Homework 2, STATS 315A

Stanford University, Winter 2019

Joe Higgins, Jess Wetstone, Remmelt Ammerlaan

Question 7

7. Obtain the zipcode train and test data from the ESL website.

i. Compare the test performance of a) linear regression b) linear discriminant analysis and c) multiclass linear logistic regression.

```
rm(list = ls())

read_data_as_matrix <- function(file_string){
  table <- read.table(file_string)
  output <- matrix(, nrow=dim(table)[1], ncol=dim(table)[2])
  for(i in 1:ncol(table)){
    output[,i] <- table[,i]
  }
  return(output)
}

get_accuracy <- function(y_hat, y){
  correct <- y_hat == y
  pct_correct <- sum(correct)/length(correct)
  return(pct_correct)
}

#import data
train <- read_data_as_matrix("zip.train")
y_train <- train[,1]
x_train <- train[,2:dim(train)[2]]
test <- read_data_as_matrix("zip.test")
y_test <- test[,1]
x_test <- test[,2:dim(test)[2]]

#encode y_train as one hot matrix
to_one_hot_matrix <- function(vector){
  K <- 10
  one_hot_matrix <- c()
  for(i in 1:length(vector)){
    class <- vector[i]
    new_row <- rep(0,K)
    new_row[class+1] <- 1
    one_hot_matrix <- rbind(one_hot_matrix, new_row)
  }
  return(one_hot_matrix)
}
y_train_1hot <- to_one_hot_matrix(y_train)


#train models
```

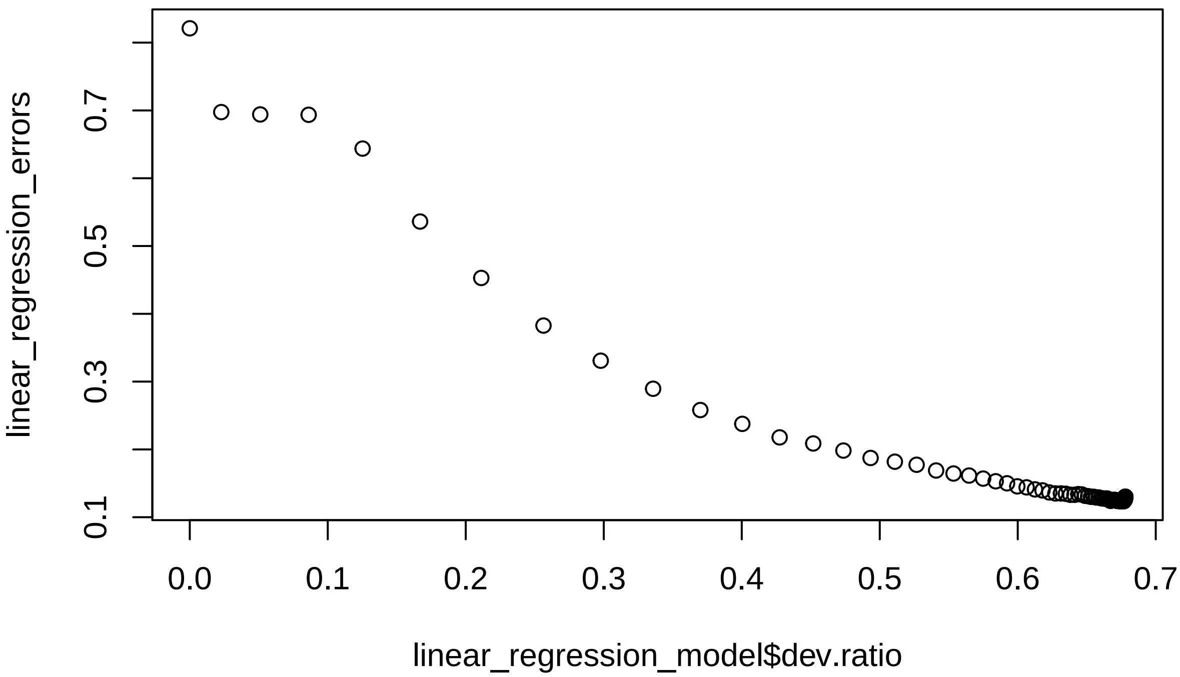
```r
linear_regression_model <- glmnet(
  x_train, y_train_1hot,
  family=c("mgaussian"), alpha = 0.3
)
multinomial_regression_model <- glmnet(
  x_train, y_train,
  family=c("multinomial"), alpha = 0.3
)
lda_model <- lda(
  y_train ~ ., data=data.frame(x_train),
  na.action="na.omit", CV=FALSE
)

#create predictions
linear_regression_output <- predict(
  linear_regression_model, x_test, type=c("response")
)
linear_regression_prdct <- apply(
  linear_regression_output, 3, function (x) apply(
    x, 1, function(y) which.max(y) - 1
  )
)
multinomial_regression_output <- predict(
  multinomial_regression_model, x_test, type=c("response")
)
multinomial_regression_prdct <- apply(
  multinomial_regression_output, 3, function (x) apply(
    x, 1, function(y) which.max(y) - 1
  )
)
lda_output <- predict(lda_model, data.frame(x_test))
lda_prdct <- lda_output$class

#get errors
linear_regression_errors <- apply(
  linear_regression_prdct, 2,
  function(x) 1 - get_accuracy(x, y_test)
)
multinomial_regression_errors <- apply(
  multinomial_regression_prdct, 2,
  function(x) 1 - get_accuracy(x, y_test)
)
lda_error <- 1 - get_accuracy(lda_prdct, y_test)

#report errors
cat("Test performance:\n")
cat(
  "Linear Regression, min error over Lambda:",
  min(linear_regression_errors), "\n"
)
cat(
  "Linear Discriminant Analysis error:",
  lda_error, "\n"
```

```r
)
cat(
  "Multinomial Regression, min error over Lambda:",
  min(multinomial_regression_errors), "\n"
)
```

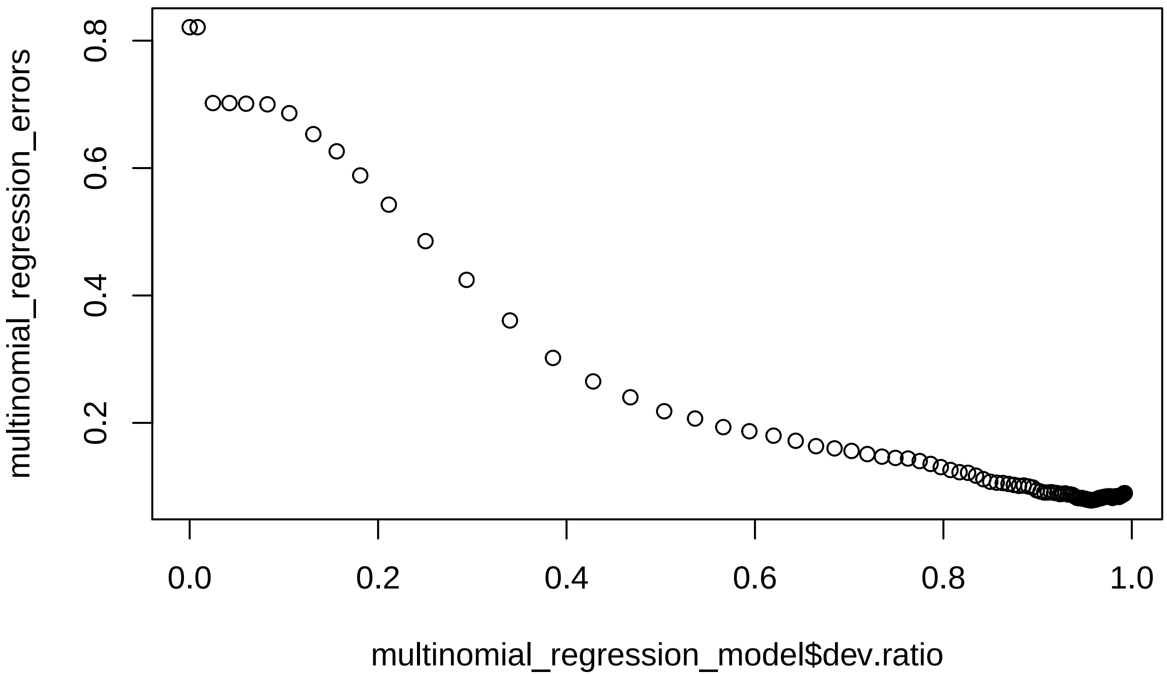
```
## Test performance:
## Linear Regression, min error over Lambda: 0.1235675
## Linear Discriminant Analysis error: 0.1145989
## Multinomial Regression, min error over Lambda: 0.07822621
```

ii. For a) and c), use the package `glmnet` (available in R, matlab and python) to run elastic-net regularized versions of each (use $\alpha = 0.3$). For these two, plot the test error as a functions of the training R^2 for a) and D^2 for c) (% training deviance explained).

```r
plot(linear_regression_model$dev.ratio, linear_regression_errors)
```




```r
plot(multinomial_regression_model$dev.ratio, multinomial_regression_errors)
```



iii. In ii., what is the optimization problem being solved?

$\min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda[\alpha||\beta||_2^2 + (1-\alpha)||\beta||_1]$