# Homework 2 - Data Mining

Max Machalek

March 6, 2019

Revisited Jan 04, 2021 (updated hypotheses, phrasing)

## H.2.1: Fit the first-order Logistic Regression model.

**(a) Perform the hypothesis test associated with Pclass at alpha = 0.05. Give the hypotheses, test statistic, statistical decision, and practical conclusion of the test.**

Ho: $\beta_1 = 0$

*Passenger Class has no effect on the survival rate of passengers in the population.*

Ha: $\beta_1 \neq 0$

*Passenger Class has a linear effect on the survival rate of passengers in the population.*

z: -5.833
p: 5.44e-09 (=0.00000000544), (< 0.05)

Because $p < 0.05$, we reject the null hypothesis and conclude that $\beta_1$ is not 0. There is evidence at the 0.05 level of linear association in the population between the passenger class indicator and the log-odds of survival.

**(b) Interpret the odds estimate and the associated 95% confidence interval for Pclass.**

95% Interval for Odds Estimate: (0.24926793, 0.5013479)

Odds Estimate: exp(-1.039840939) = 0.35351091

We are 95% confident that the true value of the Pclass odds is between 0.24926793 and 0.5013479.

95% Interval for Log Odds Estimate: (-1.389, -.690)

Log Odds Estimate: -1.039840939

We are 95% confident that the true value of the Pclass log odds estimate is between -1.389 and -0.690.

The *estimated log odds* of survival is 1.039x lower for a passenger in a lower class compared to a passenger in the class above them (3rd class being the 'lowest', 1st the 'highest'). So, the *estimated odds* of survival are 0.3535 times as high for a passenger in one class than another passenger in the class above (1st class being above 2nd, 2nd above 3rd).

**(c) Give the confusion matrix and error rate using resubstitution. Explain what the confusion matrix indicates about the types of prediction errors made by the logistic regression model.**

Sensitivity: 139/200 = 0.695 = 69.5%
Specificity: 290/334 = 0.868 = 86.8%

Our confusion matrix shows a low sensitivity (0.695). Specificity is 0.868, which is quite a bit higher. This suggests that the model is more likely to have false negative errors (type 2), which would be predicting that a passenger does not survive (a 0 value), when in actuality they do (a 1 value).

```
      y
yh   0   1
0   290 61
1   44  139
```

*err: 0.1966292*

```
#err.f function
err.f = function(y,yh) {ct = table(yh,y)
  err = 1-sum(diag(ct))/length(y); list(table = ct, err = err)}

y = titanic$Survived
x1 <- titanic$Pclass
x2 <- titanic$Sex
x3 <- titanic$Age
x4 <- titanic$SibSp
x5 <- titanic$Parch
x6 <- titanic$Fare
x <- cbind(x1,x2,x3,x4,x5,x6)
standard.x <- scale(x)

n <- nrow(titanic)

lrFit <- glm(Survived~.,family=binomial(link="logit"), data = titanic)
```

```r
# logit link default
sfit <- summary(lrFit)

bh.o <- cbind(sfit$coefficients, confint.default(lrFit))
bh.o
```

```
##                Estimate  Std. Error     z value     Pr(>|z|)
2.5 %
## (Intercept)  4.816979278 0.676708440   7.1182491 1.093067e-12
3.490655106
## Pclass      -1.039840939 0.178261439  -5.8332354 5.436278e-09 -
1.389226940
## Sex         -2.734136675 0.256960758 -10.6402888 1.935286e-26 -
3.237770507
## Age         -0.041517035 0.009872805  -4.2051915 2.608612e-05 -
0.060867376
## SibSp       -0.328510543 0.138170667  -2.3775708 1.742710e-02 -
0.599320075
## Parch       -0.120901657 0.154706824  -0.7814888 4.345150e-01 -
0.424121460
## Fare         0.002679197 0.003023651   0.8860800 3.755744e-01 -
0.003247051
##                   97.5 %
## (Intercept)  6.143303449
## Pclass      -0.690454938
## Sex         -2.230502843
## Age         -0.022166693
## SibSp       -0.057701012
## Parch        0.182318147
## Fare         0.008605444
```

```r
exp(cbind(ORh=bh.o[,1],bh.o[,5:6]))[-1,]
```

```
##                ORh      2.5 %    97.5 %
## Pclass 0.35351091 0.24926793 0.5013479
## Sex    0.06495006 0.03925131 0.1074744
## Age    0.95933299 0.94094802 0.9780772
## SibSp  0.71999534 0.54918491 0.9439321
## Parch  0.88612110 0.65434440 1.1999959
## Fare   1.00268279 0.99675822 1.0086426
```

```r
#confusion matrix ( not using resubstitution )
p.lr = predict(lrFit, type="response")
yh.lr = as.numeric(p.lr>=0.5)
err.f(y,yh.lr)
```

```
## $table
##    y
## yh    0   1
##   0 290  61
##   1  44 139
```

```
##
## $err
## [1] 0.1966292

yh.knn = knn(standard.x,standard.x,y,k=5) ## resubstitution because
standard.x == standard.x (same data)
```

## H.2.2 Fit all input variables for LDA, QDA, and KNN

**(a) Which variables are most important for discrimination with LDA? Explain**

Sex has the largest impact on ability to discriminate, because magnitude of the scaled LD1 coefficient is the largest for this predictor (|-0.875| = 0.875).

**(b) Assess the assumption of equal population covariance matrices at alpha = 0.05. Give the statistical hypotheses, test statistic, statistical decision, and practical conclusion of the test. Should QDA predict better than LDA given this test result and the discussion in Section 4.5?**

Ho: $\Sigma_1 = \Sigma_2$

*The two population covariance matrices do not differ.*

Ha: $\Sigma_1 \neq \Sigma_2$

*The two population covariance matrices are different.*

Test statistic c(21) = 291.65
p < 2e-16

Because c(21) = 291.65 and p < 2e-16, we reject the null hypothesis at the 0.05 level and conclude that there is evidence to question the assumption of equal population covariance matrices across default status. Additionally, the Cholesky residuals deviate quite a bit from the theoretical normal line, especially toward the right tail.

Given this test result, we would expect QDA to predict better than LDA because its quadratic boundaries allow for a little more flexibility. Additionally we found evidence that the two covariance matrices did not match, a requisite assumption to *appropriately* use LDA.

**(c) Compare the LOOCV error rates for LDA, QDA, and KNN (K = 5). Which method appears to be better in predicting Survival?**

LDA Error: 0.2097378

QDA Error: 0.2059925

KNN (k=5) Error: 0.2022472

KNN (k=5) has the lowest LOOCV error rate. QDA has the second lowest, followed by LDA with the worst error rate. Given these error rates, KNN (k=5) appears to be the best method to predict survival.

LDA confusion matrix

```
    y
yh   0   1
0  287 65
1   47 135
```

*err: 0.2097378*

QDA confusion matrix

```
    y
yh   0   1
0  288 64
1   46 136
```

*err: 0.2059925*

KNN confusion matrix

```
    y
yh   0   1
0  286  60
1   48 140
```

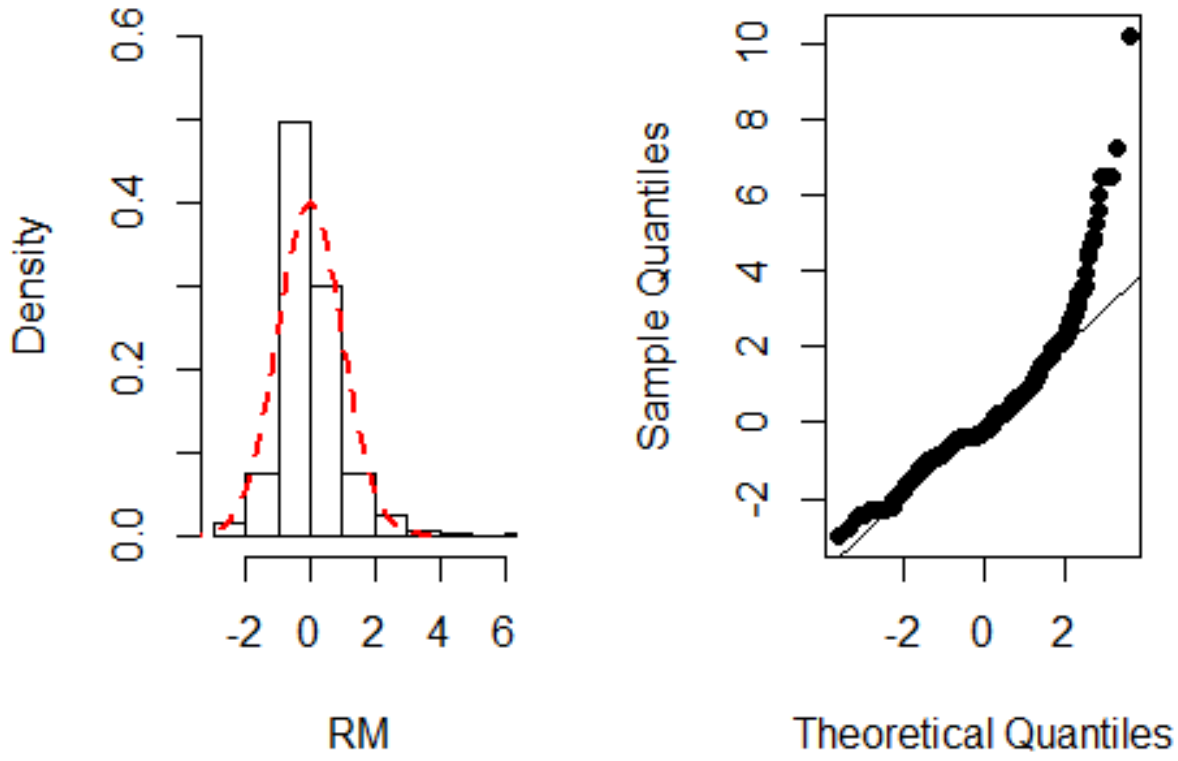
*err: 0.2022472*

```r
boxM(x,y) # box M test #

##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  x
## Chi-Sq (approx.) = 291.65, df = 21, p-value < 2.2e-16

mlm <- lm(x~y)
sp <- Manova(mlm)$SSPE/mlm$df.residual # pooled covariance matrix #
RM <- mlm$residuals%*%t(chol(solve(sp))) # Cholesky Residuals #
outlier.p <- c(exp=2*(1-pnorm(3)),obs=sum(abs(RM)>=3)/n) #
0.002699796, 0.069288390 #
```

```r
#y for first plot is density instead of frequency like in notes, ?
par(mfrow=c(1,2))
hist(RM, main=NULL, freq=F, ylim=c(0,0.6), xlim=c(-3,6))
curve(dnorm,-4, 4, add=T, lty=2, col='red',lwd=2)
qqnorm(RM, pch=16, main=NULL); abline(0,1)
```




```r
# model fit - LDA and QDA #
prior0 <- c(1,1)/2
da1 <-  lda(y~x, prior=prior0, CV=FALSE)
ld.raw <- da1$scaling
ld.std <- diag(sqrt(diag(sp)))%*%ld.raw
rbind(t(ld.raw),t(ld.std))
```

```
##            xx1       xx2         xx3        xx4         xx5
xx6
## LD1 -0.6826738 -2.158794 -0.02681108 -0.1865927 -0.09883254
0.00203539
```

```
## LD1 -0.5466948 -0.874855 -0.35497800 -0.2050589 -0.07804787
0.09265525

prior00 <- as.vector(table(y)/n)
CVS <- TRUE

da1 <- lda(y~x,prior=prior00,CV=TRUE)
da2 <- qda(y~x,prior=prior00,CV=TRUE)

da <- da1
if (CVS==F) yh.da = predict(da, as.data.frame(y))$class
if (CVS==T) yh.da = da$class
err.f(y,yh.da)

## $table
##    y
## yh    0   1
##   0 287  65
##   1  47 135
##
## $err
## [1] 0.2097378

da <- da2
if (CVS==F) yh.da = predict(da, as.data.frame(y))$class
if (CVS==T) yh.da = da$class
err.f(y,yh.da)

## $table
##    y
## yh    0   1
##   0 288  64
##   1  46 136
##
## $err
## [1] 0.2059925

## knn LOOCV error rate##
yh.knn = knn.cv(standard.x,y,k=5) ## loocv


err.f(y,yh.knn)

## $table
##    y
## yh    0   1
##   0 286  60
##   1  48 140
##
## $err
## [1] 0.2022472
```