UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

BUDT758T

DATA MINING AND PREDICTIVE ANALYTICS

Homework 3

NAME (in capitals): ___________AKHIL GUPTA____

- Please submit on Canvas.
- Your submission should consist of this document (with answers filled in in the appropriate places).
- Please ensure that answers are appropriately numbered and clearly legible.
- In the space below please enter the following text and initial below: “I pledge on my honor that I have not given or received unauthorized assistance on this assignment.”

HONOR PLEDGE: I pledge on my honor that I have not given or received unauthorized assistance on this assignment.”

YOUR INITIALS: AG

This assignment is a continuation of Homework 2. The data is the same and you will follow the same steps for data preparation. In particular continue to use the same seed, and R's **sample** function to partition the data set. Thereafter you will evaluate performance of your classifier for this data set.

The Assignment

The data in the accompanying file "VoterPref.csv" (posted on Canvas) contains data from a survey of random sample of registered voters in a state. The subjects were asked whether they were "For" or "Against" a proposal on the ballot to increase the state sales tax by 0.5%, with the stipulation that the additional tax revenues be spent on education. In addition to their position on the proposition, some additional demographic information is collected. The variables in the data set are:

PREFERENCE	"For" or "Against"
AGE	Years of age at time of survey
INCOME	Annual income in thousands of US dollars
GENDER	"M" or "F"

The intent of the survey is to develop a strategy to target individuals for a marketing campaign designed to "get out the vote".

```
#install.packages("ROCR")
library(ROCR)

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

#


getwd()

## [1] "D:/Sem2/1. self/dataMininingPredictiveAnalysis/HW/3. HW3"

setwd("D:/Sem2/1. self/dataMininingPredictiveAnalysis/HW/3. HW3")
getwd()

## [1] "D:/Sem2/1. self/dataMininingPredictiveAnalysis/HW/3. HW3"
```

(1) Data Preparation

a. Read the data set in *R*. For the PREFERENCE variable ensure that "Against" is the success class

```
#1a
pref<-read.csv('VoterPref.csv')
attach(pref)
```

```r
PREFERENCE <- factor(PREFERENCE,levels=c("For","Against"))
L_PREF <- (as.numeric(PREFERENCE)-1)

pref<-cbind(pref,L_PREF)
```

b. Set the seed to 123457

```r
#1b
set.seed(123457)
```

c. Randomly partition the data set into the *training* and *test* data sets. The proportion of observations in the training data set should be 70%. The remaining 30% of observations should be in the test data set.

```r
#1c
train_ind<-sample(seq_len(nrow(pref)),size= .7*nrow(pref))
train<-pref[train_ind, ]
test<-pref[-train_ind, ]
nrow(train)

## [1] 700

nrow(test)

## [1] 300
```

(2) Run a logistic regression model of PREFERENCE on all independent variables. Use only the training data set for this. Part (a) is repetition.

a. Use a cutoff of 0.5 and do the classification. Compute the confusion matrix for both in-sample and out-of-sample predictions (using the training and test data sets respectively). [No credit]

```r
fit_logistic <- glm(as.numeric(L_PREF)~AGE+INCOME+factor(GENDER), family = "binomial",
 data = train)
summary(fit_logistic)

## 
## Call:
## glm(formula = as.numeric(L_PREF) ~ AGE + INCOME + factor(GENDER), 
##     family = "binomial", data = train)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.23799  -0.38579  -0.13440  -0.02922   2.81772  
## 
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)      0.13300    0.76992   0.173    0.863    
## AGE              0.23953    0.02462   9.729   <2e-16 ***
## INCOME          -0.13184    0.01268 -10.398   <2e-16 ***
```

```
## factor(GENDER)M -0.53005    0.27957  -1.896    0.058 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 680.71  on 699  degrees of freedom
## Residual deviance: 340.35  on 696  degrees of freedom
## AIC: 348.35
##
## Number of Fisher Scoring iterations: 7
```

```
predicted_train_logistic <- predict(fit_logistic, newdata = train, type = "response")

tlogistic_train <- ifelse(predicted_train_logistic > 0.5,1,0)

confusion_m_logistic_train<-table(as.numeric(train$L_PREF),tlogistic_train)
confusion_m_logistic_train
```

```
##    tlogistic_train
##       0   1
##   0 545  22
##   1  47  86
```

```
confusion_m_logistic_train_probability<-confusion_m_logistic_train/sum(confusion_m_log
istic_train)
confusion_m_logistic_train_probability
```

```
##    tlogistic_train
##              0          1
##   0 0.77857143 0.03142857
##   1 0.06714286 0.12285714
```

```
#***************************************#
predicted_test_logistic <- predict(fit_logistic, newdata = test, type = "response")

tlogistic_test <- ifelse(predicted_test_logistic > 0.5,1,0)

confusion_m_logistic_test<-table(as.numeric(test$L_PREF),tlogistic_test)
confusion_m_logistic_test
```

```
##    tlogistic_test
##       0   1
##   0 225  17
##   1  24  34
```

```
confusion_m_logistic_test_probability<-confusion_m_logistic_test/sum(confusion_m_logis
tic_test)
confusion_m_logistic_test_probability
```

```
##    tlogistic_test
##              0          1
##   0 0.75000000 0.05666667
##   1 0.08000000 0.11333333
```

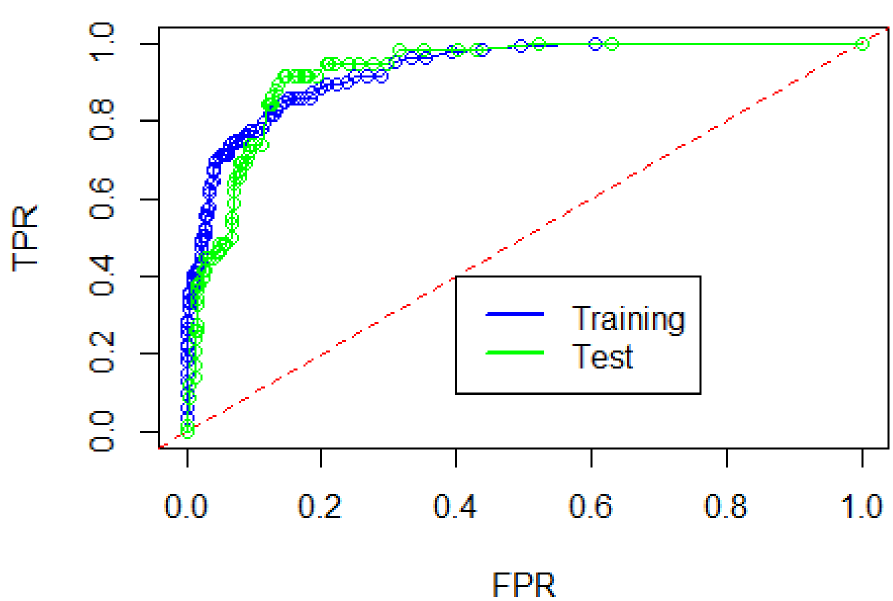
b. Compute the **sensitivity**, **specificity**, **accuracy**, **error rate**, **PPV**, **NPV**.

```
#2b.

accuracy_logistic_train<- ((confusion_m_logistic_train[1,1]+confusion_m_logistic_train
[2,2])/(confusion_m_logistic_train[1,1]+confusion_m_logistic_train[2,1]+confusion_m_lo
gistic_train[1,2]+confusion_m_logistic_train[2,2]))
accuracy_logistic_train

## [1] 0.9014286

senstivity_logistic_train<-((confusion_m_logistic_train[2,2])/(confusion_m_logistic_tr
ain[2,1]+confusion_m_logistic_train[2,2]))
senstivity_logistic_train

## [1] 0.6466165

specificity_logistic_train<-((confusion_m_logistic_train[1,1])/(confusion_m_logistic_t
rain[1,1]+confusion_m_logistic_train[1,2]))
specificity_logistic_train

## [1] 0.9611993

errorRate_logistic_train<-((confusion_m_logistic_train[1,2]+confusion_m_logistic_train
[2,1])/(confusion_m_logistic_train[1,1]+confusion_m_logistic_train[2,1]+confusion_m_lo
gistic_train[1,2]+confusion_m_logistic_train[2,2]))
errorRate_logistic_train

## [1] 0.09857143

ppv_logistic_train<-((confusion_m_logistic_train[2,2])/(confusion_m_logistic_train[2,2
]+confusion_m_logistic_train[1,2]))
ppv_logistic_train

## [1] 0.7962963

npv_logistic_train<-((confusion_m_logistic_train[1,1])/(confusion_m_logistic_train[1,1
]+confusion_m_logistic_train[2,1]))
npv_logistic_train

## [1] 0.9206081
```

c. Plot the ROC curves for both the training and test data sets on the same graph (distinguishing with different colors). What can you infer from a scrutiny of this graph? (I always find it useful to change the seed and repeat the whole analysis a few times to get a good sense).

```
#2c.
cutoff <- seq(0, 1, length = 100)
fpr_train <- numeric(100)
tpr_train <- numeric(100)

roc_table_train <- data.frame(Cutoff = cutoff, FPR = fpr_train,TPR = tpr_train)
Actual_train <- train$PREFERENCE
#Actual_train
```

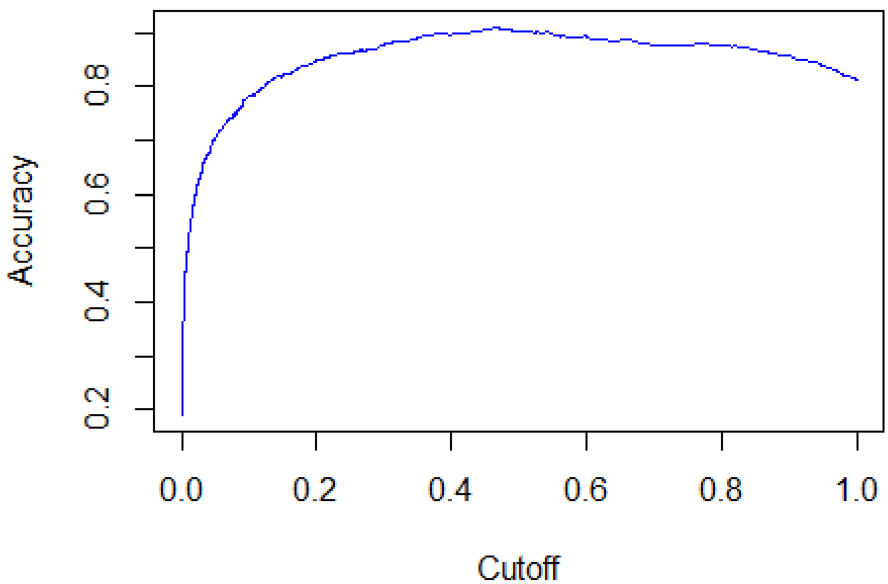
```
for (i in 1:100) {
  roc_table_train$FPR[i] <- sum(predicted_train_logistic > cutoff[i] & Actual_train ==
 "For")/sum(Actual_train == "For")
  roc_table_train$TPR[i] <- sum(predicted_train_logistic > cutoff[i] & Actual_train ==
 "Against")/sum(Actual_train == "Against")
}

plot(TPR ~ FPR, data = roc_table_train, type = "o",xlab="FPR",ylab="TPR",col="blue")
abline(a = 0, b = 1, lty = 2,col="red")

cutoff <- seq(0, 1, length = 100)
FPR_test <- numeric(100)
TPR_test <- numeric(100)
Actual_test <- test$PREFERENCE
roc_table_test <- data.frame(Cutoff = cutoff, FPR = FPR_test,TPR = TPR_test)

for (i in 1:100) {
  roc_table_test$FPR[i] <- sum(predicted_test_logistic > cutoff[i] & Actual_test == "F
or")/sum(Actual_test == "For")
  roc_table_test$TPR[i] <- sum(predicted_test_logistic > cutoff[i] & Actual_test == "A
gainst")/sum(Actual_test == "Against")
}
lines(TPR~FPR,data = roc_table_test, type="o",col="green")
legend(0.4,0.4,c("Training", "Test"),lty=c(1,1), lwd=c(2.5,2.5), col=c("blue","green")
)
```



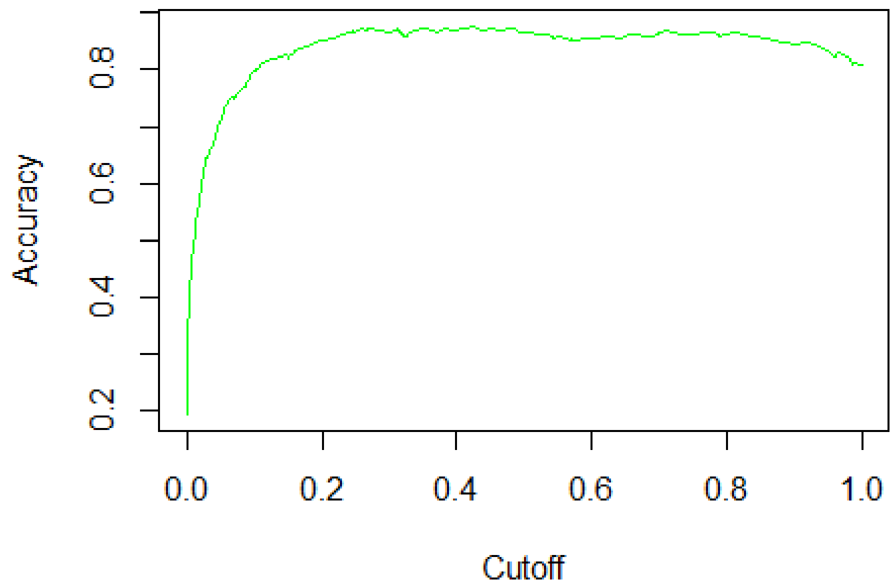
From the two ROC curves we observe that the area under the curve is almost equal. For lower FPR the Training model is slightly better but from .1 FPR to .4 FPR the test data is better. There is no overfitting.

d. Plot the accuracy against cutoff for both the training and validation data set.

```r
#2d.
pred1<-prediction(predicted_train_logistic, train$L_PREF)
perf_train <- performance(pred1, "acc")
plot( perf_train , show.spread.at=seq(0, 1, by=0.1), col="red")
```




```r
pred2<-prediction(predicted_test_logistic, test$L_PREF)
perf_test <- performance( pred2, "acc")
plot( perf_test , show.spread.at=seq(0, 1, by=0.1), col="green")
```



e. At which value of the cutoff is accuracy maximized? What is the maximum accuracy value?

```
#2e
max_accuracy_train <- max(perf_train@y.values[[1]])
max_accuracy_train

## [1] 0.91

Cutoff_train <- perf_train@x.values[[1]][which.max(perf_train@y.values[[1]])]
Cutoff_train

##       766
## 0.4625541
```

f. What is accuracy in the validation data set using the cutoff found in (e)?

```
#2f.
flag_test<-ifelse(predicted_test_logistic>0.4212197,1,0)
flag_test_table<-table(as.numeric(test$L_PREF),flag_test)
#flag_test_table
accuracy_test<-(flag_test_table[1,1]+flag_test_table[2,2])/(flag_test_table[1,1]+flag_
test_table[2,2]+flag_test_table[1,2]+flag_test_table[2,1])
accuracy_test

## [1] 0.8766667
```

(3) We use the model estimated in (2), but now include misclassification costs. Suppose that there are no costs or benefits associated with correct classification but misclassifying someone who is "For" as being "Against" has a cost of 4, whereas misclassifying someone who is "Against" as being "For" has a cost of 1.

a. What value of the cutoff minimizes misclassification cost in the training set?

```
#3a
cost <- matrix(c(0,1,4,0),nrow = 2, ncol = 2)
cost

##      [,1] [,2]
## [1,]    0    4
## [2,]    1    0

miss_cost <- performance(pred1, "cost", cost.fp = 4, cost.fn = 1)
cutoff_new <- pred1@cutoffs[[1]][which.min(miss_cost@y.values[[1]])]
cutoff_new

##        13
## 0.8219539
```

b. What is the misclassification cost in the training set? In the test set?

```
#3b.
flag_train_table <- ifelse(predicted_train_logistic > 0.8219539,1,0)

confusion_logistic_train <- table(as.numeric(train$L_PREF),flag_train_table)
confusion_logistic_train

##    flag_train_table
##       0   1
##   0 565   2
##   1  86  47

flag_test_table <- ifelse(predicted_test_logistic >0.8219539,1,0)
flag_test_table

confusion_logistic_test <- table(as.numeric(test$L_PREF),flag_test_table)
confusion_logistic_test

##    flag_test_table
##       0   1
##   0 238   4
##   1  37  21

misclassif_cost_training <- confusion_logistic_train * cost
misclassif_cost_training

##    flag_train_table
##      0  1
##   0  0  8
##   1 86  0

sum(misclassif_cost_training)

## [1] 94
```

```r
misclassif_cost_testing <- confusion_logistic_test * cost
misclassif_cost_testing

##    flag_test_table
##       0  1
##    0  0 16
##    1 37  0

sum(misclassif_cost_testing)

## [1] 53
```

c. Compare your results with the cutoff obtained in (2).

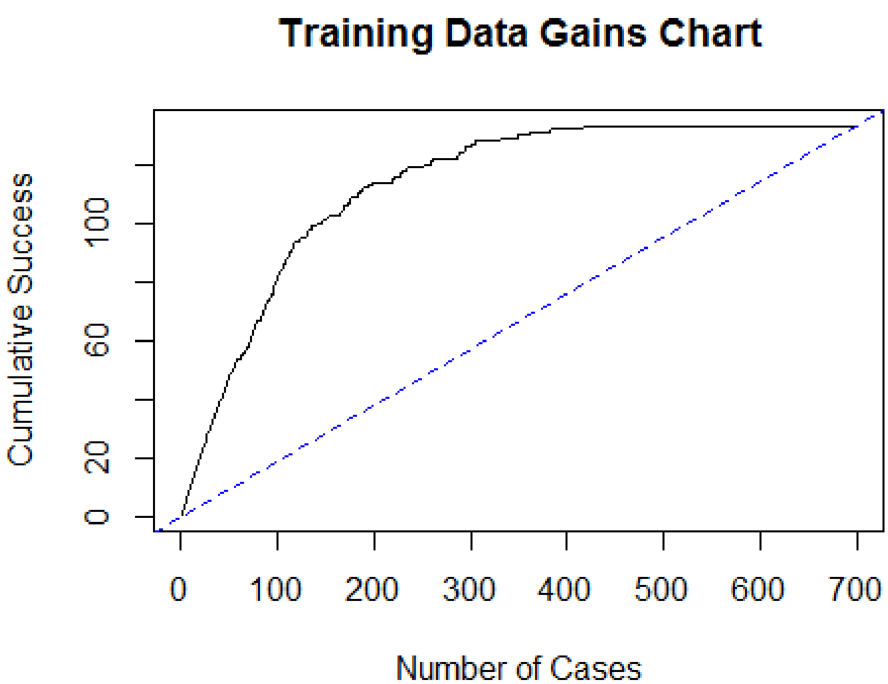
```r
#3c
flag_train_table_new <- ifelse(predicted_train_logistic > 0.4625541,1,0)

confusion_logistic_train_old <- table(as.numeric(train$L_PREF),flag_train_table_new)
confusion_logistic_train_old

##    flag_train_table_new
##        0   1
##    0 543  24
##    1  40  93

flag_test_table_new<- ifelse(predicted_test_logistic >0.4625541,1,0)
confusion_logistic_test_old <- table(as.numeric(test$L_PREF),flag_test_table_new)
confusion_logistic_test_old

##    flag_test_table_new
##        0   1
##    0 225  17
##    1 21   37

misclassification_cost_training_old <- confusion_logistic_train_old * cost
misclassification_cost_training_old

##    flag_train_table_new
##       0  1
##    0  0 96
##    1 40  0

sum(misclassification_cost_training_old)

## [1] 136

misclassification_cost_testing_old <- confusion_logistic_test_old * cost
misclassification_cost_testing_old

##    flag_test_table_new
##        0   1
##    1   0   68
##    2   21  0

sum(misclassification_cost_testing_old)
```

```
## [1] 89
```

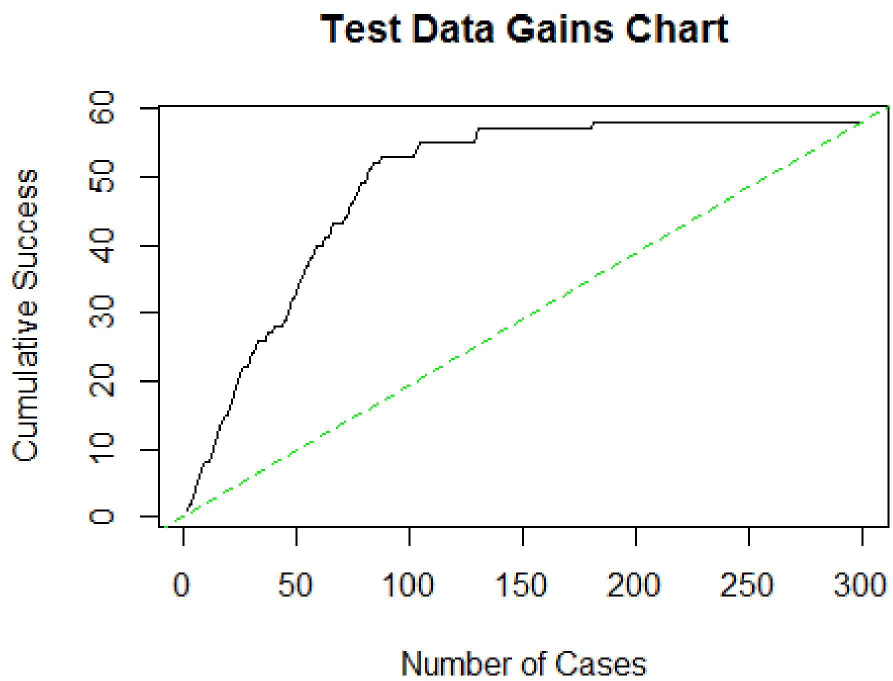
We observe that the misclassification cost for both the training and testing data sets increase, which proves that model where the cutoff is 0.82 is better than the model with cutoff 0.46.

(4) Using the model estimated in (2), plot the training data lift curve/gains chart as well as the validation data lift curve/gains chart. *Use different graphs for the two data sets.*

```
#Training Data
actual <- train$L_PREF
df_train <- data.frame(predicted_train_logistic,actual)
df_train_sort <- df_train[order(-predicted_train_logistic),]
df_train_sort$Gains <- cumsum(df_train_sort$actual)
plot(df_train_sort$Gains,type="n",main="Training Data Gains Chart",xlab="Number of Cas
es",ylab="Cumulative Success")
lines(df_train_sort$Gains)
abline(0,sum(df_train_sort$actual)/nrow(df_train_sort),lty = 2, col="blue")
```




```
########################################################################################

#Test Data
actual <- test$L_PREF
df_test <- data.frame(predicted_test_logistic,actual)
df_test_sort <- df_test[order(-predicted_test_logistic),]
df_test_sort$Gains <- cumsum(df_test_sort$actual)
plot(df_test_sort$Gains,type="n",main="Validation Data Gains Chart",xlab="Number of Ca
ses",ylab="Cumulative Success")
lines(df_test_sort$Gains)
abline(0,sum(df_test_sort$actual)/nrow(df_test_sort),lty = 2, col="green")
```



Notes: You may find it useful to look at the scripts for the beer data that I have posted online. Most likely you will be able to reuse some code from there. There are some basic control loops (**for**, **ifelse**) and some useful functions (**whichmax**, **max)**. I have posted some slides on basic R programming for reference.