SS3860_assignment_1

Hussien Hussien

Question #1

a) Given the binary logistic link function, log[p/(1 − p)], derive the inverse link function , 1/[1 + e^−]. Show each step in the derivation.

equation block:

$$\eta = log\frac{p}{1-p}e^{\eta} = \frac{p}{1-p}1+e^{\eta} = \frac{p}{1-p}+\frac{1-p}{1-p} = \frac{1}{1-p}\frac{1}{1+e^{\eta}} = 1-pp = 1-\frac{1}{1+e^{\eta}} = \frac{1+e^{\eta}}{1+e^{\eta}}-\frac{1}{1+e^{\eta}}p = \frac{e^{\eta}}{1+e^{\eta}}\blacksquare$$

b) In what way does the relationship of the linear predictor and fitted value differ between normal (Gaussian) models and logistic models?

1. Logistic Functions output a discrete value when given a predictor. Guassian Models will generally give a continuous variable.
2. The error terms ( ) are not normally (Gaussian) distributed

Question #2

a) Please interpret the estimated coefficients for hmo, los and the interaction term in terms of odds and odds ratio.

los: -0.0276960

`Odds:` When a patient *is NOT* part of an HMO, the odds of a patient dying decreases by $1-e(-0.0276960) = 2.73\%$ with each additional day they stay in the hospital

`Odds Ratio:` 0.972684 is odds ratio corresponding to an increase of los by 1 day amongst non-hmo patients

hmoyes:los: -0.0277788

`Odds:` When a patient *is* part of an HMO, the odds of a patient dying decreases by $1 - (e(-0.0276960) * e(-0.0277788)) = 5.4\%$ with each additional day they stay in the hospital

`Odds Ratio:` 0.9726035 is the odds ratio corresponding to an increase of los by 1 day amongst hmo patients

$1-e^(-0.0277788) = 2.8\% $ is the increase in odds ratio when comparing the odds ratio corresponding to a change in los by 1 day for HMO-patients versus the odds ratio corresponding to a change in los by 1 day for non-hmo-patients

hmoyes 0.1925012

`Odds:` When a patient has spent 0 days in the hospital, the odds of a patient dying increases by $1 - e(0.1925012) = 21\%$ if they are part of an HMO

`Odds Ratio:` 1.212278 is the odds ratio for hmo vs. non-hmo patients when they have spent 0-days in the hospital

b) Why using the p-values (based on z-values) provided in the summary table above may not be a good way of assessing the siginificance of individual predictors? What are two other better strategies for doing such assessement?

Better ways of assessing significance of individual predictors: * Deviance based method * Better confidence interval

Question #3

a) Fit a binary (logistic) regression with Class as the response variable and the other nine variables as predictors. Report the residual deviance and associated degrees of freedom. Can this information be used to assess if this model fits the data? Explain.

Yes this information can be used to test the goodness of fit of this model using deviance based methods.

```
library(faraway)
suppressMessages(library(dplyr))
wbca <- mutate(wbca,Class_factor = factor(wbca$Class,levels=c("1","0"),labels=c("benign","malignant")))
fit_glm <- glm(Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick + UShap + USize, family=bi
summary(fit_glm)
```

```
##
## Call:
## glm(formula = Class ~ Adhes + BNucl + Chrom + Epith + Mitos +
##     NNucl + Thick + UShap + USize, family = binomial, data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48282  -0.01179   0.04739   0.09678   3.06425
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.16678    1.41491   7.892 2.97e-15 ***
## Adhes       -0.39681    0.13384  -2.965  0.00303 **
## BNucl       -0.41478    0.10230  -4.055 5.02e-05 ***
## Chrom       -0.56456    0.18728  -3.014  0.00257 **
## Epith       -0.06440    0.16595  -0.388  0.69795
## Mitos       -0.65713    0.36764  -1.787  0.07387 .
## NNucl       -0.28659    0.12620  -2.271  0.02315 *
## Thick       -0.62675    0.15890  -3.944 8.01e-05 ***
## UShap       -0.28011    0.25235  -1.110  0.26699
## USize        0.05718    0.23271   0.246  0.80589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.464  on 671  degrees of freedom
## AIC: 109.46
##
## Number of Fisher Scoring iterations: 8
```

As you can see above, `Residual deviance:  89.464  on 671  degrees of freedom`

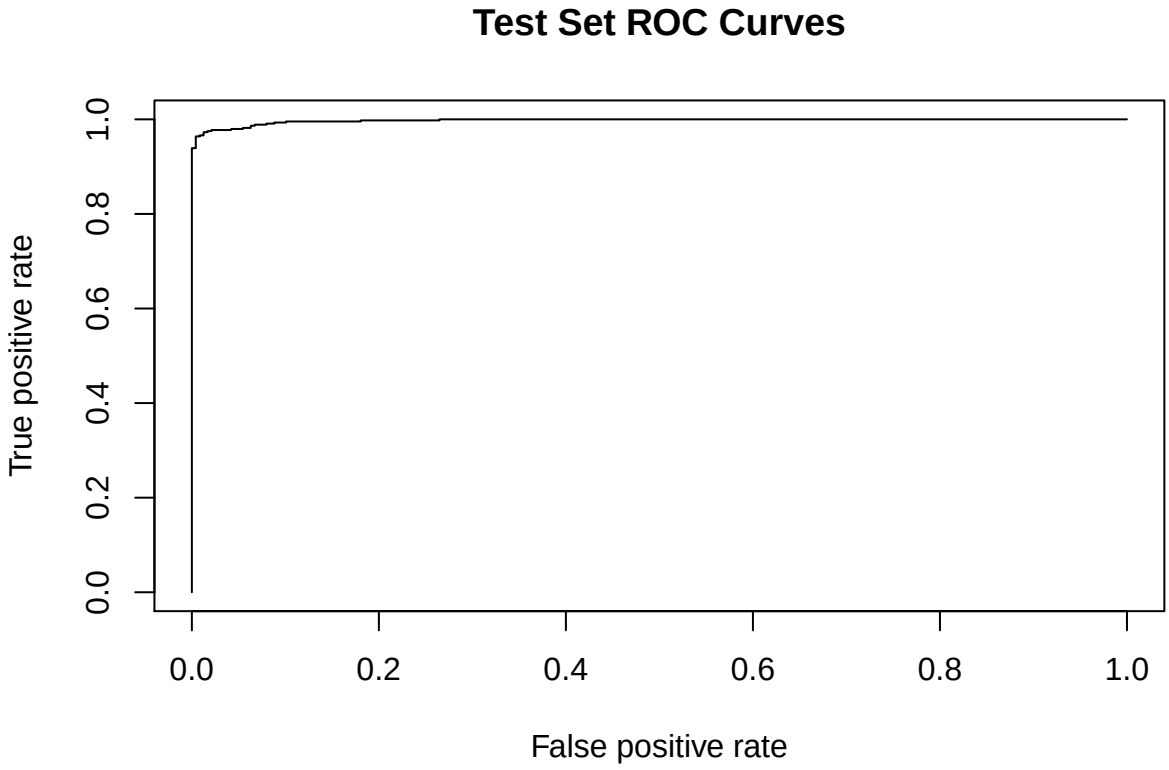
b) Use the AIC criterion to determine the best subset of variables

```
##
## Call:
## glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +
##     Thick + UShap, family = binomial, data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

```
## -2.44161  -0.01119   0.04962   0.09741   3.08205
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.0333     1.3632   8.094 5.79e-16 ***
## Adhes        -0.3984     0.1294  -3.080  0.00207 **
## BNucl        -0.4192     0.1020  -4.111 3.93e-05 ***
## Chrom        -0.5679     0.1840  -3.085  0.00203 **
## Mitos        -0.6456     0.3634  -1.777  0.07561 .
## NNucl        -0.2915     0.1236  -2.358  0.01837 *
## Thick        -0.6216     0.1579  -3.937 8.27e-05 ***
## UShap        -0.2541     0.1785  -1.423  0.15461
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.662  on 673  degrees of freedom
## AIC: 105.66
##
## Number of Fisher Scoring iterations: 8
```

c) Produce an ROC curve based on the selected model in b) and comment on the effectiveness of the new diagnostic test.

```
ROCpred <- predict(small_model, wbca)
ayo <- prediction(ROCpred,wbca$Class)
rocs <- performance(ayo,'tpr','fpr')
m <- length(ROCpred)
plot(rocs, col = as.list(1:m), main = "Test Set ROC Curves")
```



When analyzing the AUC (below) this new model looks very effective at 0.997

```
auc_ROCR <- performance(ayo, measure = "auc")
auc_ROCR@y.values[[1]]
```

```
## [1] 0.9974392
```

d) It is usually misleading to use the same data to fit a model and test its predictive ability. What would be a better approach for this? Explain and write a pseudo-code for your proposed approach (you do not need to implement it).

I would train the model only using 80% of the data. Then when validating the model and testing it's accuracy I would use the remaining 20% since the model hasn't seen it. Alternatively, I could use cross-validation. Here is pseodo-code for it.

```
randomize the dataset
partition the data into n silos
for each silo do the following:
- remove that silo of the data and us it as a test set
- fit the model using the rest of the data
- test the model using the silo you reserved
- record the performance using whatever score your choose
summarize results
```