Logistic Regression Analysis - Medical Appointments

Prediction: No Show Appointments

Participants: Sarah Fite, Matthew Przybyla, and David Tran

Summary and Statistics

```
head(Data_Sub)
```

```
##   Gender Age Scholarship Hypertension Diabetes Alcoholism Handicap
## 1      F  62           0            1        0          0        0
## 2      M  56           0            0        0          0        0
## 3      F  62           0            0        0          0        0
## 4      F   8           0            0        0          0        0
## 5      F  56           0            1        1          0        0
## 6      F  76           0            1        0          0        0
##   SMS_received NOSHOW
## 1            0     No
## 2            0     No
## 3            0     No
## 4            0     No
## 5            0     No
## 6            0     No
```

```
str(Data_Sub)
```

```
## 'data.frame':    110527 obs. of  9 variables:
##  $ Gender      : Factor w/ 2 levels "F","M": 1 2 1 1 1 1 1 1 1 1 ...
##  $ Age         : int  62 56 62 8 56 76 23 39 21 19 ...
##  $ Scholarship : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hypertension: int  1 0 0 0 1 1 0 0 0 0 ...
##  $ Diabetes    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Alcoholism  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Handicap    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SMS_received: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NOSHOW      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 2 1 1 ...
```

```
sapply(Data_Sub, sd)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling
var(x) on a factor x is deprecated and will become an error.
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling
var(x) on a factor x is deprecated and will become an error.
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.
```

```
##       Gender          Age  Scholarship Hypertension     Diabetes
##    0.4769790   23.1101759    0.2976748    0.3979213    0.2582651
##   Alcoholism     Handicap SMS_received       NOSHOW
##    0.1716856    0.1615427    0.4668727    0.4014440
```

```
xtabs(~NOSHOW + Age, data = Data_Sub)
```

```
##       Age
## NOSHOW    0    1    2    3    4    5    6    7    8    9   10   11   12
##    No  2900 1859 1366 1236 1017 1169 1205 1126 1106 1008  970  948  820
##    Yes  639  415  252  277  282  320  316  301  318  364  304  247  272
##       Age
## NOSHOW   13   14   15   16   17   18   19   20   21   22   23   24   25
##    No   800  802  889 1049 1113 1137 1151 1082 1097 1025 1006  921  980
##    Yes  303  316  322  353  396  350  394  355  355  351  343  321  352
##       Age
## NOSHOW   26   27   28   29   30   31   32   33   34   35   36   37   38
##    No   971 1048 1116 1073 1152 1119 1174 1176 1204 1089 1236 1216 1309
##    Yes  312  329  332  330  369  320  331  348  322  289  344  317  320
##       Age
## NOSHOW   39   40   41   42   43   44   45   46   47   48   49   50   51
##    No  1196 1101 1038 1007 1035 1164 1198 1177 1127 1128 1354 1322 1284
##    Yes  340  301  308  265  309  323  255  283  267  271  298  291  283
##       Age
## NOSHOW   52   53   54   55   56   57   58   59   60   61   62   63   64
##    No  1449 1332 1262 1168 1372 1325 1216 1357 1175 1143 1100 1195 1149
##    Yes  297  319  268  257  263  278  253  267  236  200  212  179  182
##       Age
## NOSHOW   65   66   67   68   69   70   71   72   73   74   75   76   77
##    No   934 1008  825  843  714  630  574  514  629  513  463  480  448
##    Yes  167  179  148  169  118   94  121  101   96   89   81   91   79
##       Age
## NOSHOW   78   79   80   81   82   83   84   85   86   87   88   89   90
##    No   452  329  430  371  326  219  276  226  218  157  114  144   86
##    Yes   89   61   81   63   66   61   35   49   42   27   12   29   23
##       Age
## NOSHOW   91   92   93   94   95   96   97   98   99  100  102  115
##    No    53   66   43   27   18   16    9    5    1    4    2    2
##    Yes   13   20   10    6    6    1    2    1    0    0    0    3
```

Split dataset into “Train” (80%) and “Test” (20%)

```
Split <- sample(2, nrow(Data_Sub), replace=TRUE, prob =c(0.8, 0.2))
Train <- Data_Sub[Split==1,]
Test <- Data_Sub[Split==2,]
```

Fitting the Model

```
model <- glm(NOSHOW ~., family=binomial(link='logit'), data=Train)
model2 <- glm(NOSHOW ~ Age + Scholarship + Hypertension + Diabetes + Alcoholism  + SMS_received,

              family=binomial(link='logit'), data=Train)
predict <- predict(model, type ='response')

summary(model)
```

```
## 
## Call:
## glm(formula = NOSHOW ~ ., family = binomial(link = "logit"), 
##     data = Train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -0.9538  -0.6843  -0.6083  -0.5372   2.1124  
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -1.3831649  0.0196638 -70.341  < 2e-16 ***
## GenderM      -0.0142421  0.0182145  -0.782  0.43427    
## Age          -0.0065303  0.0004391 -14.872  < 2e-16 ***
## Scholarship   0.1954285  0.0273091   7.156  8.3e-13 ***
## Hypertension -0.0617751  0.0274333  -2.252  0.02433 *  
## Diabetes      0.0987910  0.0380787   2.594  0.00948 ** 
## Alcoholism    0.1280998  0.0495697   2.584  0.00976 ** 
## Handicap      0.0166565  0.0545769   0.305  0.76022    
## SMS_received  0.6360021  0.0173020  36.759  < 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: 88946  on 88363  degrees of freedom
## Residual deviance: 87232  on 88355  degrees of freedom
## AIC: 87250
## 
## Number of Fisher Scoring iterations: 4
```

```
summary(model2)
```

```
## 
## Call:
## glm(formula = NOSHOW ~ Age + Scholarship + Hypertension + Diabetes + 
##     Alcoholism + SMS_received, family = binomial(link = "logit"), 
##     data = Train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -0.9525  -0.6838  -0.6072  -0.5372   2.1200  
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -1.3898475  0.0176308 -78.830  < 2e-16 ***
## Age          -0.0064831  0.0004353 -14.893  < 2e-16 ***
## Scholarship   0.1983339  0.0270540   7.331 2.28e-13 ***
## Hypertension -0.0615190  0.0274156  -2.244  0.02484 *  
## Diabetes      0.0988917  0.0380688   2.598  0.00938 ** 
## Alcoholism    0.1234178  0.0492134   2.508  0.01215 *  
## SMS_received  0.6364802  0.0172780  36.838  < 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: 88946  on 88363  degrees of freedom
## Residual deviance: 87233  on 88357  degrees of freedom
## AIC: 87247
## 
## Number of Fisher Scoring iterations: 4
```

```
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: NOSHOW
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                         88363      88946
## Gender        1     1.31     88362      88945   0.25265
## Age           1   324.73     88361      88620 < 2.2e-16 ***
## Scholarship   1    50.63     88360      88570 1.116e-12 ***
## Hypertension  1     3.61     88359      88566   0.05729 .
## Diabetes      1     3.92     88358      88562   0.04767 *
## Alcoholism    1     3.19     88357      88559   0.07391 .
## Handicap      1     0.36     88356      88558   0.54606
## SMS_received  1  1326.54     88355      87232 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(model)
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %       97.5 %
## (Intercept)  -1.421772186 -1.344690711
## GenderM      -0.049976895  0.021423687
## Age          -0.007391725 -0.005670456
## Scholarship   0.141725615  0.248780368
## Hypertension -0.115644517 -0.008104809
## Diabetes      0.023865006  0.173142209
## Alcoholism    0.030132402  0.224478586
## Handicap     -0.091898433  0.122139506
## SMS_received  0.602081037  0.669904593
```

Misclassification Rate

```
p <- predict(model2, Data_Sub)
table <- table(p, Data_Sub$Handicap)
Classification_Rate = sum(diag(table))/sum(table)
Classification_Rate
```

```
## [1] 9.047563e-06
```

```
Misclassification_Rate = 1- sum(diag(table))/sum(table)
Misclassification_Rate
```

```
## [1] 0.999991
```

Accessing the predicability of the model

```
fitted.results <- predict(model, newdata=subset(Test, select=c(1,2,3,4,5,6,7,8)), type='respons
e')
fitted.results <- ifelse(fitted.results > 0.5, 1, 0)
misClasificError <- mean(fitted.results != Test$NOSHOW)
print(paste('Accuracy', 1-misClasificError))
```

```
## [1] "Accuracy 0"
```

Model Performance Evaluation

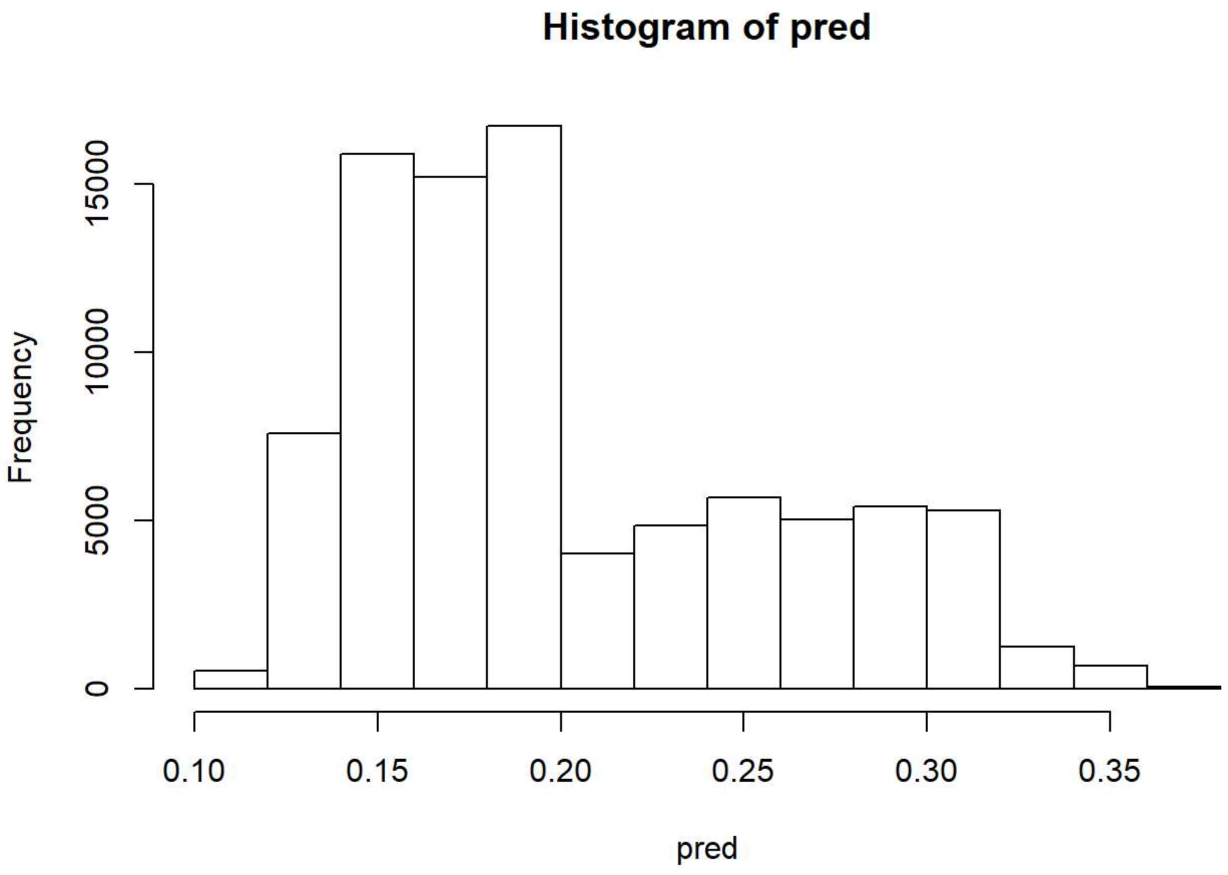
```
pred <- predict(model, Train, type= "response")
head(pred)
```

```
##         1         3         4         6         7         8 
## 0.1358943 0.1433129 0.1922576 0.1255115 0.1775024 0.1627584
```

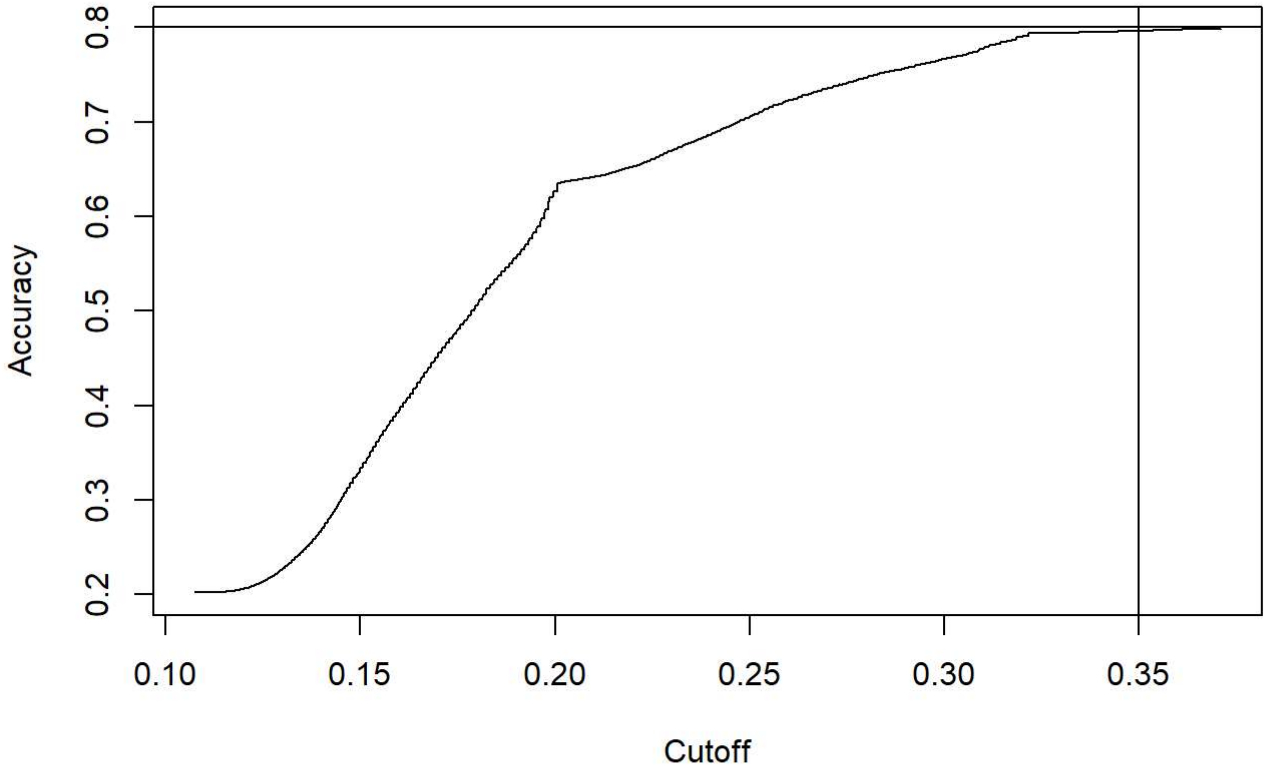
```
head(Train)
```

```
##   Gender Age Scholarship Hypertension Diabetes Alcoholism Handicap
## 1      F  62           0            1        0          0        0
## 3      F  62           0            0        0          0        0
## 4      F   8           0            0        0          0        0
## 6      F  76           0            1        0          0        0
## 7      F  23           0            0        0          0        0
## 8      F  39           0            0        0          0        0
##   SMS_received NOSHOW
## 1            0     No
## 3            0     No
## 4            0     No
## 6            0     No
## 7            0    Yes
## 8            0    Yes
```

```
hist(pred)
```

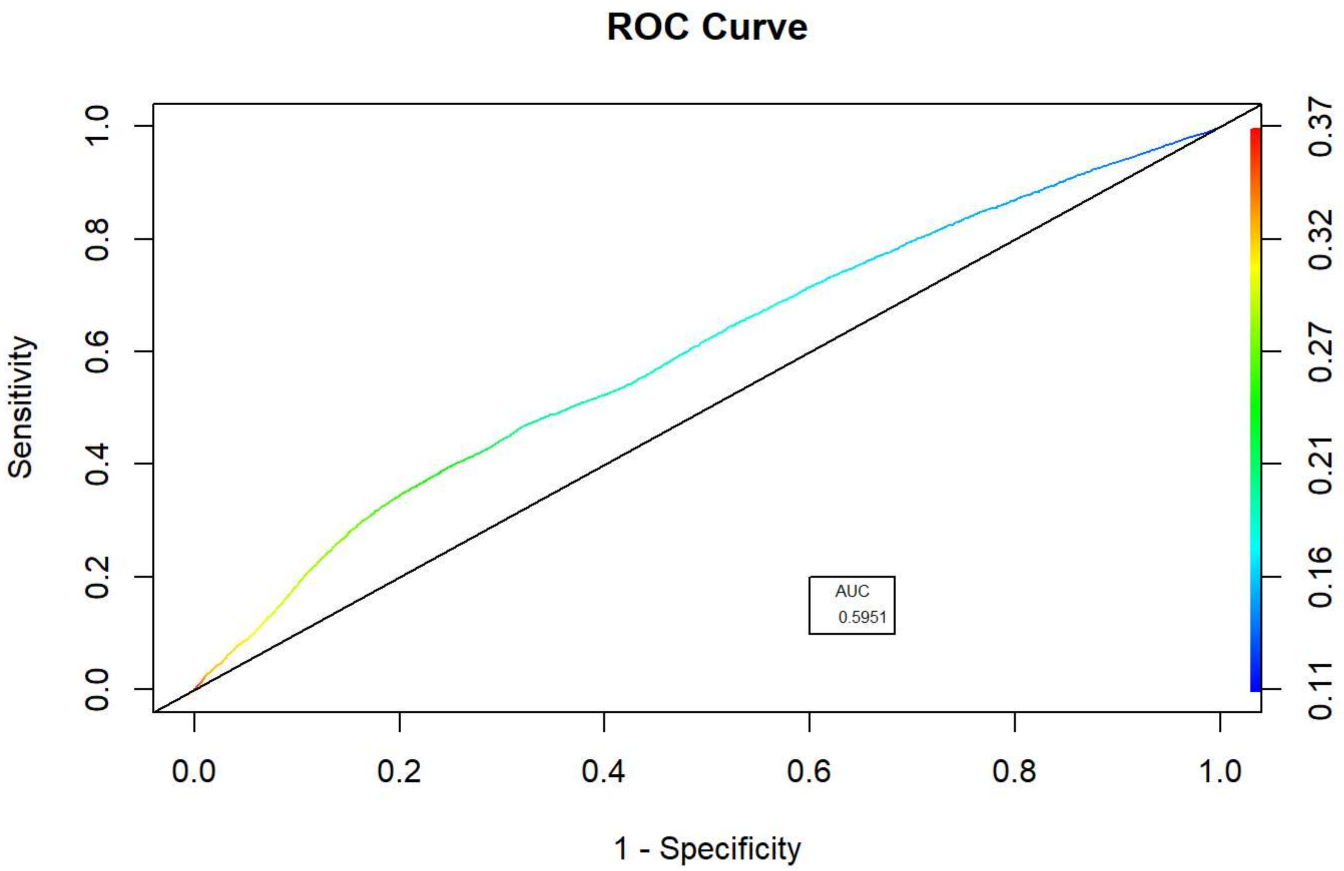


```
predf <- prediction(pred, Train$NOSHOW)
eval <- performance(predf, "acc")
plot(eval)
abline(h=0.80, v=0.35)
```



Reciever Operating Characteristic (ROC) Curve & Area Under Curve (AUC)

```
pred2 <- prediction(pred, Train$NOSHOW)
roc <- performance(pred2, "tpr", "fpr")
plot(roc,
     colorize=T,
     main = "ROC Curve",
     ylab = "Sensitivity",
     xlab = "1 - Specificity")
abline(a=0, b=1)
auc <- performance(pred2, "auc")
auc2 <- unlist(slot(auc, "y.values"))
auc <- round(auc2, 4)
legend(.6, .2, auc, title = "AUC", cex =.5)
```



Identify Best Values

```
max <- which.max(slot(eval, "y.values")[[1]])
max
```

```
## [1] 3
```

```
acc <- slot(eval, "y.values")[[1]][max]
acc
```

```
## [1] 0.7979381
```

```
cut <- slot(eval, "x.values")[[1]][max]
cut
```

```
##     32330
## 0.3694332
```