ANLY512_HW2

Hongyang Zheng

2019/2/10

Problem ch.4 #6

a)

$$p = \frac{e^{\beta_0+\beta_1*X_1+\beta_2*X_2}}{1+e^{\beta_0+\beta_1*X_1+\beta_2*X_2}}$$
$$= \frac{e^{-6+0.05*40+1*3.5}}{1+e^{-6+0.05*40+1*3.5}}$$
$$= 0.3775407$$

The probability this student gets an A is 0.3775407.

b)

$$0.5 = \frac{e^{-6+0.05*x+3.5}}{1+e^{-6+0.05*x+3.5}}$$

By solving the function, we get $e^{-6+0.05*x+3.5} = 1$ and therefore, $x = 50$

To get a 50% probability that this student gets an A, this student should study 50 hours.

Problem ch.4 #10abcd

a)

```
library(ISLR)
# Numerical summary
summary(Weekly)
```

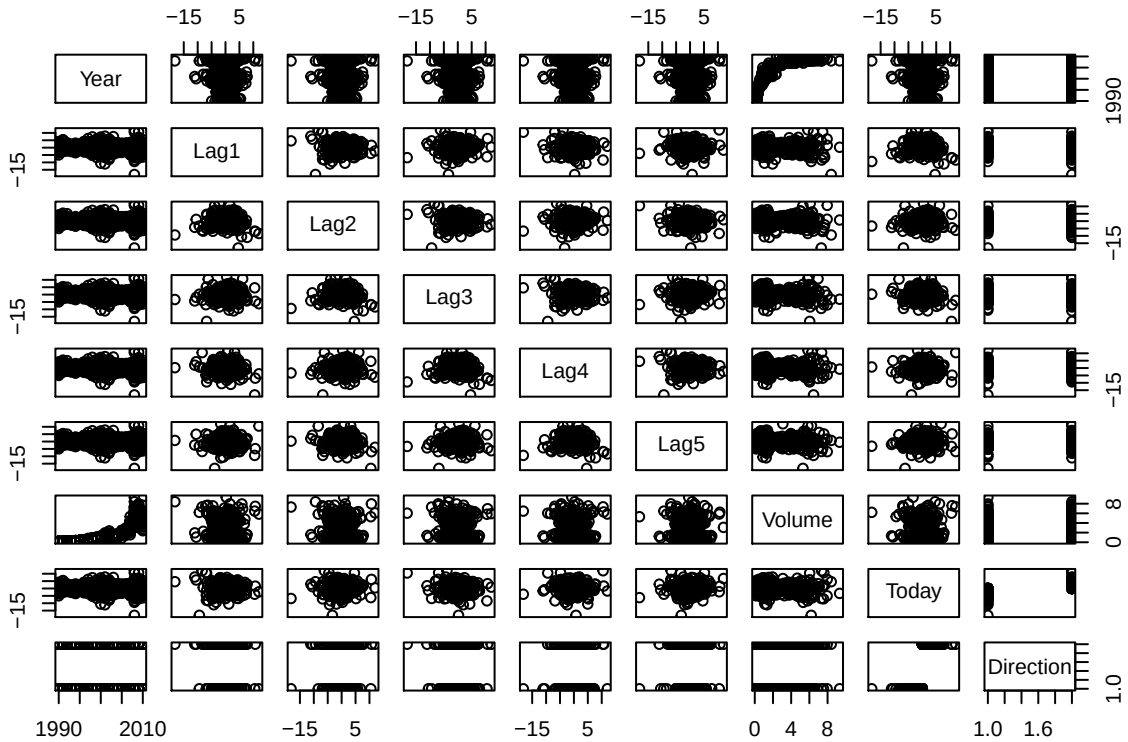
```
##       Year           Lag1               Lag2               Lag3         
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
##       Lag4               Lag5              Volume       
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202  
##  Median :  0.2380   Median :  0.2340   Median :1.00268  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821  
##      Today          Direction 
##  Min.   :-18.1950   Down:484  
```

```
##  1st Qu.: -1.1540   Up   :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

```
cor(Weekly[, -9])
```

```
##                Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

```
# Graphical summary
pairs(Weekly)
```



It seems that 'Volume' and 'Year' have a relationship.

b)

```
glm.1=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Weekly, family = binomial)
summary(glm.1)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

'Lag2' is statistically significant since its coefficient has a small p-value.

c)

```
# Predict the value
glm.1.prob=predict(glm.1, type='response')
n1=length(glm.1.prob)
glm.1.pred=rep('Down', n1)
glm.1.pred[glm.1.prob>0.5]='Up'

# Confusion matrix
table(glm.1.pred, Weekly$Direction)
```

```
##
## glm.1.pred Down  Up
##       Down   54  48
##       Up    430 557
```

```
# Overall fraction of correct predictions
mean(glm.1.pred==Weekly$Direction)
```

```
## [1] 0.5610652
```

The overall fraction of correct predictions is about 56.1%. The confusion matrix also tells us the false positive rate and false negative rate rate.

d)

```
# Separate the data into train and test
train.data=Weekly[Weekly$Year<2009,]
test.data=Weekly[Weekly$Year>=2009,]

# Build new model
glm.2=glm(Direction~Lag2, data=train.data, family='binomial')

# Predict value from the new model
glm.2.prob=predict(glm.2, test.data, type='response')
n2=length(glm.2.prob)
glm.2.pred=rep('Down', n2)
glm.2.pred[glm.2.prob>0.5]='Up'

# Confusion matrix
table(glm.2.pred, test.data$Direction)
```

```
##
## glm.2.pred Down Up
##       Down    9  5
##       Up     34 56
```

```
# Overall fraction of correct predictions
mean(glm.2.pred==test.data$Direction)
```

```
## [1] 0.625
```

The overall fraction of correct predictions is about 62.5%

Problem Xtra #23

a)

```
# Generate Purchase01
np=length(OJ$Purchase)
OJ$Purchase01=rep(0,np)
OJ$Purchase01[OJ$Purchase=='MM']=1
OJ01=OJ[,-1]

# Fit a model
fit.22a=glm(Purchase01~., data=OJ01, family = 'binomial')
summary(fit.22a)
```

```
##
## Call:
## glm(formula = Purchase01 ~ ., family = "binomial", data = OJ01)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7811  -0.5426  -0.2327   0.5304   2.7894
##
```

```
## Coefficients: (5 not defined because of singularities)
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.15806    2.02648   2.545  0.01092 *
## WeekofPurchase  -0.01181    0.01080  -1.093  0.27442
## StoreID         -0.17089    0.13847  -1.234  0.21716
## PriceCH          4.58650    1.81386   2.529  0.01145 *
## PriceMM         -3.62495    0.90259  -4.016 5.92e-05 ***
## DiscCH          10.79673   18.60661   0.580  0.56174
## DiscMM          26.46155    9.08497   2.913  0.00358 **
## SpecialCH        0.26723    0.34207   0.781  0.43468
## SpecialMM        0.31693    0.27307   1.161  0.24579
## LoyalCH         -6.30227    0.39834 -15.821  < 2e-16 ***
## SalePriceMM           NA         NA      NA       NA
## SalePriceCH           NA         NA      NA       NA
## PriceDiff             NA         NA      NA       NA
## Store7Yes        0.31128    0.71681   0.434  0.66411
## PctDiscMM      -50.69763   19.01208  -2.667  0.00766 **
## PctDiscCH      -27.33993   35.17272  -0.777  0.43698
## ListPriceDiff         NA         NA      NA       NA
## STORE                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.9  on 1069  degrees of freedom
## Residual deviance:  816.6  on 1057  degrees of freedom
## AIC: 842.6
##
## Number of Fisher Scoring iterations: 5
```

After looking at the explanation of these variables, I found that SalePriceMM, SalePriceCH, PriceDiff are related to other variables which have coefficient: SalePriceCH=PriceCH-DiscCH, SalePriceMM=PriceMM-DiscMM, and PriceDiff=PriceCH-DiscCH-PriceMM-DiscMM. Therefore, the coefficient for these three variables are NAs.

For ListPriceDiff, which is equal to PriceCH-PriceMM, since the two prices have been included in the model, the coefficient for ListPriceDiff is NA.

For STORE, maybe StoreID contains the same information, so it is NA.

b)

```
# Fit a new model
fit.22b=glm(Purchase01~.-PriceDiff-SalePriceCH-SalePriceMM-ListPriceDiff-STORE, data=OJ01, family = 'bi
summary(fit.22b)
```

```
##
## Call:
## glm(formula = Purchase01 ~ . - PriceDiff - SalePriceCH - SalePriceMM -
##     ListPriceDiff - STORE, family = "binomial", data = OJ01)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7811  -0.5426  -0.2327   0.5304   2.7894
##
```

```
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.15806    2.02648   2.545  0.01092 *
## WeekofPurchase  -0.01181    0.01080  -1.093  0.27442
## StoreID         -0.17089    0.13847  -1.234  0.21716
## PriceCH          4.58650    1.81386   2.529  0.01145 *
## PriceMM         -3.62495    0.90259  -4.016 5.92e-05 ***
## DiscCH          10.79673   18.60661   0.580  0.56174
## DiscMM          26.46155    9.08497   2.913  0.00358 **
## SpecialCH        0.26723    0.34207   0.781  0.43468
## SpecialMM        0.31693    0.27307   1.161  0.24579
## LoyalCH         -6.30227    0.39834 -15.821  < 2e-16 ***
## Store7Yes        0.31128    0.71681   0.434  0.66411
## PctDiscMM      -50.69763   19.01208  -2.667  0.00766 **
## PctDiscCH      -27.33993   35.17272  -0.777  0.43698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.9  on 1069  degrees of freedom
## Residual deviance:  816.6  on 1057  degrees of freedom
## AIC: 842.6
##
## Number of Fisher Scoring iterations: 5
```

After removing those variables, there is no change for the model.

c)

From b) we know that PriceCH, PriceMM, DiscMM, LoyalCH and PctDiscMM is statistically significant.

```
# Fit a new model
fit.22c=glm(Purchase01~PriceCH+PriceMM+DiscMM+LoyalCH+PctDiscMM, data=OJ01, family = 'binomial')
summary(fit.22c)
```

```
##
## Call:
## glm(formula = Purchase01 ~ PriceCH + PriceMM + DiscMM + LoyalCH +
##     PctDiscMM, family = "binomial", data = OJ01)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6394  -0.5800  -0.2564   0.5634   2.8592
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.5773     1.7465   3.194  0.00141 **
## PriceCH       2.7315     1.1199   2.439  0.01472 *
## PriceMM      -3.8818     0.8313  -4.669 3.02e-06 ***
## DiscMM       25.1929     8.3831   3.005  0.00265 **
## LoyalCH      -6.3725     0.3814 -16.706  < 2e-16 ***
## PctDiscMM   -47.9810    17.5153  -2.739  0.00616 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.85  on 1069  degrees of freedom
## Residual deviance:  855.19  on 1064  degrees of freedom
## AIC: 867.19
##
## Number of Fisher Scoring iterations: 5
```

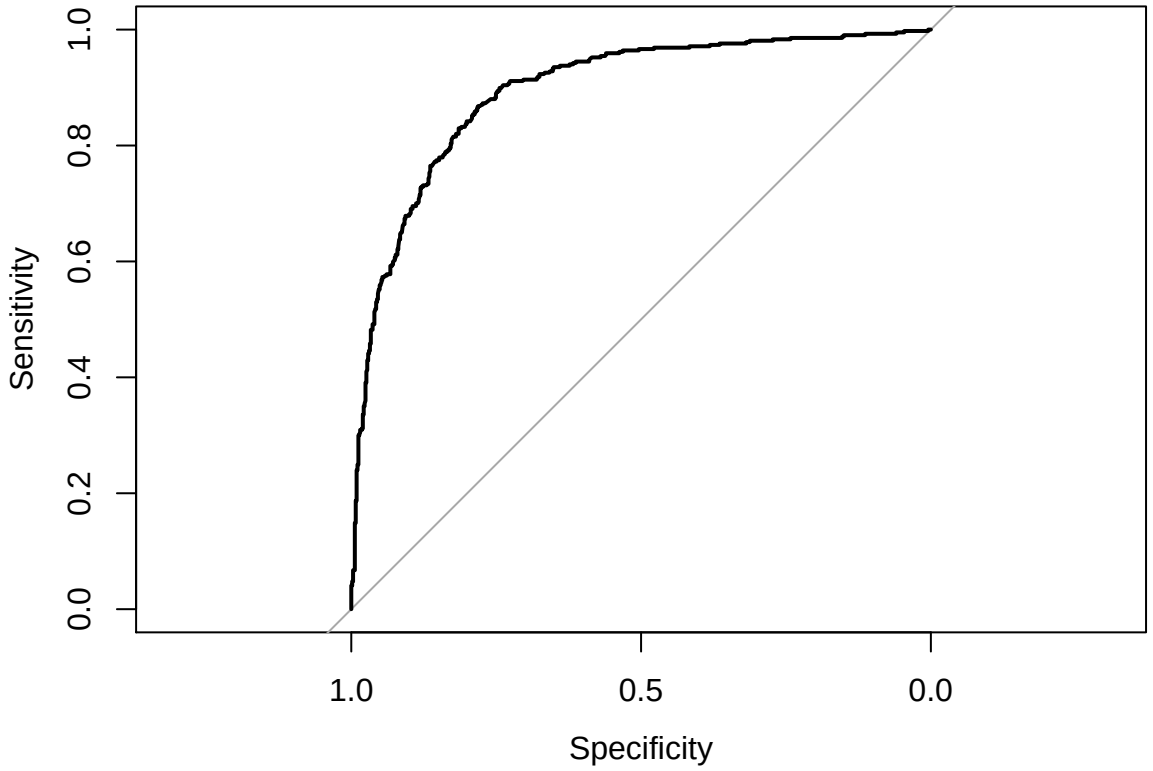
```
# ROC curve
library("pROC")
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
pred.22c=predict(fit.22c, OJ01)
plot(roc(OJ$Purchase01, pred.22c))
```




```
area=auc(roc(OJ$Purchase01, pred.22c))
print(area)
```

```
## Area under the curve: 0.8941
```

The area under the curve is 0.8941, which is approximately 0.89.

d)

The odd is:

$$odd = e^{5.5773+2.7315*PriceCH-3.8818*PriceMM+25.1929*DiscMM-6.3725*LoyalCH-47.9810*PctDiscMM}$$

If the price of Minute Maid is decreased by 0.01, then the odds times $e^0.038818 = 4\%$.

If the price of Citrus Hill is increased by 0.01, then the odds times $e^0.027315 = 2.7\%$.

If the discount offered for Minute Maid is increased by 0.01, then the odds times $e^0.251929 = 28.7\%$.

Problem Xtra #25

```
# Load data
load("~/Desktop/other/Data/mnist_all.RData")

# Extract data for digit3 and digit5
y.nist = train$y
index <- (y.nist == 3 | y.nist == 5)
x.nist <- train$x[index,]
x.train <- as.data.frame(x.nist)

y.nist1 = test$y
index1 <- (y.nist1 == 3 | y.nist1 == 5)
x.nist1 <- test$x[index1,]
x.test <- as.data.frame(x.nist1)

# y=1 if it is digit5, y=0 if it is digit3
x.train$y <- 0
x.train$y[y.nist[index] == 5] <- 1
x.test$y <- 0
x.test$y[y.nist1[index1] == 5] <- 1
```

```
# Find a variable with large variance
# Calculate all variance for V1 to V784
variance=rep(0,784)
for (i in 1:784)
{
  variance[i]=var(x.train[,i])
}
variance=as.data.frame(variance)
variance$index=seq(1:784)

# Sort from biggest to smallest
variance_sort=variance[order(variance[,1], decreasing=TRUE),]
```

```
# Use V429 with variance = 5919.911
fit.v429=glm(y~V429, data=x.train, family = 'binomial')
summary(fit.v429)
```

```
##
## Call:
## glm(formula = y ~ V429, family = "binomial", data = x.train)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -1.588  -1.056  -1.056   1.304   1.304
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.2925354  0.0207901  -14.07   <2e-16 ***
## V429         0.0047854  0.0002595   18.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15971  on 11551  degrees of freedom
## Residual deviance: 15604  on 11550  degrees of freedom
## AIC: 15608
##
## Number of Fisher Scoring iterations: 4
```

The model is:

$$P(\text{the digit is 5}) = \frac{e^{-0.2925354+0.0047854*V429}}{1+e^{-0.2925354+0.0047854*V429}}$$

Determine the fraction of true positives on the test set if the fraction of false positives on the training set is kept to 0.1

```
set.seed(143)
# Confusion matrix
# When threshold=0.5, the FP is close to 0.1.
pred.v429.train=predict(fit.v429, x.train, type='response')
pred.train.429 <- pred.v429.train > 0.5
result=table(x.train$y, pred.train.429)
FP=result[1,2]/(result[1,2]+result[1,1])
print(FP)
```

```
## [1] 0.1189039
```

```
# To find more close FP rate, use this for loop
for (x in seq(0.5,0.6,by=0.001))
{
  pred.train.429 <- pred.v429.train > x
  result=table(x.train$y, pred.train.429)
  FP=result[1,2]/(result[1,2]+result[1,1])
  if (round(FP,3)==0.100)
  {
    print(FP)
    print(x)
  }
}
```

```
## [1] 0.100473
## [1] 0.544
## [1] 0.100473
## [1] 0.545
```

```
# When x=0.544, the training set has a FP rate about 0.1
# Use this threshold for test data
pred.v429.test=predict(fit.v429, x.test, type='response')
```

```r
pred.test.429 <- pred.v429.test > 0.544
table(x.test$y, pred.test.429)
```

```
##    pred.test.429
##     FALSE TRUE
##   0   930   80
##   1   736  156
```

```r
# True positive rate
156/(156+736)
```

```
## [1] 0.1748879
```

Problem Xtra #26

```r
# Print the top 10 largest variance index
print(variance_sort[1:10,2])
```

```
##  [1] 353 325 180 187 216 324 403 382 243 208
```

```r
# After random trying, found V216, V325 have a small correlation
cor(x.train$V325,x.train$V216)
```

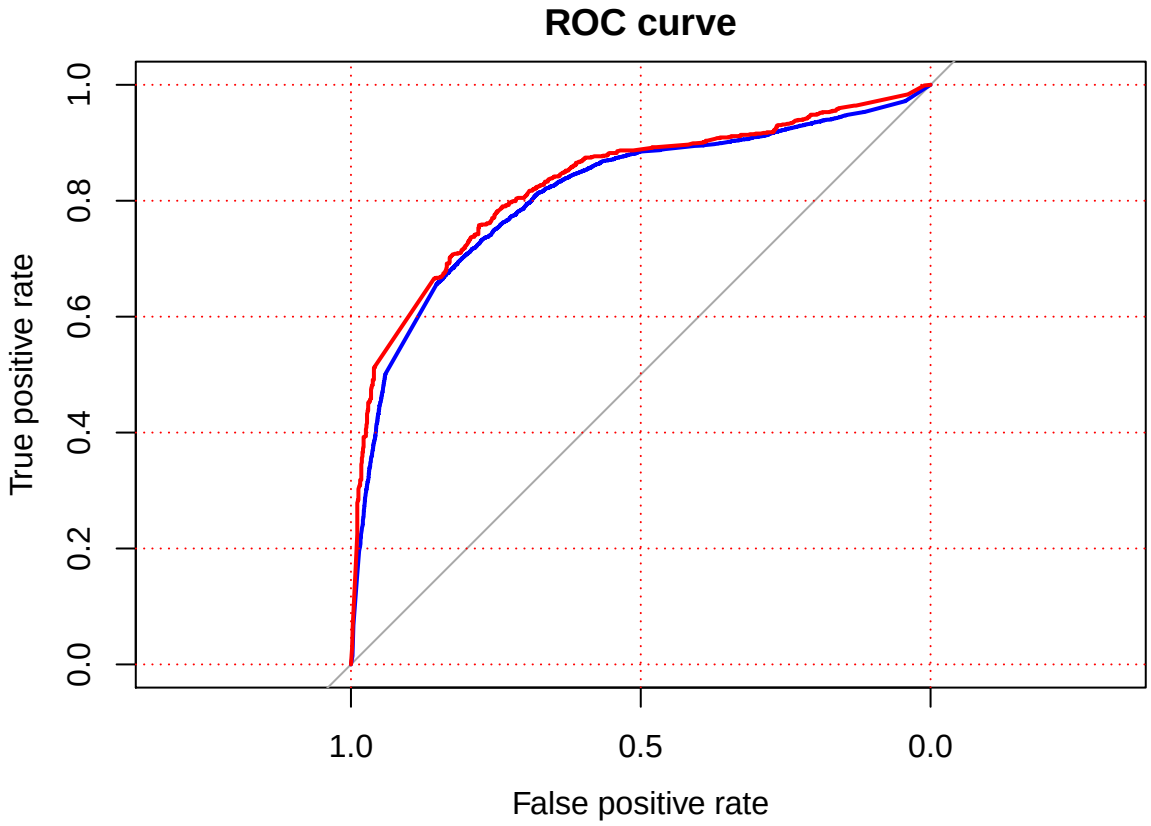
```
## [1] -0.01854714
```

```r
# Fit a model
fit.v325=glm(y~V325+V216, data=x.train, family='binomial')
summary(fit.v325)
```

```
##
## Call:
## glm(formula = y ~ V325 + V216, family = "binomial", data = x.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7717  -0.6657  -0.5369   0.7950   2.0138
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.9520545  0.0371573   25.62  < 2e-16 ***
## V325        -0.0111315  0.0002091  -53.23  < 2e-16 ***
## V216         0.0015064  0.0002001    7.53 5.08e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15971  on 11551  degrees of freedom
## Residual deviance: 12359  on 11549  degrees of freedom
## AIC: 12365
##
## Number of Fisher Scoring iterations: 4
```

```r
# ROC curve
pred.v325.train=predict(fit.v325, x.train, type='response')
```

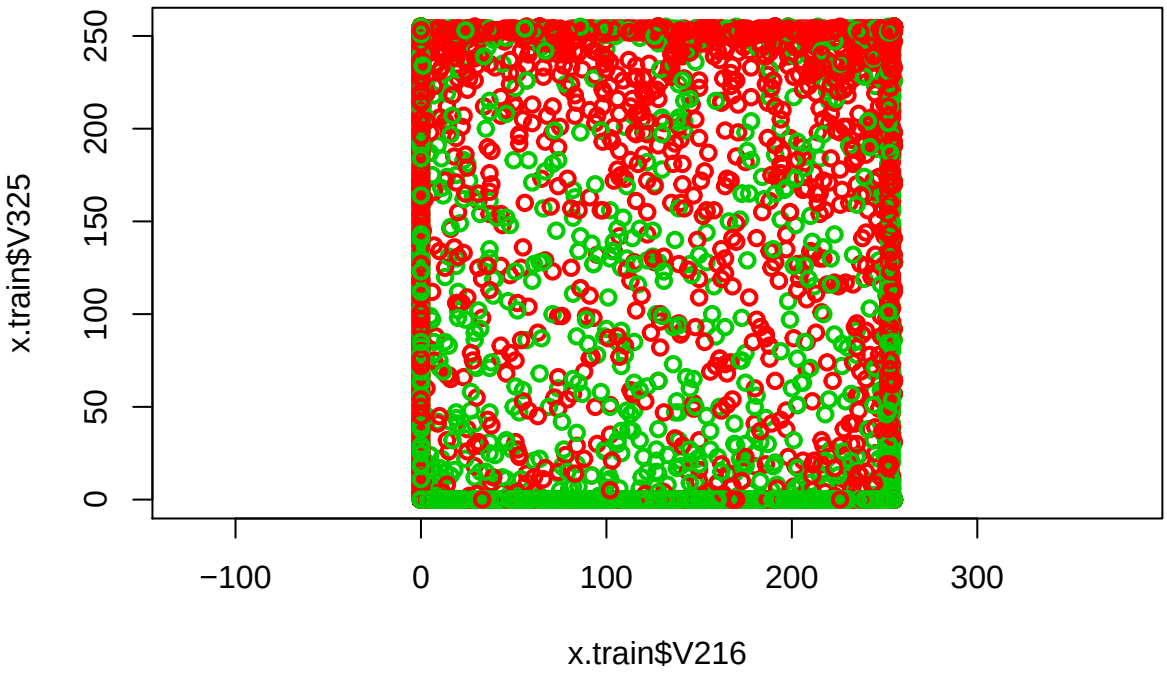
```r
pred.v325.test=predict(fit.v325, x.test, type='response')
plot(roc(x.train$y, pred.v325.train), col=4, xlab='False positive rate', ylab='True positive rate', mai
lines(roc(x.test$y, pred.v325.test), col = 2)
grid(col=2)
```




```r
# Find the auc rea
auc.train=auc(roc(x.train$y, pred.v325.train), col=4)
auc.test=auc(roc(x.test$y, pred.v325.test), col=2)
print(c(auc.train, auc.test))
```

```
## [1] 0.8124459 0.8302691
```

```r
# Scatterplot
plot(x.train$V325 ~ x.train$V216, data = x.train, col = x.train$y+2, lwd = 2, asp = 1)
```



From the scatter plot we can see that the boundary between red and green is not so clear, but there is a tendency that red points are located at the top while green points are located at the bottom. Therefore, with test auc=0.8124459, I think this classifier performs good.

Problem Xtra #27

```
# The top 10 largest variance index
print(variance_sort[1:10,2])
```

```
##  [1] 353 325 180 187 216 324 403 382 243 208
```

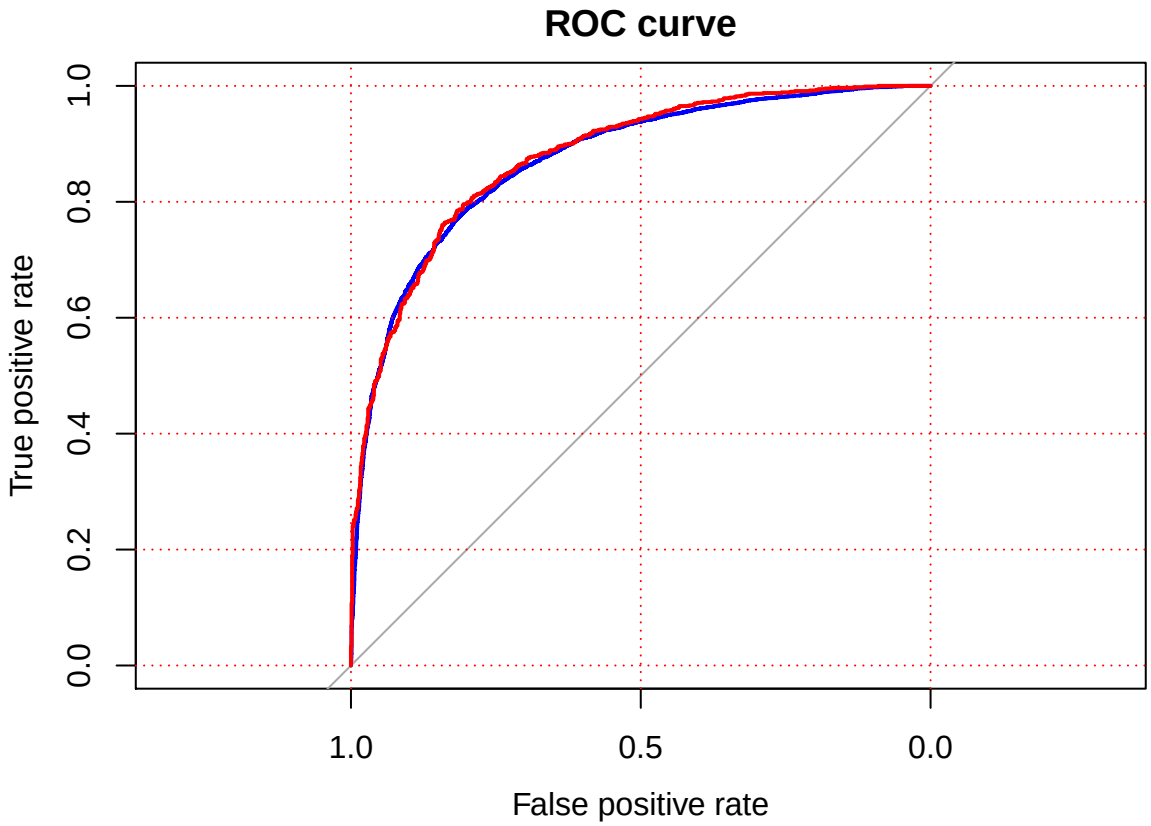
```
# Fit a model
fit.10=glm(y~V353+V325+V180+V187+V216+V324+V403+V382+V243+V208, data=x.train, family='binomial')
summary(fit.10)
```

```
##
## Call:
## glm(formula = y ~ V353 + V325 + V180 + V187 + V216 + V324 + V403 +
##     V382 + V243 + V208, family = "binomial", data = x.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6878  -0.6646  -0.2544   0.6464   2.6928
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.7735621  0.0626622  28.304  < 2e-16 ***
## V353        -0.0020276  0.0003938  -5.149 2.62e-07 ***
## V325        -0.0069939  0.0004434 -15.775  < 2e-16 ***
## V180        -0.0082685  0.0003038 -27.218  < 2e-16 ***
## V187         0.0006050  0.0002995   2.020   0.0434 *
```

```
## V216         0.0023385  0.0003554   6.581 4.69e-11 ***
## V324        -0.0019124  0.0003651  -5.238 1.63e-07 ***
## V403         0.0039901  0.0002276  17.531  < 2e-16 ***
## V382        -0.0029602  0.0003284  -9.013  < 2e-16 ***
## V243        -0.0046932  0.0002918 -16.084  < 2e-16 ***
## V208         0.0034990  0.0002996  11.679  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15971  on 11551  degrees of freedom
## Residual deviance: 10329  on 11541  degrees of freedom
## AIC: 10351
##
## Number of Fisher Scoring iterations: 5
```

```r
# Use the model to predict train and test data
pred.10.train=predict(fit.10, x.train, type='response')
pred.10.test=predict(fit.10, x.test, type='response')

# ROC curve
plot(roc(x.train$y, pred.10.train), col=4, xlab='False positive rate', ylab='True positive rate', main=
lines(roc(x.test$y, pred.10.test), col = 2)
grid(col=2)
```




```r
# auc area
auc.train.10=auc(roc(x.train$y, pred.10.train), col=4)
```

```r
auc.test.10=auc(roc(x.test$y, pred.10.test), col=2)
print(c(auc.train.10, auc.test.10))
```

```
## [1] 0.8725303 0.8781335
```

This classifier performs good since the auc area is bigger than the former two classifier. In addition, the auc on test dataset and train dataset is close, which means the classifier can make a good prediction.

I think this is not a good way to select 10 predictors, since there are 10 different predictors, we need to collect lots of information to build this model and cannot capture non-linear features. Instead, we can make some transformation of some variables like x^2, $logx$ and etc. as predictors to generate 10 predictors.