Assignment 10.2

2022-02-18

##Libraries

```
library(foreign)
library(ggplot2)
library(dplyr)
```

```
##
## 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(mlogit)
```

```
## Loading required package: dfidx
```

```
##
## Attaching package: 'dfidx'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

Binary Classifer Data

```
setwd("/Users/logan/Documents/GitHub/dsc520clone")

#Set WD
setwd("/Users/logan/Documents/GitHub/dsc520clone")

# Binary Classifer data
##read data into df
binary_class_df <- read.csv('data/binary-classifier-data.csv')
head(binary_class_df)
```

```
##   label        x        y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

```r
##create binary model using glm
binary_glm <- glm(label ~ x + y, data=binary_class_df, family=binomial(link='logit'))
summary(binary_glm)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = binomial(link = "logit"),
##     data = binary_class_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3728  -1.1697  -0.9575   1.1646   1.3989
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.424809   0.117224   3.624  0.00029 ***
## x           -0.002571   0.001823  -1.411  0.15836
## y           -0.007956   0.001869  -4.257 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2075.8  on 1497  degrees of freedom
## Residual deviance: 2052.1  on 1495  degrees of freedom
## AIC: 2058.1
##
## Number of Fisher Scoring iterations: 4
```

```r
##create new columns for probability based on model
binary_class_df$probability <- fitted(binary_glm)
head(binary_class_df)
```

```
##   label        x        y probability
## 1     0 70.88469 83.17702   0.3967211
## 2     0 74.97176 87.92922   0.3852176
## 3     0 73.78333 92.20325   0.3779152
## 4     0 66.40747 81.10617   0.4034378
## 5     0 69.07399 84.53739   0.3952460
## 6     0 72.23616 86.38403   0.3898045
```

```r
binary_class_df$probability_TrueOrFalse <- if_else(binary_class_df$probability > 0.4, 1, 0)
head(binary_class_df)
```

```
##   label        x        y probability probability_TrueOrFalse
## 1     0 70.88469 83.17702   0.3967211                       0
## 2     0 74.97176 87.92922   0.3852176                       0
## 3     0 73.78333 92.20325   0.3779152                       0
## 4     0 66.40747 81.10617   0.4034378                       1
## 5     0 69.07399 84.53739   0.3952460                       0
## 6     0 72.23616 86.38403   0.3898045                       0
```

```r
##compute model accuracy
binary_compare_table <- table(actual = binary_class_df$label, predicted = binary_class_df$probability_T
binary_compare_table
```

```
##       predicted
## actual   0   1
##      0 148 619
##      1  36 695
```

```
binary_accuracy <- ((binary_compare_table[[1,1]] + binary_compare_table[[2,2]]) / sum(binary_compare_tab
binary_accuracy_percent <- binary_accuracy*100
binary_accuracy_percent
```

```
## [1] 56.27503
```

```
###At a threshold of .4 the model is roughly 56.27% accurate.
```

Thoraric Data

```
##load thoraric DF
thoraric_df <- read.arff('data/ThoracicSurgery.Arff')
head(thoraric_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30
## 1 DGN2 2.88 2.16 PRZ1    F    F    F     T     T  OC14     F     F     F     T
## 2 DGN3 3.40 1.88 PRZ0    F    F    F     F     F  OC12     F     F     F     T
## 3 DGN3 2.76 2.08 PRZ1    F    F    F     T     F  OC11     F     F     F     T
## 4 DGN3 3.68 3.04 PRZ0    F    F    F     F     F  OC11     F     F     F     F
## 5 DGN3 2.44 0.96 PRZ2    F    T    F     T     T  OC11     F     F     F     T
## 6 DGN3 2.48 1.88 PRZ1    F    F    F     T     F  OC11     F     F     F     F
##   PRE32 AGE Risk1Yr
## 1     F  60       F
## 2     F  51       F
## 3     F  59       F
## 4     F  54       F
## 5     F  73       T
## 6     F  51       F
```

```
##use GLM() to create the model, included summary in results
log_risk <- glm(Risk1Yr ~ PRE7 + PRE8 + PRE9 + PRE11 + PRE17 + PRE30 + AGE, data=thoraric_df, family= bi
summary(log_risk)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ PRE7 + PRE8 + PRE9 + PRE11 + PRE17 +
##     PRE30 + AGE, family = binomial(link = "logit"), data = thoraric_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1309  -0.5738  -0.5030  -0.3347   2.4224
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.952191   1.052958  -2.804  0.00505 **
## PRE7T        0.503617   0.489483   1.029  0.30354
## PRE8T        0.288178   0.358392   0.804  0.42135
## PRE9T        1.083701   0.440825   2.458  0.01396 *
## PRE11T       0.528516   0.332206   1.591  0.11163
## PRE17T       0.980237   0.412828   2.374  0.01758 *
## PRE30T       0.872242   0.433429   2.012  0.04418 *
```

```
## AGE          0.001254   0.015915   0.079  0.93719
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.61  on 469  degrees of freedom
## Residual deviance: 376.05  on 462  degrees of freedom
## AIC: 392.05
##
## Number of Fisher Scoring iterations: 5
```

```
### According to the summary PRE9 and P17 are having the greatest effect on the outcome.
### The P values of .013 and .017 also indicate they are statistically significant.

##create columns for probability based on model
thoraric_df$probability <- fitted(log_risk)
head(thoraric_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30
## 1 DGN2 2.88 2.16 PRZ1    F    F    F     T     T  OC14     F     F     F     T
## 2 DGN3 3.40 1.88 PRZ0    F    F    F     F     F  OC12     F     F     F     T
## 3 DGN3 2.76 2.08 PRZ1    F    F    F     T     F  OC11     F     F     F     T
## 4 DGN3 3.68 3.04 PRZ0    F    F    F     F     F  OC11     F     F     F     F
## 5 DGN3 2.44 0.96 PRZ2    F    T    F     T     T  OC11     F     F     F     T
## 6 DGN3 2.48 1.88 PRZ1    F    F    F     T     F  OC11     F     F     F     F
##   PRE32 AGE Risk1Yr probability
## 1     F  60       F  0.18600471
## 2     F  51       F  0.11753467
## 3     F  59       F  0.11857933
## 4     F  54       F  0.05292682
## 5     F  73       T  0.23654785
## 6     F  51       F  0.05273854
```

```
thoraric_df$probability_TrueOrFalse <- if_else(thoraric_df$probability > .25, T, F)
head(thoraric_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30
## 1 DGN2 2.88 2.16 PRZ1    F    F    F     T     T  OC14     F     F     F     T
## 2 DGN3 3.40 1.88 PRZ0    F    F    F     F     F  OC12     F     F     F     T
## 3 DGN3 2.76 2.08 PRZ1    F    F    F     T     F  OC11     F     F     F     T
## 4 DGN3 3.68 3.04 PRZ0    F    F    F     F     F  OC11     F     F     F     F
## 5 DGN3 2.44 0.96 PRZ2    F    T    F     T     T  OC11     F     F     F     T
## 6 DGN3 2.48 1.88 PRZ1    F    F    F     T     F  OC11     F     F     F     F
##   PRE32 AGE Risk1Yr probability probability_TrueOrFalse
## 1     F  60       F  0.18600471                   FALSE
## 2     F  51       F  0.11753467                   FALSE
## 3     F  59       F  0.11857933                   FALSE
## 4     F  54       F  0.05292682                   FALSE
## 5     F  73       T  0.23654785                   FALSE
## 6     F  51       F  0.05273854                   FALSE
```

```
thoraric_compare_table <- table(actual = thoraric_df$Risk1Yr, predicted = thoraric_df$probability_TrueO
thoraric_compare_table
```

```
##       predicted
```

```
## actual FALSE TRUE
##      F   366   34
##      T    54   16
```

```
thoraric_accuracy <- ((thoraric_compare_table[[1,1]]+thoraric_compare_table[[2,2]]) / sum(thoraric_compa
thoraric_accuracy_percent <- thoraric_accuracy*100
thoraric_accuracy_percent
```

```
## [1] 81.2766
```

```
## The model is roughly 81.28% accurate.
```