HeartDisease_Model

Kevin Troncoso

4/11/2022

Logistic Model

```
#Grabbing Saved Csv file created from python
hd.df <- read.csv('HeartDisease_clean.csv', stringsAsFactors = T)

#Checking to make sure everything is how it's supposed to be
#str(hd.df)
#dim(hd.df)
```

```
set.seed(42)
train.index <- sample(c(1:dim(hd.df)[1]), dim(hd.df)[1]*0.6)
train.df <- hd.df[train.index, ]
valid.df <- hd.df[-train.index, ]

# run logistic regression
# use glm() (generalized linear model) with family = "binomial" to fit a logistic
# regression.
logit.reg <- glm(HeartDisease ~ ., data = train.df, family = "binomial")
summary(logit.reg)
```

```
##
## Call:
## glm(formula = HeartDisease ~ ., family = "binomial", data = train.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2047  -0.4185  -0.2613  -0.1475   3.4689
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.982182   0.078556 -76.152  < 2e-16 ***
## BMI               0.013502   0.001453   9.291  < 2e-16 ***
## Smoking           0.457013   0.018204  25.106  < 2e-16 ***
## AlcoholDrinking  -0.289324   0.043552  -6.643 3.07e-11 ***
## Stroke            1.147333   0.028887  39.717  < 2e-16 ***
## PhysicalHealth    0.022282   0.000966  23.067  < 2e-16 ***
## MentalHealth      0.009787   0.001121   8.730  < 2e-16 ***
## DiffWalking       0.407066   0.022986  17.709  < 2e-16 ***
## Sex               0.736602   0.018545  39.719  < 2e-16 ***
```

```
## AgeCategory       0.282363   0.003858  73.196  < 2e-16 ***
## Race              0.024226   0.008183   2.960 0.003072 **
## Diabetic          0.299656   0.010320  29.036  < 2e-16 ***
## PhysicalActivity -0.077089   0.020493  -3.762 0.000169 ***
## GenHealth        -0.030193   0.006577  -4.590 4.42e-06 ***
## SleepTime        -0.031416   0.005579  -5.631 1.79e-08 ***
## Asthma            0.347512   0.024620  14.115  < 2e-16 ***
## KidneyDisease     0.744684   0.030978  24.039  < 2e-16 ***
## SkinCancer        0.129114   0.024820   5.202 1.97e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 112665  on 191876  degrees of freedom
## Residual deviance:  89577  on 191859  degrees of freedom
## AIC: 89613
##
## Number of Fisher Scoring iterations: 6
```

$$\log \frac{\widehat{P(HeartDisease = 1|X)}}{1 - P(HeartDisease = 1|X)} = \beta_0 + \beta_1(x_1) + ... + \beta_{17}(x_{17}) + \varepsilon_i \ ,$$

where X~Bernoulli Distribution.

We can respectively replace the x values with the appropriate coefficients from summary(logit.reg). It is important to remember that because it is a logistic regression/model–in terms of interpretation–we have to define the average outcome (prediction of whether an individual has a heart disease), by its log of its coeffs. In other words, because we have a log performed on the target variable, we have to perform a natural log on both sides of the equation in order to interpret the data and coefficients.

Logistic Regression Assumptions

```
library(car)
```

```
## Loading required package: carData
```

```
car::vif(logit.reg) # really good in terms of multicolinearity because nothing is above 2
```

```
##              BMI          Smoking  AlcoholDrinking           Stroke
##         1.164010         1.035150         1.018863         1.030259
##   PhysicalHealth     MentalHealth      DiffWalking              Sex
##         1.395021         1.220200         1.424225         1.068930
##      AgeCategory             Race         Diabetic PhysicalActivity
##         1.239205         1.055830         1.118845         1.171042
##        GenHealth        SleepTime           Asthma     KidneyDisease
##         1.010863         1.040068         1.056930         1.043661
##       SkinCancer
##         1.075697
```

```r
library(arm)
```

```
## Loading required package: MASS
```

```
## Loading required package: Matrix
```

```
## Loading required package: lme4
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

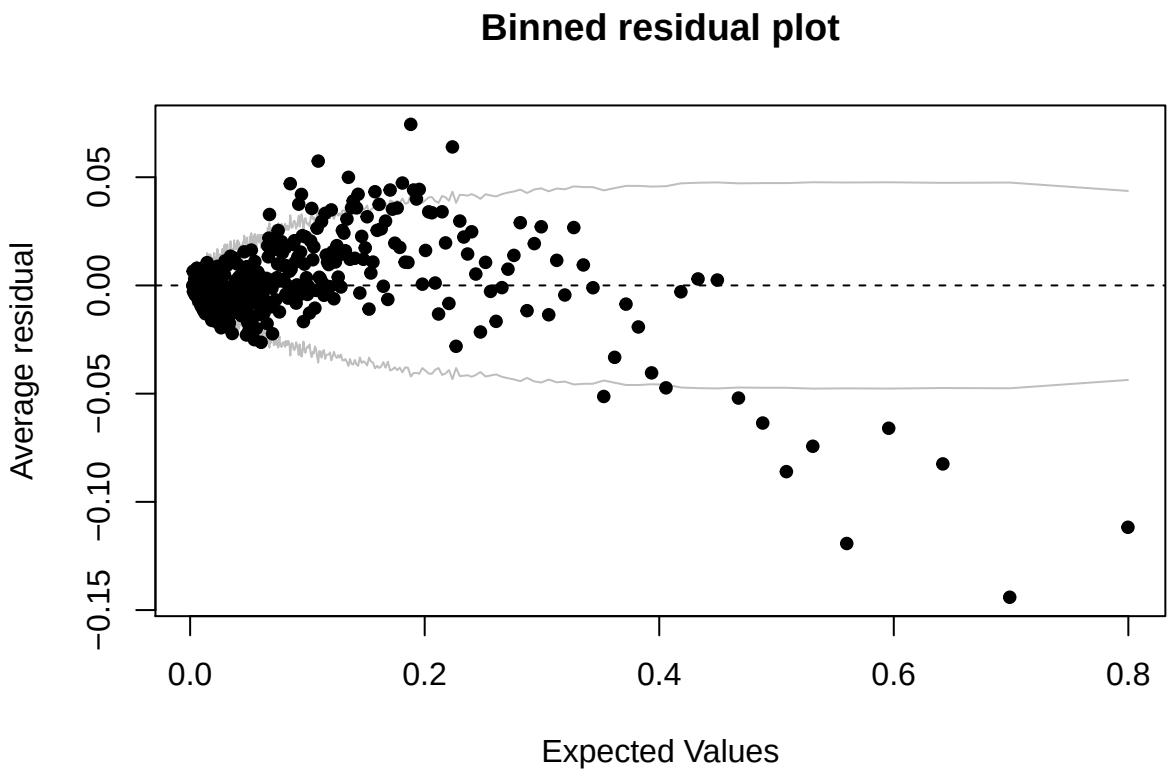
```
##
## arm (Version 1.12-2, built: 2021-10-15)
```

```
## Working directory is /Users/kevintroncoso/Desktop/Internships
```

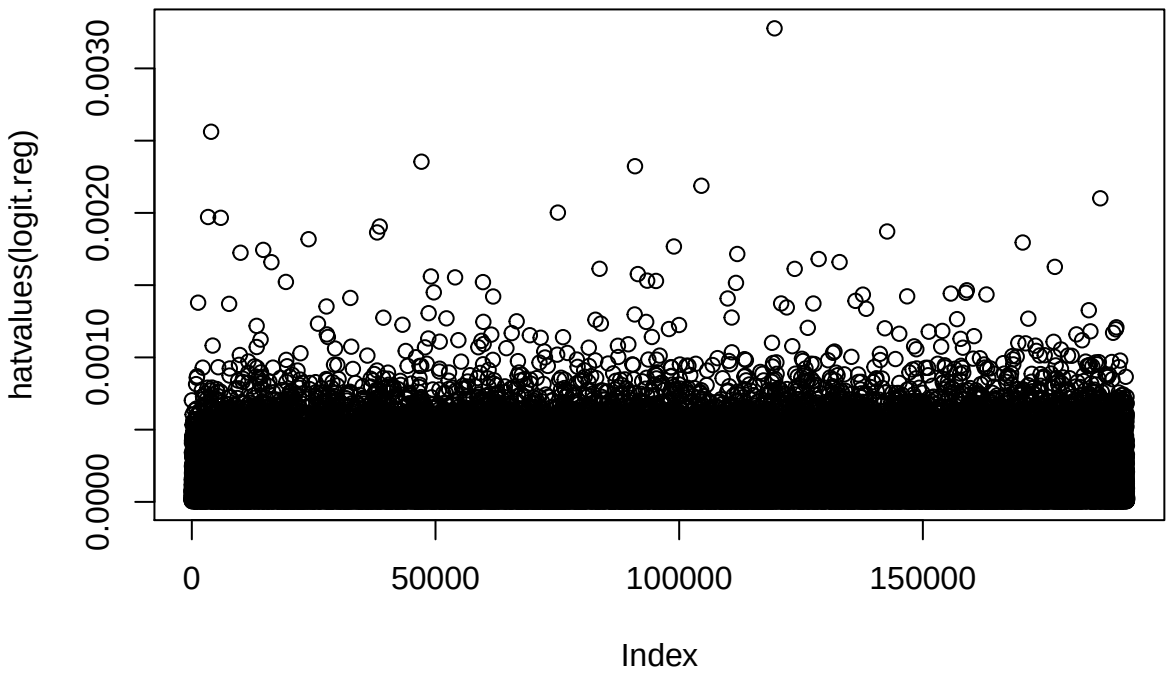
```
##
## Attaching package: 'arm'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```r
binnedplot(fitted(logit.reg),
           residuals(logit.reg, type = "response"),
           nclass = NULL,
           xlab = "Expected Values",
           ylab = "Average residual",
           main = "Binned residual plot",
           cex.pts = 0.8,
           col.pts = 1,
           col.int = "gray")
```




```
log.pred<-predict(logit.reg, valid.df, type = "response")
#plot(log.pred, which = 5)
plot(hatvalues(logit.reg))
```



Measure of accuracy and Rmse

```
#Produce the confusion matrix with cutoff=0.5, through trial and error we see its the best one.
yhat<-ifelse(log.pred > 0.5, 1,0)
y<-valid.df$HeartDisease
con.matrix<-table(yhat, y)
con.matrix
```

```
##      y
## yhat      0      1
##    0 116116   9827
##    1    963   1012
```

```
#Classification Rate: one measure of accuracy
class.rate = (con.matrix[1]+con.matrix[4])/(sum(con.matrix))
class.rate
```

```
## [1] 0.9156491
```

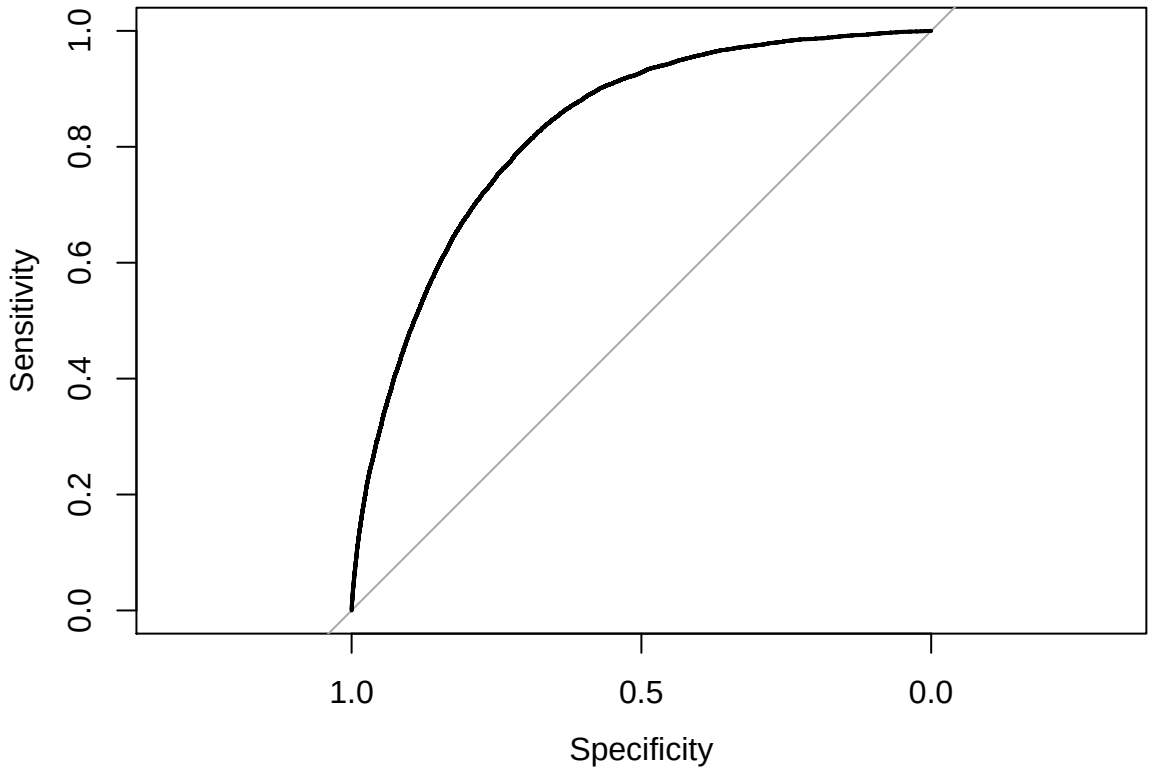
```
#Produce ROC
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
log.r<-roc(y,log.pred)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

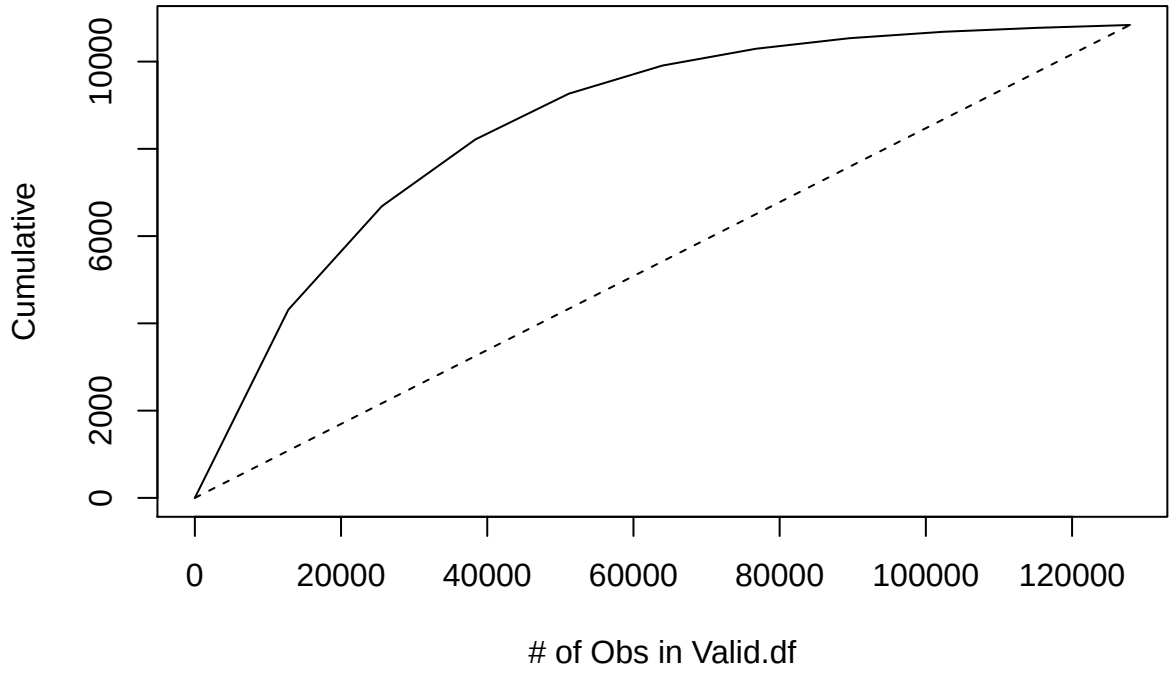
```
plot.roc(log.r)
```




```
#better then baseline model
log.auc<-auc(log.r) # 0.8273 is a really good score
log.auc
```

```
## Area under the curve: 0.8273
```

```r
#Producing a lift curve
library(gains)
gain <- gains(y, log.pred, groups=10)
plot(c(0,gain$cume.pct.of.total*sum(y))~c(0,gain$cume.obs),
     xlab="# of Obs in Valid.df", ylab="Cumulative", main="", type="l")
lines(c(0,sum(y))~c(0, dim(valid.df)[1]), lty=2)
```




```r
gain
```

```
## Depth                                    Cume   Cume Pct                    Mean
##   of            Cume      Mean      Mean   of Total    Lift   Cume      Model
## File     N       N       Resp      Resp      Resp     Index   Lift      Score
## ------------------------------------------------------------------------------
##    10 12791  12791       0.34      0.34     39.8%      398    398       0.36
##    20 12792  25583       0.19      0.26     61.7%      219    308       0.17
##    30 12792  38375       0.12      0.21     75.8%      141    253       0.11
##    40 12792  51167       0.08      0.18     85.5%       97    214       0.07
##    50 12792  63959       0.05      0.15     91.4%       59    183       0.05
##    60 12791  76750       0.03      0.13     95.0%       36    158       0.04
##    70 12792  89542       0.02      0.12     97.2%       22    139       0.03
##    80 12792 102334       0.01      0.10     98.6%       14    123       0.02
##    90 12792 115126       0.01      0.09     99.4%        8    110       0.01
##   100 12792 127918       0.01      0.08    100.0%        6    100       0.01
```

```
##### Lift Index = 308 in the output of gains() means that the top 39.8% of
##### of the data ranked by the logistic regression identifies 3.98 times
##### as many successful individuals with heart disease as would a random selection of 39.8% of the dat
```