lab5

YILIN LI

2019/10/14

Lab 5

Question 1

```
war <- read.csv("http://www.stat.cmu.edu/~cshalizi/uADA/15/hw/06/ch.csv", row.names = 1)
war["exports2"] = war$exports^2
war = na.omit(war)

log_model = glm(start~exports2 + exports + schooling + growth + peace + concentration + lnpop + fractio

summary(log_model)
```

```
##
## Call:
## glm(formula = start ~ exports2 + exports + schooling + growth +
##     peace + concentration + lnpop + fractionalization + dominance,
##     family = binomial, data = war)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3655  -0.3627  -0.1893  -0.0932   3.3636
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.307e+01  2.795e+00  -4.677 2.91e-06 ***
## exports2          -2.944e+01  1.178e+01  -2.499 0.012449 *
## exports            1.894e+01  5.865e+00   3.229 0.001243 **
## schooling         -3.156e-02  9.784e-03  -3.225 0.001259 **
## growth            -1.152e-01  4.307e-02  -2.675 0.007466 **
## peace             -3.713e-03  1.093e-03  -3.397 0.000681 ***
## concentration     -2.487e+00  1.005e+00  -2.474 0.013357 *
## lnpop              7.677e-01  1.658e-01   4.632 3.63e-06 ***
## fractionalization -2.135e-04  9.102e-05  -2.345 0.019020 *
## dominance          6.704e-01  3.535e-01   1.896 0.057920 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 337.73  on 687  degrees of freedom
## Residual deviance: 256.42  on 678  degrees of freedom
## AIC: 276.42
##
## Number of Fisher Scoring iterations: 7
```

As reported above, variables that are significant at 5% level are squared exports, exports, schooling, growth, peace, concentration, lnpop, and fractionalization.

Question 2

1

```r
# make the prediction data set
new_data = war[272,]
new_data = do.call('rbind',replicate(3,new_data,simplify = FALSE))
new_data[2,5] = new_data[2,5] + 30
new_data[3,4] = new_data[3,4] + 0.1
new_data["exports2"] = new_data$exports^2

pred = predict(log_model, newdata = new_data, type = "response")
pred
```

```
##       500      5001      5002
## 0.3504199 0.1730900 0.6961378
```

As shown above, probability for a civil war in India in the period beginning 1975 is 0.35. Probability for a country just like India in 1975, except that its male secondary school enrollment rate was 30 points higher, is 0.17. Probability for a country just like India in 1975, except that the ratio of commodity exports to GDP was 0.1 higher, is 0.70.

2

```r
# make the prediction data set
new_data2 = war[464,]
new_data2 = do.call('rbind',replicate(3,new_data2,simplify = FALSE))
new_data2[2,5] = new_data2[2,5] + 30
new_data2[3,4] = new_data2[3,4] + 0.1
new_data2["exports2"] = new_data2$exports^2

pred = predict(log_model, newdata = new_data2, type = "response")
pred
```

```
##        802       8021       8022
## 0.17099172 0.07410315 0.33100440
```

As shown above, probability for a civil war in Nigeria in the period beginning 1965 is 0.17. Probability for a country just like Nigeria in 1965, except that its male secondary school enrollment rate was 30 points higher, is 0.074. Probability for a country just like Nigeria in 1965, except that the ratio of commodity exports to GDP was 0.1 higher, is 0.33.

3

Clearly the changes of the probabilities of two different countries are different by changing the same amount of predictor variables. If we look at these two countries without any change of predictors, the base probabilities are different because of all the predictors of these two countries are totally different. You cannot expect that change same amount of 1 predictor will result in change of same predicted probability.

Question 3

1

```
prob = predict(log_model, type = "response")
p = rep(0, 688)
p[prob>.5] = 1
t = table(p, war$start)
t
```

```
##
## p     0   1
##   0 637  43
##   1   5   3
```

2

```
(sum(t)-sum(diag(t)))/sum(t)
```

```
## [1] 0.06976744
```

Thus the misclassificaiton rate is 0.070.

3

```
Whole_data_set = (1075+14)/1288
On_log_model = (688-sum(p))/688
Whole_data_set
```

```
## [1] 0.8454969
```

```
On_log_model
```

```
## [1] 0.9883721
```

As shown above, the pundit will predict correctly with 0.85 probability on the whole data set and with 0.99 probability on the predicted model.

Question 4

1

```
library(MASS)
lda_model = lda(start~exports2 + exports + schooling + growth + peace + concentration + lnpop + fraction

prob = predict(lda_model)
p = prob$class
t = table(p, na.omit(war)$start)
rate = (sum(t)-sum(diag(t)))/sum(t)
rate
```

```
## [1] 0.06686047
```

As shown above, the training misclassification rate is 0.067.

2

```
qda_model = qda(start~exports2 + exports + schooling + growth + peace + concentration + lnpop + fractio

prob = predict(qda_model)
p = prob$class
t = table(p, na.omit(war)$start)
rate = (sum(t)-sum(diag(t)))/sum(t)
rate
```

```
## [1] 0.07267442
```

As shown above, the training misclassification rate is 0.073.

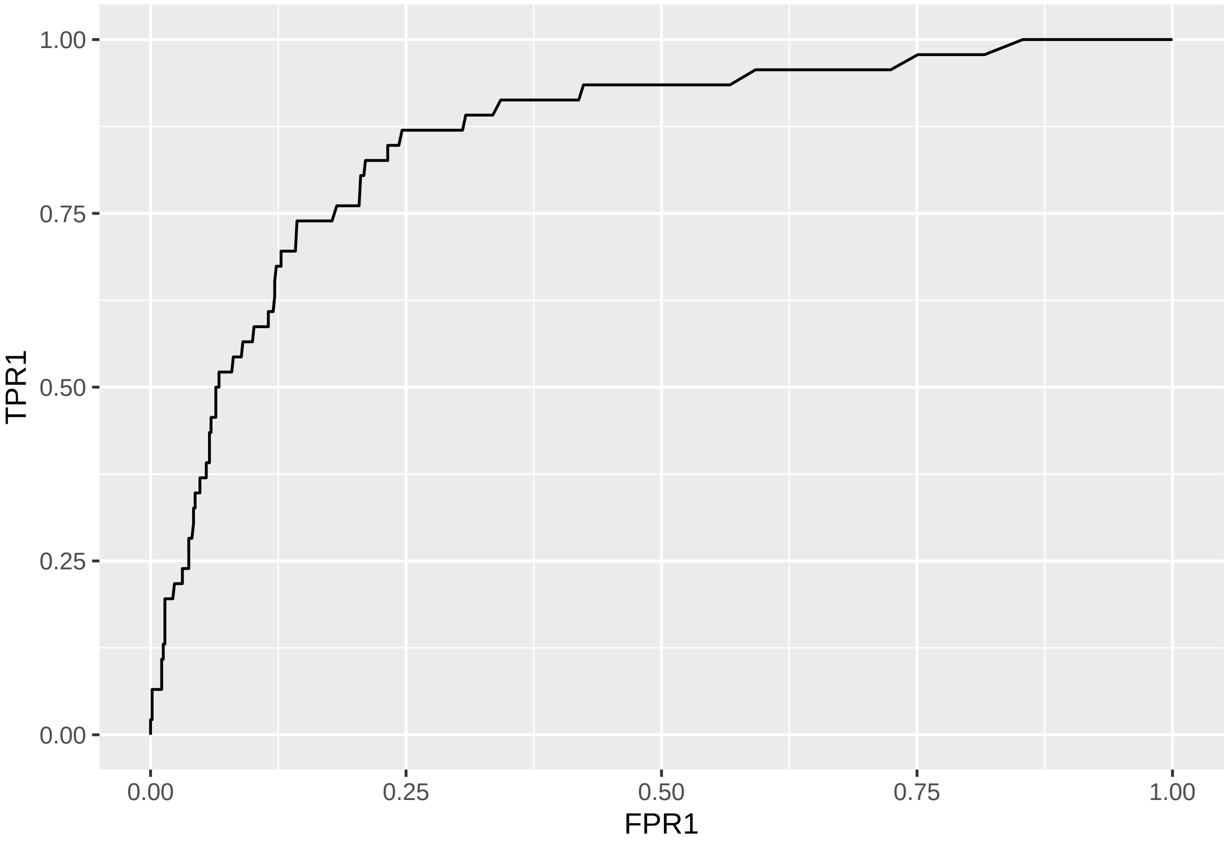
3

Comparing three training models, clearly, the lowest training misclassification rate is the LDA model. The highest training misclassification rate is the QDA model. The logistic model is in bewteen. This makes sense because QDA is the most flexible model in this scenario and most likely to be overfitting. The result that LDA is better than Logistic shows that the predictors tend to be normally distributed.

Challenge Problem

```
library(ggplot2)
thres = seq(1,0,length=1000)
TPR1 = rep(0,1000)
FPR1 = rep(0,1000)
prob = predict(log_model, type = "response")
for(i in seq(1,1000,length = 1000)){
  p = rep(0,688)
  p[prob>thres[i]] = 1
  t = table(p, war$start)
  if(length(rownames(t))>1){
  TPR1[i] = t[2,2]/(t[1,2]+t[2,2])
  FPR1[i] = t[2,1]/(t[2,1]+t[1,1])
  } else if(rownames(t)==1){
    TPR1[i] = 1
    FPR1[i] = 1
  } else{
    TPR1[i] = 0
    FPR1[i] = 0
  }
}

ggplot() + geom_line(aes(x = FPR1, y = TPR1))
```



Problem Set 5

Exercise 4

a. On average 10%, because for each X, the fraction is 10% and X is uniformly distributed. So the overall average is also 10%.

b. On average 1%, because for each (X_1, X_2) pair, there're $0.1 \times 0.1 = 0.01$ observations that predict the response. Also, the pairs are unifomrly distributed, so overall average is also 1%.

c. Clearly, following the rules of part a and b, we have 0.1^{100} observations.

d. As p increases linearly, the training observations that near the test observation decreases exponentially.

e.

$$p = 1 \rightarrow l = 0.1$$
$$p = 2 \rightarrow l^2 = 0.1 \rightarrow l = \sqrt{0.1} = 0.32$$
$$p = 3 \rightarrow l^3 = 0.1 \rightarrow l = \sqrt[3]{0.1} = 0.46$$
$$p = n \rightarrow l^n = 0.1 \rightarrow l = \sqrt[n]{0.1}$$

Exercise 6

a.

```
prob = 1/(1+exp(-(-6+0.05*40+1*3.5)))
prob
```

```
## [1] 0.3775407
```

So the probability of getting an “A” is 37.75%.

b. We solved equation:

$$0.5 = \frac{1}{1 + e^{-(-6+0.05\times X_1+1\times 3.5)}}$$

and found that $X_1 = 50$ hours

Exercise 7

$$\begin{aligned}
P(Y=1|X=x) &= \frac{P(X=x|Y=1)\times P(Y=1)}{P(X=x|Y=1)\times P(Y=1)+P(X=x|Y=0)\times P(Y=0)}\\
&= \frac{f_1\times\pi_1}{f_1\times\pi_1+f_0\times\pi_0}\\
&= \frac{0.8\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}}{0.8\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}+0.2\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu_0)^2}{2\sigma^2}}}\\
&= \frac{0.8\times e^{-\frac{(x-10)^2}{2\times 36}}}{0.8\times e^{-\frac{(x-10)^2}{2\times 36}}+0.2\times e^{-\frac{x^2}{2\times 36}}}
\end{aligned}$$

$$P(Y=1|X=4) = \frac{0.8\times e^{-\frac{(4-10)^2}{2\times 36}}}{0.8\times e^{-\frac{(4-10)^2}{2\times 36}}+0.2\times e^{-\frac{4^2}{2\times 36}}} = 0.752$$