

HW_6

Jared Brotamonte

10/30/2023

Exercise 1

(a)

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.1.3
```

```
data("College")
```

```
# Set the seed for reproducibility  
set.seed(578)
```

```
# Split the data into training (80%) and validation (20%) sets  
train_indices <- sample(1:nrow(College), 0.8 * nrow(College))  
train_data <- College[train_indices, ]  
validation_data <- College[-train_indices, ]
```

(b)

```
# Fit a linear model using least squares on the training set  
lm_fit <- lm(Apps ~ ., data = train_data)
```

```
# Predict on the validation set  
predictions <- predict(lm_fit, newdata = validation_data)
```

```
# Calculate the mean squared error (test error)  
test_error <- mean((predictions - validation_data$Apps)^2)  
test_error
```

```
## [1] 1206042
```

(c)

```

# Prepare the predictors and response variables for the training set
x_train <- model.matrix(~., data = train_data[, -which(names(train_data) %in% "Apps")])
y_train <- train_data$Apps

# Remove rows with NA values
complete_rows <- complete.cases(x_train, y_train)
x_train <- x_train[complete_rows, ]
y_train <- y_train[complete_rows]

# Perform cross-validated Ridge regression
set.seed(578)
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0) # Use alpha = 0 for Ridge regression

# Best lambda chosen by cross-validation
best_lambda <- cv_ridge$lambda.min

# Prepare predictors for the validation set
x_valid <- model.matrix(~., data = validation_data[, -which(names(validation_data) %in% "Apps")])

# Predict on the validation set
ridge_predictions <- predict(cv_ridge, s = best_lambda, newx = x_valid)

# Calculate the mean squared error (test error)
ridge_test_error <- mean((ridge_predictions - validation_data$Apps)^2)
ridge_test_error

```

```
## [1] 1310134
```

(d)

```

# Perform cross-validated Lasso regression
set.seed(578)
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1) # Use alpha = 1 for Lasso regression

# Best lambda chosen by cross-validation
best_lambda_lasso <- cv_lasso$lambda.min

# Fit Lasso regression model with the best lambda on the training set
lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = best_lambda_lasso)

# Prepare predictors for the validation set
x_valid <- model.matrix(~., data = validation_data[, -which(names(validation_data) %in% "Apps")])

# Predict on the validation set
lasso_predictions <- predict(lasso_model, s = best_lambda_lasso, newx = x_valid)

# Calculate the mean squared error (test error) for Lasso regression
lasso_test_error <- mean((lasso_predictions - validation_data$Apps)^2)
lasso_test_error

```

```
## [1] 1211278
```

(e)

In comparing the three models, the least squares linear regression showed the best performance with the lowest test error, suggesting its accuracy on the validation set. Both Ridge and Lasso regression had slightly higher test errors, indicating a slightly weaker performance in this case. Regarding the differences in test errors among the models, the errors were similar implying that the choice between regression methods had minor impact on prediction performance.

Exercise 2

(a)

- iii. Steadily increase

When lambda is equal to zero, the ridge regression equation is equal to the least squares estimate for beta which should already minimize the training error value. Thus as lambda increases, the training error value can only increase.

(b)

- ii. Decrease initially and then eventually start increasing in a U shape.

As lambda increases, the test error initially decreases due to a reduced variance. But a large lambda leads to underfitting, thus increasing bias and test error.

(c)

- iv. Steadily decrease.

As lambda increases, variance will decrease by shrinking the coefficients towards zero. To the point that excessive shrinkage can reduce variance to nearly zero.

(d)

- iii. Steadily increase.

As lambda increases, bias will increase by shrinking the coefficients towards zero. Increasing lambda leads to a decrease in flexibility which in turn will lead to an increase in bias as the coefficients reach zero.

(e)

- v. remain constant

As lambda increases, irreducible error remains constant no matter what the value of lambda is. This is because in this equation, lambda affects bias and variance, not the irreducible error.