elastic net? If the 1SE rule is applicable, implement it to select the tuning parameters. If not, explain why.

Using the minimal MSE rule

```
set.seed(8106)
enet.fit2 <- train(Sale_Price ~ .,
                   data = housing_train,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                          lambda = exp(seq(10, 0, length = 200))),
                   trControl = ctrl2)
```

Here's the selected tuning parameter

```
enet.fit2$bestTune
```

```
##     alpha   lambda
## 328  0.05 591.0553
```

The best tuning parameter is 591.055

And the test error is

```
enet.pred2 <- predict(enet.fit2, newdata = housing_test)
# test error
mean((enet.pred2 - housing_test$Sale_Price)^2)
```

```
## [1] 438501913
```

MSE=4.3850191×10^8

Using the 1SE rule

```
set.seed(8106)
enet.fit1 <- train(Sale_Price ~ .,
                   data = housing_train,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                          lambda = exp(seq(10, 0, length = 200))),
                   trControl = ctrl1)
```

Here's the selected tuning parameter

```
enet.fit1$bestTune
```

```
##     alpha   lambda
## 176     0 6594.359
```

The best tuning parameter is 6594.359

And the test error is

```r
enet.pred1 <- predict(enet.fit1, newdata = housing_test)
# test error
mean((enet.pred1 - housing_test$Sale_Price)^2)
```

```
## [1] 426612358
```

MSE=4.2661236×10^8

Given the substantial difference in lambda values between the minimal MSE and the 1SE rule in the results, it suggests that the simpler model under the 1SE rule is significantly more regularized. Given that the 1SE rule led to a model with lower MSE on the test data, it would be reasonable to favor this approach for selecting tuning parameters in the elastic net model.

Also, the change from alpha = 0.05 to alpha = 0 under the 1SE rule indicates a shift from a slight Lasso preference towards a pure Ridge regression approach. In this way, all predictors are kept in the model, leading to models that may be less sparse but can handle multicollinearity better.

(c) Fit a partial least squares model on the training data and report the test error. How many components are included in your model?

(d) Choose the best model for predicting the response and explain your choice.

(e) If "caret" was used for the elastic net in (b), retrain this model with "tidymodels", and vice versa. Compare the selected tuning parameters between the two software approaches. Should there be discrepancies in the chosen parameters, discuss potential reasons for these differences.