STAD80: Assignment 4

Vladislav Trukhin

Due: February 28th, 2022

Contents

Question 1

Question 1.1.a

```
source("/Users/vladislavtrukhin/Downloads/A4_datasets/functions.R")
```

```
pos_col <- readPNG(
  sprintf("/Users/vladislavtrukhin/Downloads/A4_datasets/pngdata/pos/%d.png", sample(1:500, 1)))
neg_col <- readPNG(
  sprintf("/Users/vladislavtrukhin/Downloads/A4_datasets/pngdata/neg/%d.png", sample(1:500, 1)))
writePNG(pos_col, target = "pos_col.png")
writePNG(neg_col, target = "neg_col.png")
```

```
pos_gray <- rgb2gray(pos_col)
neg_gray <- rgb2gray(neg_col)
```

Figure 1: Colored

```
writePNG(pos_gray, target = "pos_gray.png")
writePNG(neg_gray, target = "neg_gray.png")
```

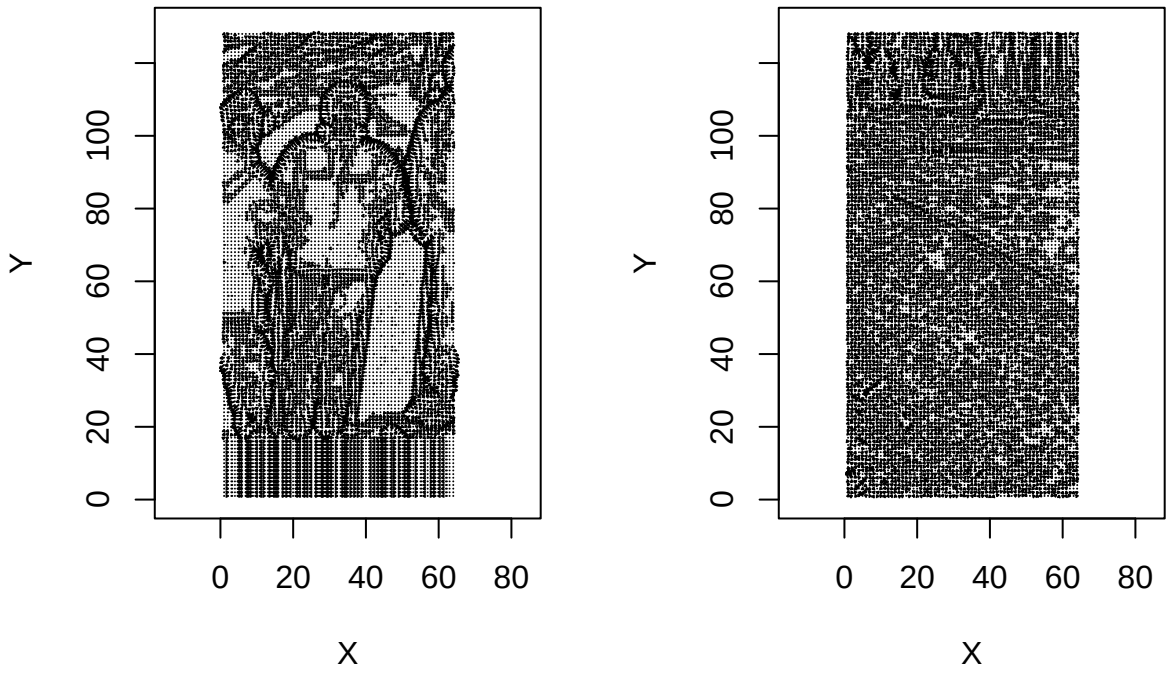
```
neg_crop <- crop.r(neg_gray, 160, 96)
writePNG(neg_crop, target = "neg_crop.png")
```

```
par(mfrow=c(1, 2))

pos_grad <- grad(pos_gray, 128, 64, TRUE)
neg_grad <- grad(neg_crop, 128, 64, TRUE)
```

Figure 2: Grayscale

Figure 3: Cropped




```
pos_fet <- hog(pos_grad[[1]], pos_grad[[2]], 4, 4, 6)
neg_fet <- hog(neg_grad[[1]], neg_grad[[2]], 4, 4, 6)

head(pos_fet, 6)
```

```
## [1] 0.07421875 0.20898438 0.17382812 0.13476562 0.22460938 0.18359375
```

```
head(neg_fet, 6)
```

```
## [1] 0.1992188 0.2089844 0.1640625 0.1328125 0.1230469 0.1718750
```

Question 1.1.b

```
feature <- function(pos, neg) {
  pos_gray <- rgb2gray(pos)
  neg_gray <- rgb2gray(neg)

  neg_crop <- crop.r(neg_gray, 160, 96)

  pos_grad <- grad(pos_gray, 128, 64, FALSE)
  neg_grad <- grad(neg_crop, 128, 64, FALSE)

  pos_fet <- hog(pos_grad[[1]], pos_grad[[2]], 4, 4, 6)
  neg_fet <- hog(neg_grad[[1]], neg_grad[[2]], 4, 4, 6)

  return(list(pos_fet, neg_fet))
}
```

```
fet_data <- c()
pos_data <- c()
for (i in 1:500) {
  pos <- readPNG(
    sprintf("/Users/vladislavtrukhin/Downloads/A4_datasets/pngdata/pos/%d.png", i))
```

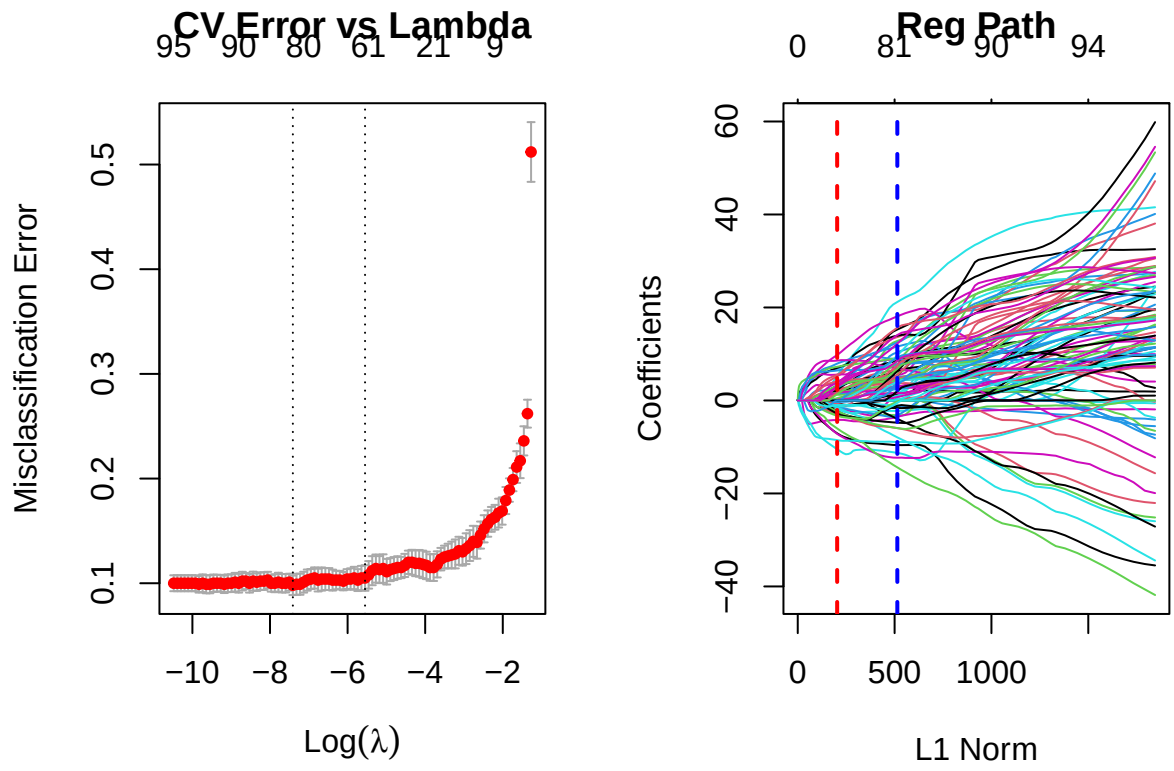
```
  neg <- readPNG(
    sprintf("/Users/vladislavtrukhin/Downloads/A4_datasets/pngdata/neg/%d.png", i))
  fet <- feature(pos, neg)
  fet_data <- rbind(fet_data, fet[[1]], fet[[2]])
  pos_data <- rbind(pos_data, 1, 0)
}
```

Question 1.2

```
par(mfrow=c(1, 2))
fit <- glmnet(fet_data, pos_data, family="binomial", type.measure="class")
cv <- cv.glmnet(fet_data, pos_data, family="binomial", type.measure="class")

plot(cv)
title("CV Error vs Lambda")

plot(fit)
abline(v=sum(abs(coef(cv, s="lambda.min")[2:97])), col="blue", lwd=2, lty=2)
abline(v=sum(abs(coef(cv, s="lambda.1se")[2:97])), col="red", lwd=2, lty=2)
title("Reg Path")
```




```
predicted <- predict(cv, fet_data, s="lambda.min")
misclass <- sum(as.integer(predicted > 0) != pos_data)
misclass # Number misclassified
```

```
## [1] 61
```

```
(500-misclass)/500 # Training accuracy
```

```
## [1] 0.878
```

Question 2

Question 2.a

```
load("/Users/vladislavtrukhin/Downloads/A4_datasets/Amazon_SML.RData")
colnames(dat) # Column names
```

```
## [1] "name"   "review" "rating"
```

```
sum(table(dat$rating)) # Number of reviews
```

```
## [1] 1312
```

```
nrow(as.data.frame(unique(dat$name))) # Number of unique products
```

```
## [1] 20
```

```
count5 <- function(data) {
  return(sum(data$rating == 5))
}

count1 <- function(data) {
  return(sum(data$rating == 1))
}

x <- dlply(dat, .(name), count5)
y <- dlply(dat, .(name), count1)
x[which.max(x)] # Product with most number of 5 ratings and count
```

```
## $`Vulli Sophie the Giraffe Teether`
## [1] 526
```

```
y[which.max(y)] # Product with most number of 1 ratings and count
```

```
## $`Infant Optics DXR-5 2.4 GHz Digital Video Baby Monitor with Night Vision`
## [1] 68
```

Question 2.b

```
# Number of reviews of each rating
sum(dat$rating == 1)
```

```
## [1] 656
```

```
sum(dat$rating == 5)
```

```
## [1] 656
```

The best performance of a constant classifier is 50%, one which assigns a rating 5 for every review.

```
source("/Users/vladislavtrukhin/Downloads/A4_datasets/tdMat.R")
```

```
## Loading required package: NLP
```

Question 2.c

```
source("/Users/vladislavtrukhin/Downloads/A4_datasets/splitData.R")
```

```r
set.seed(10)
lambda <- exp(seq(-20, -1, length.out = 99))
cv <- cv.glmnet(train.x, train.y, family="binomial", type.measure="class", lambda=lambda)

# Number of covariates with non-zero coefficients in the model selected by lambda.1se
sum(coef(cv, lamda="lambda.1se") != 0)
```

```
## [1] 356
```

```r
neg_order <- order(coef(cv, lamda="lambda.1se"), decreasing = FALSE)
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```r
pos_order <- order(coef(cv, lamda="lambda.1se"), decreasing = TRUE)
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```r
head(row.names(coef(cv, lamda="lambda.1se"))[neg_order], 20) # 20 most negative words
```

```
##  [1] "swallow"    "downstair"  "tummi"      "solv"       "dissapoint"
##  [6] "unlink"     "avoid"      "philip"     "bin"        "wast"
## [11] "useless"    "click"      "knock"      "scissor"    "massiv"
## [16] "sad"        "speaker"    "cool"       "return"     "ball"
```

```r
head(row.names(coef(cv, lamda="lambda.1se"))[pos_order], 20) # 20 most positive words
```

```
##  [1] "wimper"       "round"        "endur"        "abov"         "scrape"
##  [6] "whichev"      "love"         "lol"          "neighborhood" "laundri"
## [11] "precious"     "fyi"          "teeth"        "poster"       "grandma"
## [16] "channel"      "sum"          "bet"          "describ"      "result"
```

Question 2.d

```r
most_neg = 0
for (i in row.names(coef(cv, lamda="lambda.1se"))[neg_order]) {
  if (sum(train.x[,i] != 0) > 10) {
    most_neg = i
    break
  }
}
most_neg # Most negative word in more than 10 reviews
```

```
## [1] "wast"
```

```r
most_pos = 0
for (i in row.names(coef(cv, lamda="lambda.1se"))[pos_order]) {
  if (sum(train.x[,i] != 0) > 10) {
    most_pos = i
    break
  }
}
most_pos # Most positive word in more than 10 reviews
```

```
## [1] "love"
```

```r
# Reviews with most negative word in more than 10 reviews and rating 5
sum(dat$rating[train.tag[which(train.x[, most_neg] > 0)]] == 5)
```

```
## [1] 6
```

```
# Reviews with most negative word in more than 10 reviews and rating 1
sum(dat$rating[train.tag[which(train.x[, most_neg] > 0)]] == 1)
```

```
## [1] 80
```

```
# Reviews with most positive word in more than 10 reviews and rating 5
sum(dat$rating[train.tag[which(train.x[, most_pos] > 0)]] == 5)
```

```
## [1] 364
```

```
# Reviews with most positive word in more than 10 reviews and rating 1
sum(dat$rating[train.tag[which(train.x[, most_pos] > 0)]] == 1)
```

```
## [1] 63
```

```
# First review using the most negative word in more than 10 reviews
print(dat$review[train.tag[which(train.x[, most_neg] > 0)[1]]])
```

```
## [1] We have had this monitor for four years now and we are getting ready to purchase another one to u
## 182643 Levels:  ...
```

We have had this monitor for four years now and we are getting ready to purchase another one to use with our second baby that is due soon. Our home is about 4,000 sq. feet and our daughters room is at the other end of the house from ours. We have no problems hearing her perfectly, anywhere in the house. We have never replaced the battery and we take it with us whenever we travel. Do not waste your time or money on any other product, we tried the Graco, Fisher Price, and the Summer and were disappointed. Congratulations on your little one and enjoy hearing those precious sounds through this monitor!

```
# First review using the most positive word in more than 10 reviews
print(dat$review[train.tag[which(train.x[, most_pos] > 0)[1]]])
```

```
## [1] It\\'s easy to hold by little fingers n gives sound when she presses it. She loves it very much a
## 182643 Levels:  ...
```

It\'s easy to hold by little fingers n gives sound when she presses it. She loves it very much and smile every time she sees Sophie.

Question 2.e

```
predict_test <- predict(cv, test.x, s="lambda.1se")
misclass <- sum(as.integer(predict_test > 0) != test.y)
misclass # Number misclassified
```

```
## [1] 9
```

```
(1312-misclass)/1312 # Test accuracy
```

```
## [1] 0.9931402
```

The performance of the logistic model exceeds the constant classifier.

Question 3

Question 3.a

$\ell(\beta) = \ln \Pi_{i=1}^{n} P(Y = y_i | X = x_i)$

$= \ln \Pi_{i=1}^{n} \frac{\lambda(x_i)^{y_i}}{y_i!} e^{-\lambda(x_i)}$

$= \Sigma_{i=1}^{n} y_i \ln \lambda(x_i) - \ln y_i! - \lambda(x_i)$

$= \Sigma_{i=1}^{n} y_i \beta^\top x_i - \ln y_i! - e^{\beta^\top x_i}$

Question 3.b

For Poisson regression

$\frac{\partial \ell(\beta)}{\partial \beta} = \Sigma_{i=1}^{n} \frac{\partial}{\partial \beta} y_i \beta^\top x_i - \frac{\partial}{\partial \beta} \ln y_i! - \frac{\partial}{\partial \beta} e^{\beta^\top x_i}$

$= \Sigma_{i=1}^{n} y_i x_i - e^{\beta^\top x_i} x_i$

At $\beta = \hat{\beta}$

$\Sigma_{i=1}^{n} y_i x_i - e^{\hat{\beta}^\top x_i} x_i = 0$

$\Rightarrow \Sigma_{i=1}^{n} y_i x_i = \Sigma_{i=1}^{n} e^{\hat{\beta}^\top x_i} x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i x_i = \Sigma_{i=1}^{n} \lambda(x_i) x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i x_i = \Sigma_{i=1}^{n} E_{\hat{\beta}}[Y|X = x_i] x_i$

For logistic regression

$\frac{\partial \ell(\beta)}{\partial \beta} = -\Sigma_{i=1}^{n} (1 - y_i) \frac{\partial}{\partial \beta} \ln(1 + e^{\beta x_i}) - \Sigma_{i=1}^{n} y_i \frac{\partial}{\partial \beta} \ln(1 + e^{-\beta x_i})$

$= -\Sigma_{i=1}^{n} (1 - y_i) \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i + \Sigma_{i=1}^{n} y_i \frac{e^{-\beta x_i}}{1+e^{-\beta x_i}} x_i$

At $\beta = \hat{\beta}$

$-\Sigma_{i=1}^{n} (1 - y_i) \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i + \Sigma_{i=1}^{n} y_i \frac{e^{-\beta x_i}}{1+e^{-\beta x_i}} x_i = 0$

$\Rightarrow \Sigma_{i=1}^{n} y_i \frac{e^{-\beta x_i}}{1+e^{-\beta x_i}} x_i = \Sigma_{i=1}^{n} (1 - y_i) \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i \frac{1}{1+e^{\beta x_i}} x_i = \Sigma_{i=1}^{n} \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i - y_i \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i \frac{1}{1+e^{\beta x_i}} x_i + y_i \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i = \Sigma_{i=1}^{n} \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i (\frac{1}{1+e^{\beta x_i}} x_i + \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i) = \Sigma_{i=1}^{n} \frac{e^{\beta x_i}}{1+e^{\beta x_i}} x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i \frac{1+e^{\beta x_i}}{1+e^{\beta x_i}} x_i = \Sigma_{i=1}^{n} \theta(x_i) x_i$

$\Rightarrow \Sigma_{i=1}^{n} y_i x_i = \Sigma_{i=1}^{n} E_{\hat{\beta}}[Y|X = x_i] x_i$

Question 4

Question 4.a

$\ell(\beta) = \ln \Pi_{i=1}^{n} P(Y = y_i | X = x_i)$

$= \ln \Pi_{i=1}^{n} (1 - \eta(x_i))^{1-y_i} \eta(x_i)^{y_i}$

$= \Sigma_{i=1}^{n} (1 - y_i) \ln(1 - \eta(x_i)) + y_i \ln \eta(x_i)$

$= \Sigma_{i=1}^{n} (1 - y_i) \ln(1 - \eta(x_i)) + y_i \ln \eta(x_i)$

$= \Sigma_{i=1}^{n} (1 - y_i) \ln(\frac{1}{1+e^{\beta x_i}}) + y_i \ln(\frac{1}{1+e^{-\beta x_i}})$

$= -\Sigma_{i=1}^{n} (1 - y_i) \ln(1 + e^{\beta x_i}) - \Sigma_{i=1}^{n} y_i \ln(1 + e^{-\beta x_i})$

Question 4.b

$-\Sigma_{i=1}^{n}(1-y_i)\ln(1+e^{\beta x_i}) - \Sigma_{i=1}^{n} y_i \ln(1+e^{-\beta x_i})$

$\forall x_i \leq 0, y_i = 0$

$\Rightarrow -\Sigma_{i=1}^{n}(1-y_i)\ln(1+e^{\beta x_i}) = -\Sigma_{i=1,y_i=0}^{n}\ln(1+e^{-\beta|x_i|})$

$\forall x_i \geq 0, y_i = 1$

$\Rightarrow -\Sigma_{i=1}^{n} y_i \ln(1+e^{-\beta x_i}) = -\Sigma_{i=1,y_i=1}^{n}\ln(1+e^{-\beta|x_i|})$

$\Rightarrow -\Sigma_{i=1}^{n}(1-y_i)\ln(1+e^{\beta x_i}) - \Sigma_{i=1}^{n} y_i \ln(1+e^{-\beta x_i})$

$= -\Sigma_{i=1,y_i=0}^{n}\ln(1+e^{-\beta|x_i|}) - \Sigma_{i=1,y_i=1}^{n}\ln(1+e^{-\beta|x_i|})$

$= -\Sigma_{i=1}^{n}\ln(1+e^{-\beta|x_i|})$

$\Rightarrow argmax_\beta - \Sigma_{i=1}^{n}\ln(1+e^{-\beta|x_i|})$

$= argmin_\beta \Sigma_{i=1}^{n}\ln(1+e^{-\beta|x_i|})$

$= argmin_\beta e^{-\beta}$

$= argmax_\beta \beta = \infty = \hat{\beta}$

Question 5

Question 5.1

```
X <- read.csv("/Users/vladislavtrukhin/Downloads/A4_datasets/framingham.csv")
X <- na.omit(X)
X.all <- scale(X[,-16])
Y.all <- X[,16]

fit <- glm(TenYearCHD~male+age+education+currentSmoker+cigsPerDay+BPMeds
           +prevalentStroke+prevalentHyp+diabetes+totChol+sysBP+diaBP
           +BMI+heartRate+glucose, family=binomial, data=X)
summary(fit)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ male + age + education + currentSmoker +
##     cigsPerDay + BPMeds + prevalentStroke + prevalentHyp + diabetes +
##     totChol + sysBP + diaBP + BMI + heartRate + glucose, family = binomial,
##     data = X)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9582  -0.5939  -0.4264  -0.2829   2.8409
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -8.328186   0.715449 -11.641  < 2e-16 ***
## male             0.555279   0.109033   5.093 3.53e-07 ***
## age              0.063515   0.006679   9.509  < 2e-16 ***
## education       -0.047767   0.049395  -0.967  0.33353
## currentSmoker    0.071601   0.156752   0.457  0.64783
## cigsPerDay       0.017914   0.006238   2.872  0.00408 **
```

```
## BPMeds           0.162496   0.234326   0.693  0.48802
## prevalentStroke  0.693660   0.489569   1.417  0.15652
## prevalentHyp     0.234208   0.138026   1.697  0.08973 .
## diabetes         0.039167   0.315506   0.124  0.90120
## totChol          0.002332   0.001127   2.070  0.03850 *
## sysBP            0.015403   0.003808   4.044 5.24e-05 ***
## diaBP           -0.004159   0.006438  -0.646  0.51831
## BMI              0.006672   0.012758   0.523  0.60097
## heartRate       -0.003246   0.004211  -0.771  0.44082
## glucose          0.007127   0.002234   3.190  0.00142 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3121.2  on 3657  degrees of freedom
## Residual deviance: 2754.5  on 3642  degrees of freedom
## AIC: 2786.5
##
## Number of Fisher Scoring iterations: 5
```

The variables male, age, cigsPerDay, totChol, sysBP, and glucose are statistically significant with p-val $<$ 0.05.

Question 5.2

```
set.seed(100)

total <- 1:nrow(X)

# 1/5 Test
test_idx <- sample(total, nrow(X)/5)
test <- X[test_idx, ]

# 4/5 Train
train_idx <- total[!total%in%test_idx]
train <- X[train_idx, ]

fit <- glm(TenYearCHD~male+age+education+currentSmoker+cigsPerDay+BPMeds
           +prevalentStroke+prevalentHyp+diabetes+totChol+sysBP+diaBP
           +BMI+heartRate+glucose, family=binomial, data=train)

predicted <- predict.glm(fit, test)
misclass <- sum(as.integer(predicted > 0) != test$TenYearCHD)
misclass # Misclassification error for test
```

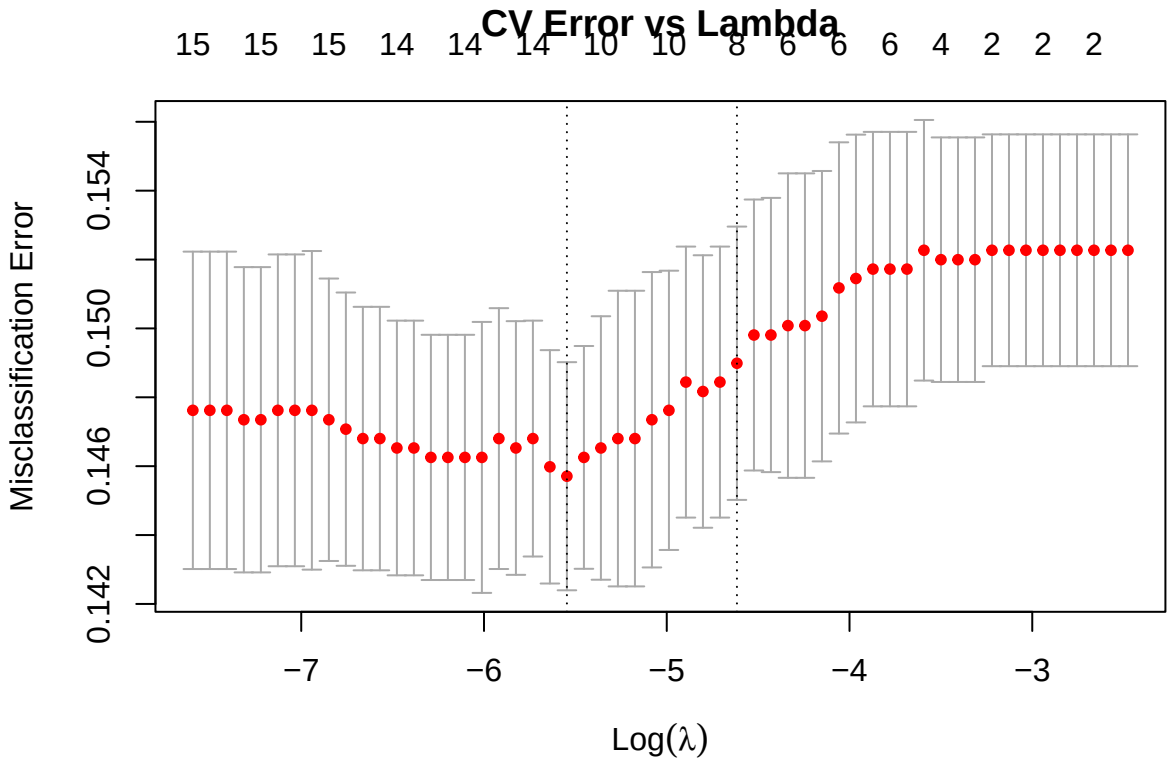
```
## [1] 97
```

```
(nrow(test)-misclass)/nrow(test) # Test accuracy
```

```
## [1] 0.8673051
```

Question 5.3

```
cv <- cv.glmnet(X.all, Y.all, family="binomial", type.measure="class", alpha=1, nfolds=5)
plot(cv)
title("CV Error vs Lambda")
```



No, the shape is not a typical U-curve but does go down a little initially and up a little afterwards. Regularization is most likely not needed in this problem.