Lab-13

your name here

April 29, 2022

Preface

The goal of this assignment is to help you gain familiarity with using ridge and lasso models to make predictions. Unlike in recent assignments, we've included some scaffolded code since these methods are a bit complicated/new. As always, please come to office hours and reach out to your teaching staff if you have any questions.

Data

We will work with the Airbnb listings from Inside Airbnb.

Splitting data into train and test samples

First, let's split the data into test and train samples to help evaluate in-sample and out-of-sample fit. We are providing code here so you can focus on the more substantive parts of the lab. Below we randomly split `listings` into two subsamples, with the training sample of 10% of the sample size, and the testing sample of the remaining 90% of the sample.

```
# randomly split the sample into a training set and a testing set
smp_size <- round(0.1*nrow(listings)) # 0.1 gives 10% of the full sample
set.seed(2850) # set random number generator's so results are reproducible when knitted
train_rows <- sample(seq_len(nrow(listings)), size = smp_size)
listings_train <- listings[train_rows,]
listings_test <- listings[-train_rows,]

# do those numbers look right?
nrow(listings)
```

```
## [1] 13072
```

```
nrow(listings_train) # should be ~10% of nrow(listings)
```

```
## [1] 1307
```

```
stopifnot(nrow(listings_test) == (nrow(listings) - nrow(listings_train)))
```

1. Let's start by building OLS regression models to predict the price of airbnb listings. Use the training data in `listings_train` to build three OLS models: (1) `lm_null` that includes only an intercept as the explanatory variable; (2) `lm_small` that includes `room_type`, `host_is_superhost`, `host_listings_count`, `host_has_profile_pic`, and `host_identity_verified` as the explanatory variables; (3) `lm_big` with all variables except for price as the explanatory variables. Compute and store the mean squared error for all three models in a single data frame for comparison with other models we'll estimate later. Do prediction errors increase or decrease as we add more variables to the linear model? Why?

Note: If you don't know where to start on this lab, review example-13-solutions on RStudio Cloud!

```
# build ols models


# compute mean squared error and assign it to mse_train


# print mse_train
```

...

2. Use `summary()` to print the coefficient estimates for `lm_small`. Do the results make intuitive sense? Do you notice any surprising patterns?

Note: Feel free to inspect the coefficients for `lm_big`, too! There are interesting patterns, for example the differences in price across boroughs holding other factors fixed. But for simplicitly we're not asking you to print all the many coefficients in `lm_big` in your PDF submission.

. . .

3. Now use these three OLS models to predict prices in the testing data. What are the mean squared error of these three models? How do they compare to the mean squared error of the three models in the training data?

```
# compute mean squared error and assign it to mse_test


# how does it compare to what we saw before?
```

...

4. Use ridge regression to predict the price in the training data. Use `cv.glmnet()` to search for the optimal value of lambda, then use that to make predictions. Compute MSE for these predictions and add this to the data frame with each model's MSE you created in question 1. How does this MSE compare with the big OLS model in the train data?

```
# prep the train data for glmnet (we have done this for you)
x_train <- model.matrix(price ~ .,
                        data = listings_train)
y_train <- listings_train %>% pull(price)

# use cv.glmnet to search for the optimal lambda


# use the cross-validated "lambda.min" that minimizes MSE for predictions
# make predictions, then use them to compute MSE


# print mse_train
```

5. Now use the ridge regression and optimal lambda from question 4 to predict the price in the testing data. Add them in the data frame you created in question 3. How do the mean squared errors compare with those from the ridge regression with the training data and from the big OLS model?

```
# prep the test data for glmnet (we have done this for you)
x_test <- model.matrix(price ~ .,
                       data = listings_test)
y_test <- listings_test %>% pull(price)

# then predict in the test data


# how does it compare to what we saw before?
```

...

6. OPTIONAL: Use lasso regression to predict the price with the training data and the testing data. Does it do better than the big OLS model?

...

7. OPTIONAL: Use the lambda values we found for ridge and lasso in earlier questions to estimate ridge and lasso models on the full sample. Print the coefficients. You can also print the OLS coefficients for comparison if you'd like. What do you see?