CreditCardReport3

Introduction

In the third report, I will be using data from the Credit Card Acceptance Report. In this project, I will be incorporating results from my first and second reports while rulling LASSO and Ridge regressions. LASSO, or Least Absolute Selection and Shrinkage Operator, and Ridge are shrinkage methods. I will be conducting Ridge first in my report. Then, I will be running a LASSO regression. This would allow me to then compare both methods to how my logit method in report 2 performed. I will then run a regression or classification tree. Also, for extra credit, I will use boot-strap and fit 100 different trees to my boot-strap subsamples.

```
library(ISLR)
library(AER)
library(ggplot2)
library(dplyr)
library(glmnet)
data("CreditCard")
CreditCard = data.frame(CreditCard)
names(CreditCard)
```

```
##  [1] "card"        "reports"     "age"         "income"      "share"
##  [6] "expenditure" "owner"       "selfemp"     "dependents"  "months"
## [11] "majorcards"  "active"
```

```
dim(CreditCard)
```

```
## [1] 1319   12
```

```
CreditCard$card <- ifelse(CreditCard$selfemp=="yes", 1, 0)
CreditCard$owner <- ifelse(CreditCard$owner=="yes", 1, 0)
CreditCard$selfemp <- ifelse(CreditCard$selfemp=="yes", 1, 0)
```

1)Divide credit card data into training and testing subsets and setting up for ridge and lasso regressions

```
set.seed(1)
train = CreditCard %>% sample_frac(.7)
test = CreditCard %>% setdiff(train)
x_train = model.matrix(card~., train)[,-1]
x_test = model.matrix(card~., test)[,-1]
y_train = train$card
y_test = test$card

x = model.matrix(card~., CreditCard)[,-1]
y = CreditCard$card
class(y_test)
```

```
## [1] "numeric"
```

```
head(y_test)
```

```
## [1] 0 0 0 0 0 0
```

With this part, I divided half of my data into a testing and training subset.

RIDGE REGRESSION

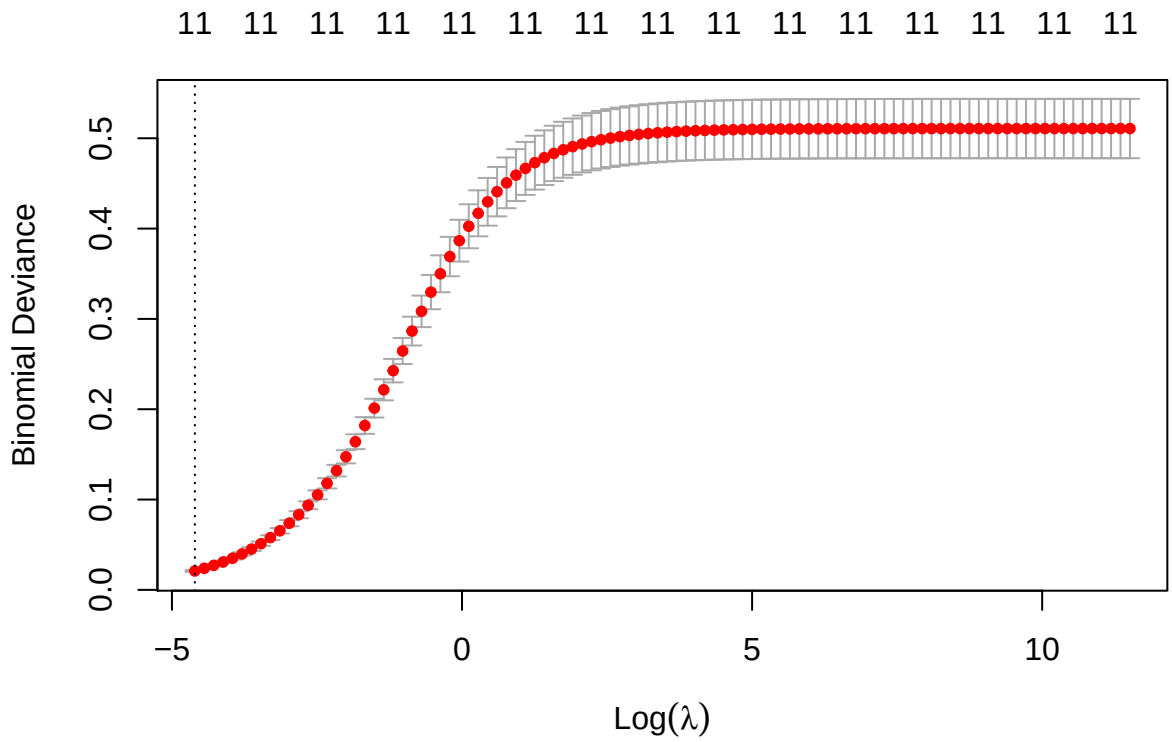
2a.Tuning the model: Cross-Validation to find flexibility of the model

```
grid = 10^seq(5, -2, length = 100)
#grid = c(.1,.2,.3)
ridge_mod = glmnet(x_train, y_train, alpha = 0, lambda = grid, family = "binomial")
cv.out = cv.glmnet(x_train, y_train, alpha = 0, family = "binomial", lambda = grid) # Fit ridge regress

bestlam = cv.out$lambda.1se  # Select lamda that minimizes training MSE
bestlam
```

```
## [1] 0.01
```

```
plot(cv.out)
```



```
#seq(5, -3, length = 100)
```

We see that .01816285 is the value of λ that results inthe smallest cross-validation error.

2b. Choosing lambda within one standard error of minimum lambda

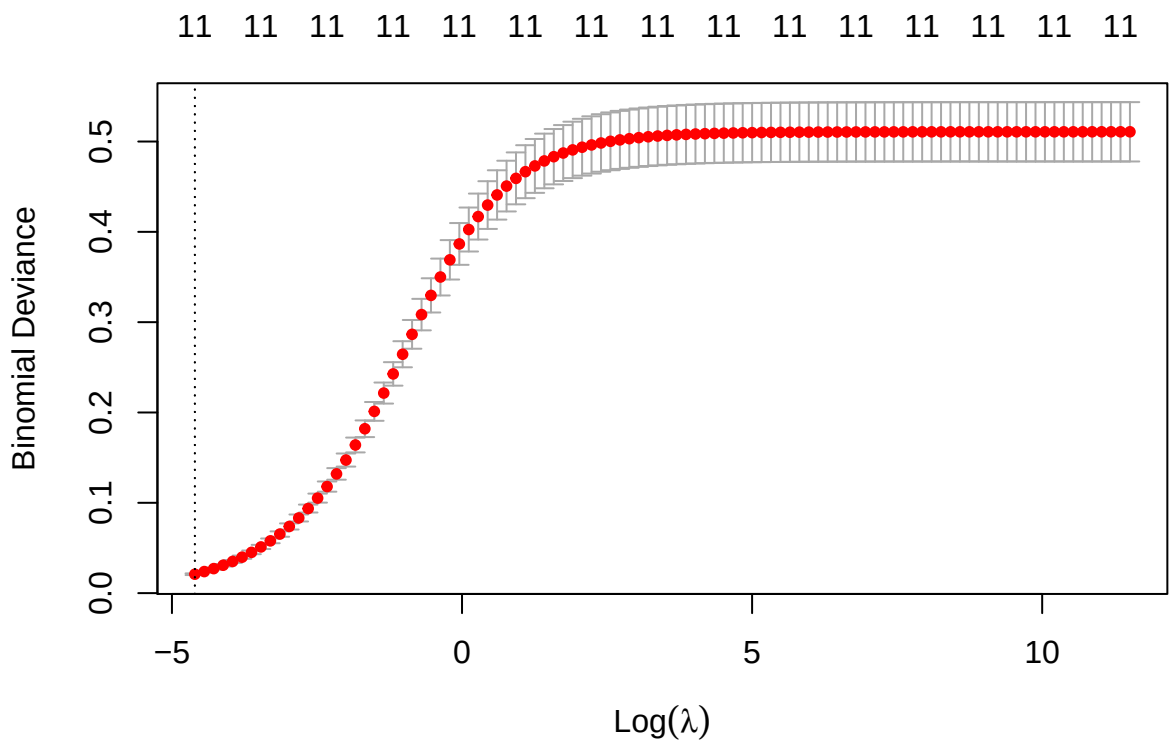
```
#bestlam = cv.out$lambda.min  # Select lamda that minimizes training MSE
bestlam
```

```
## [1] 0.01
```

The lambda I will choose to be one standard error of my minimum lambda will be one standard error away so I will make it .036324.

2c. Show the cross-validation error and the chosen lambda in a graph

```
bestlam = cv.out$lambda.min
plot(cv.out)
```



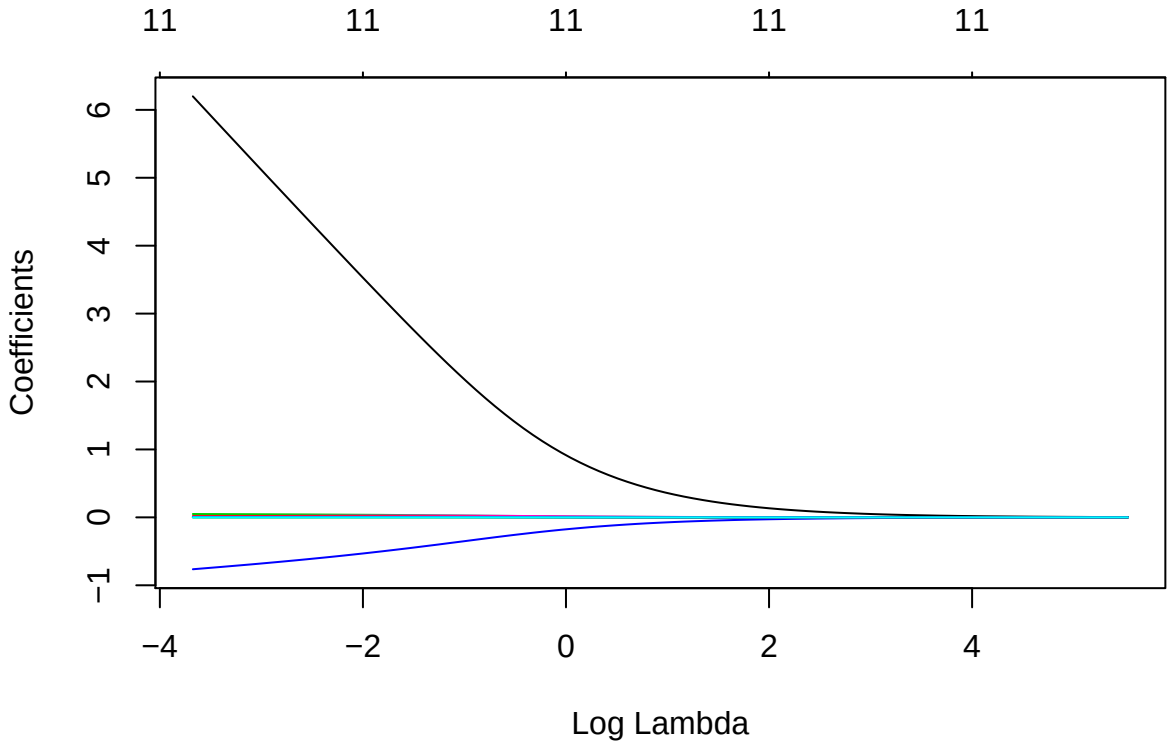
In this graph, it is showing me the values of lambda that results in the smallest cross validation for .036324. This plot will allow me to plot the binomial deviance as a function of lambda.

2d. Show how the coefficients vary with lambda in a graph

```
ridge_pred = predict(ridge_mod, s = bestlam, newx = x_test) # Use best lambda to predict test data
mean((ridge_pred - y_test)^2) # Calculate test MSE
```

```
## [1] 25.71759
```

```
out = glmnet(x, y, alpha = 0, family = "binomial") # Fit ridge regression model on the FULL dataset (tr
plot(out, xvar = "lambda")
```



In this graph, we see that none of the coefficients are exactly zero, which is correct as shown below. We are plotting the coefficients for the different values of lambda. Also, we see that none of the coefficiencts are exactly zero because ridge regression doesnot perform variable selection.

2e. Report the coefficients correspondent with the chosen l

```
predict(out, type = "coefficients", s = bestlam)[1:12,] # Display coefficients using lambda chosen by C
```

```
##   (Intercept)       reports           age        income         share
## -4.907284e+00  6.605044e-03  6.348101e-03  4.934543e-02 -7.638744e-01
##   expenditure         owner       selfemp    dependents        months
## -9.240538e-05  1.466916e-02  6.198754e+00  1.825964e-02  6.001199e-04
##    majorcards        active
##  4.693712e-03  1.767348e-03
```

We see that the coefficient that is the largest is shares so shares has the largest effect in our model.

2f. Report the MSE (Error Rate for classification) in the test subset

```
ridge_mod = glmnet(x_train, y_train, alpha = 0, lambda = bestlam, family = 'binomial')
ridge_pred = predict(ridge_mod, s = bestlam, newx = x_test) # Use best lambda to predict test data
y_pred = ifelse(ridge_pred>0.2,1,0)
table(y_pred, y_test)
```

```
##       y_test
## y_pred   0   1
##      0 368  26
##      1   2   0
```

```
ridge_mod
```

```
##
## Call:  glmnet(x = x_train, y = y_train, family = "binomial", alpha = 0,      lambda = bestlam)
##
##   Df   %Dev Lambda
## 1 11 -4.994   0.01
```

THE accuracy rate was 96.9%. #start of lasso regression

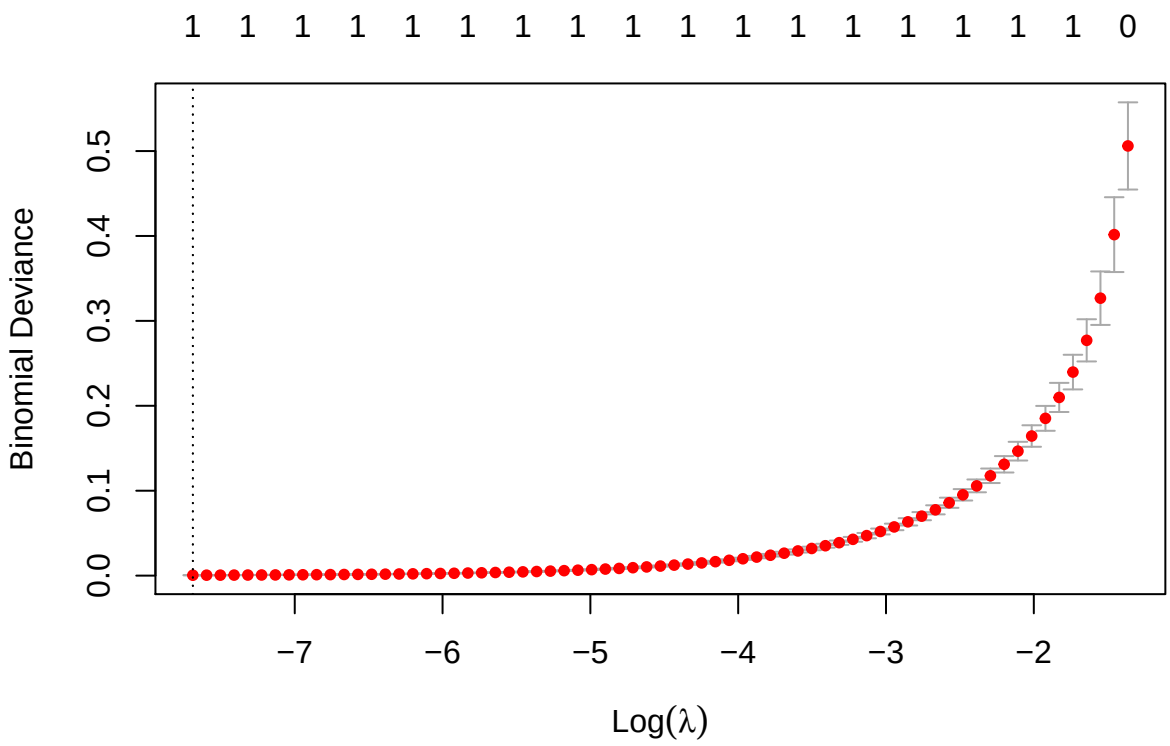
LASSO REGRESSION

2a.Tuning the model: Cross-Validation

```
lasso_mod = glmnet(x_train,
                   y_train,
                   alpha = 1,
                   family="binomial") # Fit lasso model on training data
```

2b. Choosing lambda within one standard error of minimum lambda

```
set.seed(1)
cv.out = cv.glmnet(x_train, y_train, alpha = 1, family="binomial") # Fit lasso model on training data
plot(cv.out) # Draw plot of training MSE as a function of lambda
```

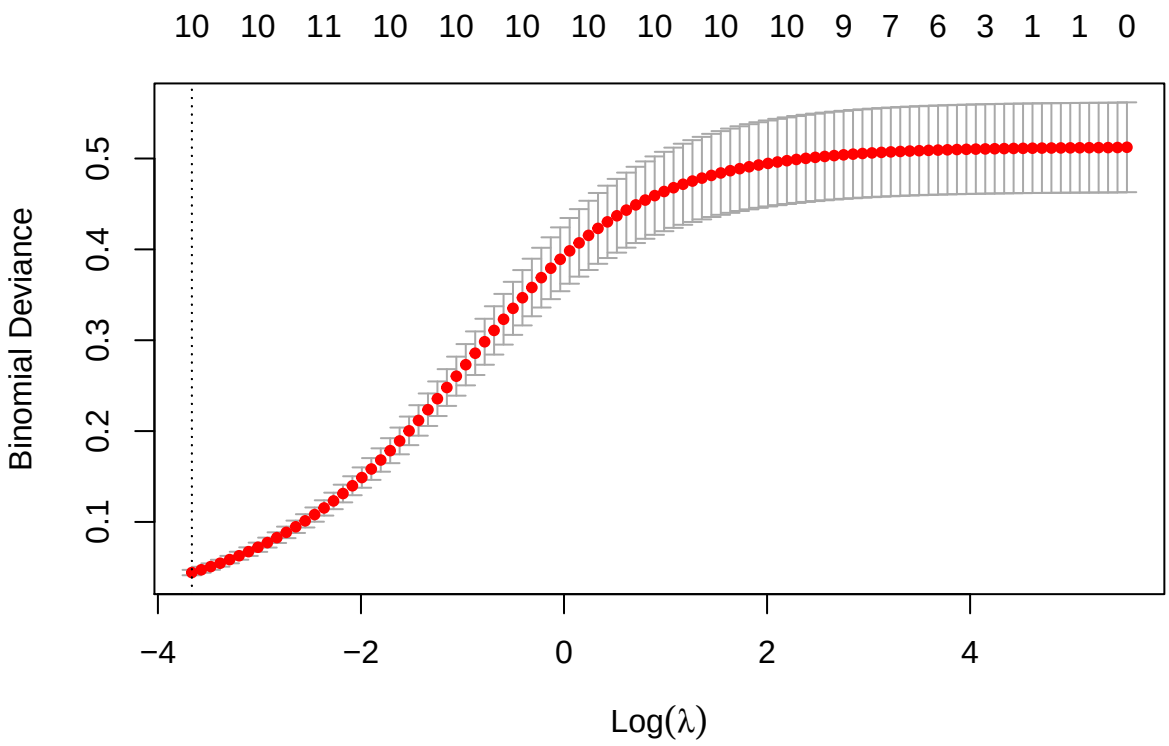



```
bestlam = cv.out$lambda.min # Select lamda that minimizes training MSE
bestlam
```

```
## [1] 0.0004576399
```

2c. Show the cross-validation error and the chosen l in a graph

```
set.seed(1)
cv.out = cv.glmnet(x_train, y_train, alpha = bestlam, family="binomial") # Fit lasso model on training
plot(cv.out) # Draw plot of training MSE as a function of lambda
```



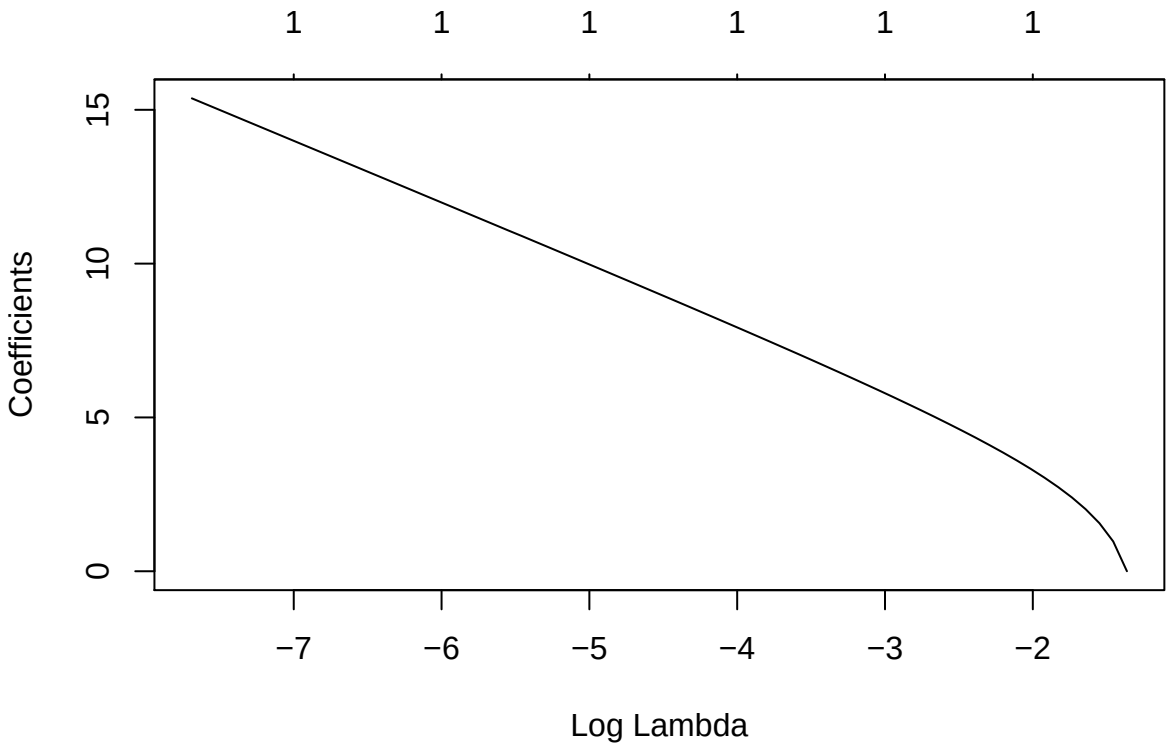
2d. Show how the coefficients vary with l in a graph

```
ls(lasso_mod)
```

```
##  [1] "a0"         "beta"       "call"       "classnames" "dev.ratio"
##  [6] "df"         "dim"        "jerr"       "lambda"     "nobs"
## [11] "npasses"    "nulldev"    "offset"
```

```
plot(lasso_mod,  xvar = "lambda")
```

```
## Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev = x$dev.ratio, : 1
## or less nonzero coefficients; glmnet plot is not meaningful
```



In this graph, we see that there is a lot of It looks like a lot of the coefficients are overlapping each other.

2e. Report the coefficients correspondent with the chosen l

```
out = glmnet(x, y, alpha = 1, lambda = grid, family = "binomial") # Fit lasso model on full dataset
lasso_coef = predict(out, type = "coefficients", s = bestlam)[1:12,] # Display coefficients using lambd
lasso_coef
```

```
## (Intercept)     reports         age      income       share expenditure
##    -5.90339     0.00000     0.00000     0.00000     0.00000     0.00000
##       owner     selfemp  dependents      months  majorcards      active
##     0.00000     9.16979     0.00000     0.00000     0.00000     0.00000
```

```
lasso_coef[lasso_coef != 0] # Display only non-zero coefficients
```

```
## (Intercept)     selfemp
##    -5.90339     9.16979
```

```
bestlam
```

```
## [1] 0.0004576399
```

2f. Report the MSE (Error Rate for classification) in the test subset

3b. Error rate and mse on tree