PA: HW6

Jieda Li, Kristian Nikolov, Kristiyan Dimitrov, Parth Patel

November 13, 2019

Problem 7.6

a)

```
alone = 14/21
withvcr = 38/42

SLO = log((withvcr/(1-withvcr))/(alone/(1-alone)))
varSLO = (1/(21*(alone)*(1-alone)))+(1/(42*(withvcr)*(1-withvcr)))
z_score = (SLO - 0)/sqrt(varSLO)

print(paste("The sample log-odds ratio is:", round(SLO, digits = 3)))
```

```
## [1] "The sample log-odds ratio is: 1.558"
```

```
print(paste0("The z-test value is: ", round(z_score, digits =3), ", which is statistically different fro
```

```
## [1] "The z-test value is: 2.225, which is statistically different from 0"
```

b)

```
p_hat = (21*alone + 42*withvcr)/(21+42)
z_score2 = (withvcr - alone)/sqrt((p_hat*(1-p_hat))*((1/21)+(1/41)))

print(paste0("The computed large sample z-test is: ", z_score2, ", which is fairly close to the result f
```

```
## [1] "The computed large sample z-test is: 2.33721667225477, which is fairly close to the result from
```

Problem 7.7

a) Women in this subset apply to departments which have overall lower admission rates, which is why they tend to not get accepted, while men apply to departments with higher admissions rates, resulting in the overall trend for men to be accepted more.

b)

```
UCB <- read.csv('UCBAdmissions.csv')
p0 = 0.304
p1 = 0.445
slo = log((p1/(1-p1))/(p0/(1-p0)))

fit1 = glm(Admit ~ Gender, family = binomial, data = UCB)
summary(fit1)
```

```
##
## Call:
## glm(formula = Admit ~ Gender, family = binomial, data = UCB)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0855  -1.0855  -0.8506   1.2722   1.5442
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.83049    0.05077 -16.357   <2e-16 ***
## Gender       0.61035    0.06389   9.553   <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: 6044.3  on 4525  degrees of freedom
## Residual deviance: 5950.9  on 4524  degrees of freedom
## AIC: 5954.9
##
## Number of Fisher Scoring iterations: 4
```

Both numbers are ~0.61

c)

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.5.3
```

```
UCB = UCB[1:8]
UCB_melt = melt(UCB, id = c("Admit", "Gender"))
colnames(UCB_melt) = c("Admit", "Gender", "Department", "value")
UCB_melt = UCB_melt[which(UCB_melt$value==1), names(UCB_melt) %in% c("Admit", "Gender", "Department")]
UCB_melt$Department = as.numeric(UCB_melt$Department)
fit2 = glm(Admit ~ Gender + Department, family = binomial, data = UCB_melt)
summary(fit2)
```

```
##
## Call:
## glm(formula = Admit ~ Gender + Department, family = binomial,
##     data = UCB_melt)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.504  -1.029  -0.507   1.099   2.060
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.28607    0.10141  12.682   <2e-16 ***
## Gender      -0.00839    0.07350  -0.114    0.909
## Department  -0.54545    0.02307 -23.638   <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: 6044.3  on 4525  degrees of freedom
## Residual deviance: 5272.6  on 4523  degrees of freedom
## AIC: 5278.6
##
## Number of Fisher Scoring iterations: 3
```

As anticipated, the department plays a much stronger role in determining whether a person is accepted or not, and the gender sign has been reversed, since, as seen, for more than half of the departments it was overall more common for women to be accepted more.

Problem 7.8

a)

```r
rad = read.csv('Radiation.csv')
rad = rad[1:24,]
fit1 = glm(Success ~ Days, family = binomial, data = rad)
rad_sum = summary(fit1)
```

b)

```r
CIlower = exp(5*(rad_sum$coefficients[2]-1.96*(rad_sum$coefficients[4])))
CIupper = exp(5*(rad_sum$coefficients[2]+1.96*(rad_sum$coefficients[4])))
print(paste0("The 95% confidence interval for the odds of absence of tumor vs. presence of tumor if the
```

```
## [1] "The 95% confidence interval for the odds of absence of tumor vs. presence of tumor if the numbe
```

0 is not in the interval, so we can assume that 5 days of therapy makes a difference on the outcome.

c)

```r
new = data.frame(rad$Days)
testpredict = predict(fit1, newdata = rad$days, type = 'response')
p_star = 0
max_ccr = 0
best_pstar = 0
for(i in seq(from=0.01, to=1.0, by=0.01)){
  curr_tab= table(rad$Success, testpredict>i)
  curr_ccr = sum(diag(curr_tab))/sum(curr_tab)
  if(curr_ccr > max_ccr){
    max_ccr = curr_ccr
    best_pstar = i
    }
}
print(max_ccr)
```

```
## [1] 0.75
```

```r
print(best_pstar)
```

```
## [1] 0.51
```

```r
tab = table(rad$Success, testpredict>best_pstar)
measures <- function(m){
  CCR = sum(diag(m))/sum(m)
  spe = m[1]/sum(m[1,])
  sen = m[2,2]/sum(m[2,])
  pre = m[2,2]/sum(m[,2])
  re = m[2,2]/sum(m[2,])
  f1 = (2*pre*re)/(pre+re)
  all = paste("CCR is:", CCR, "\nSensitivtiy is", sen, "\nSpecificity is", spe, "\nPrecision is", pre,
  cat(all, sep='\n')
}
measures(tab)
```

```
## CCR is: 0.75
## Sensitivtiy is 0.785714285714286
## Specificity is 0.7
## Precision is 0.785714285714286
## Recall is 0.785714285714286
## F1-score is 0.785714285714286
```

Problem 7.10

```
m1 <- c(8100, 100, 900, 900)
dim(m1) <- c(2,2)
m2 <- c(8910, 10, 990, 90)
dim(m2) <- c(2,2)

measures(m1)
```

```
## CCR is: 0.9
## Sensitivtiy is 0.9
## Specificity is 0.9
## Precision is 0.5
## Recall is 0.9
## F1-score is 0.642857142857143
```

```
measures(m2)
```

```
## CCR is: 0.9
## Sensitivtiy is 0.9
## Specificity is 0.9
## Precision is 0.0833333333333333
## Recall is 0.9
## F1-score is 0.152542372881356
```

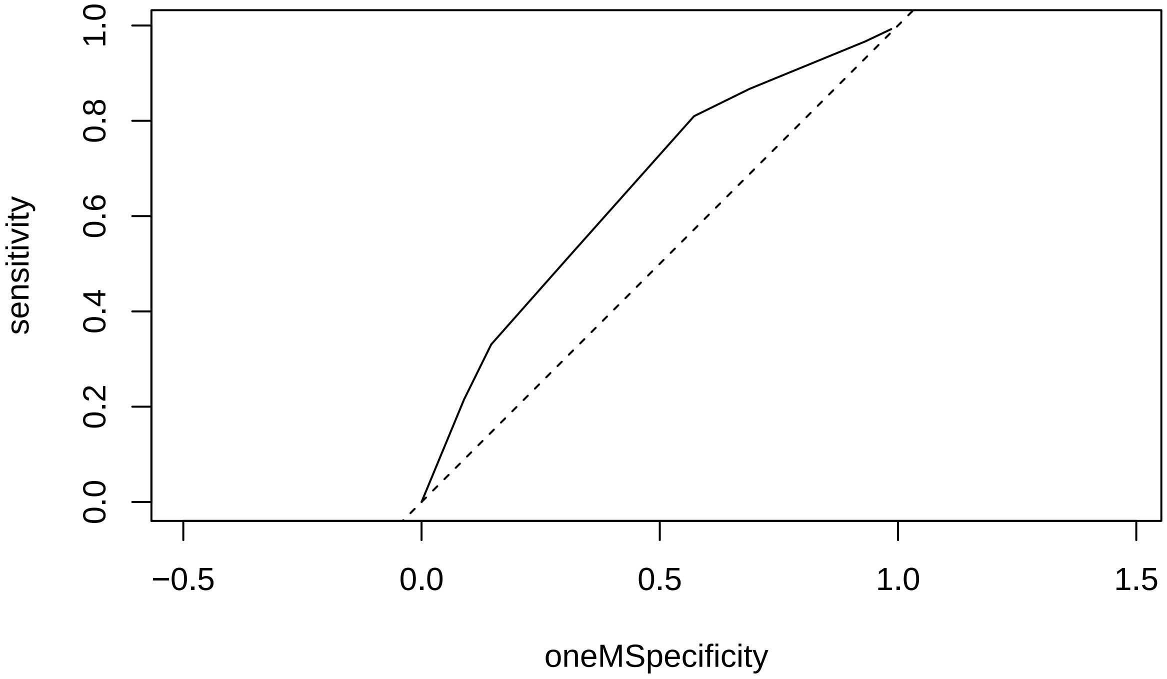
When the samples are imbalanced, such as in the second classifier, precision and with it the F1-score are much more easily affected when the model misclassifies (in this case negatives as positives). Both models perform well, but the first classifier performs a bit better.

Problem 7.11 a)

```
art = read.csv('Art-education group data.csv')
sumY = 0
sumN = 0
i = 1
sensitivity = c()
oneMSpecificity = c()
while (i < (nrow(art)+1)){
  Y = art[i,2]
  N = art[i,3]
  sumY = sumY + Y
  sens = (sum(art["Yes"])-sumY)/sum(art["Yes"])
  sensitivity = append(sensitivity, sens)
  sumN = sumN + N
  spec = 1 - sumN/sum(art["No"])
  oneMSpecificity = append(oneMSpecificity, spec)
  i = i+1
}
ROC <- data.frame("oneMSpecificity" = oneMSpecificity, "sensitivity" = sensitivity)
#plot sensitivity against specificity
```

b)

```
plot(ROC, type = 'l', asp = 1)
abline(0, 1, lty = 2)
```



It is the same graph as the one in Fig 7.6.