Logistic_Regression1

author: Dr. Marko Mitic

Problem Description: **PREDICTING PAROLE VIOLATORS**

In many criminal justice systems around the world, inmates deemed not to be a threat to society are released from prison under the parole system prior to completing their sentence. They are still considered to be serving their sentence while on parole, and they can be returned to prison if they violate the terms of their parole.

Parole boards are charged with identifying which inmates are good candidates for release on parole. They seek to release inmates who will not commit additional crimes after release. In this problem, we will build and validate a model that predicts if an inmate will violate the terms of his or her parole. Such a model could be useful to a parole board when deciding to approve or deny an application for parole.

For this prediction task, we will use data from the United States 2004 National Corrections Reporting Program http://www.icpsr.umich.edu/icpsrweb/NACJD/series/38/studies/26521?archive=NACJD&sortBy=7, a nationwide census of parole releases that occurred during 2004. We limited our focus to parolees who served no more than 6 months in prison and whose maximum sentence for all charges did not exceed 18 months. The dataset contains all such parolees who either successfully completed their term of parole during 2004 or those who violated the terms of their parole during that year. The dataset contains the following variables:

-**male**: 1 if the parolee is male, 0 if female

-**race**: 1 if the parolee is white, 2 otherwise

-**age**: the parolee's age (in years) when he or she was released from prison

-**state**: a code for the parolee's state. 2 is Kentucky, 3 is Louisiana, 4 is Virginia, and 1 is any other state. The three states were selected due to having a high representation in the dataset.

-**time.served**: the number of months the parolee served in prison (limited by the inclusion criteria to not exceed 6 months).

-**max.sentence**: the maximum sentence length for all charges, in months (limited by the inclusion criteria to not exceed 18 months).

-**multiple.offenses**: 1 if the parolee was incarcerated for multiple offenses, 0 otherwise.

-**crime**: a code for the parolee's main crime leading to incarceration. 2 is larceny, 3 is drug-related crime, 4 is driving-related crime, and 1 is any other crime.

-**violator**: 1 if the parolee violated the parole, and 0 if the parolee completed the parole without violation.

First, let us observe the structure of the data:

```r
parole = read.csv("parole.csv")
str(parole)
```

```
## 'data.frame':    675 obs. of  9 variables:
##  $ male             : int  1 0 1 1 1 1 1 0 0 1 ...
##  $ race             : int  1 1 2 1 2 2 1 1 1 2 ...
##  $ age              : num  33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...
##  $ state            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ time.served      : num  5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...
##  $ max.sentence     : int  18 12 12 18 12 18 18 12 13 12 ...
##  $ multiple.offenses: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ crime            : int  4 3 3 1 1 4 3 1 3 2 ...
##  $ violator         : int  0 0 0 0 0 0 0 0 0 0 ...
```

We can observe that the dataframe has 675 parolees in total. The number of the parolees who violated the terms of their parole is:

```
table(parole$violator)
```

```
## 
##   0   1 
## 597  78
```

Statistical summary of the dataframe can be obtained using `summary` command:

```
summary(parole)
```

```
##       male             race            age            state      
##  Min.   :0.0000   Min.   :1.000   Min.   :18.40   Min.   :1.000  
##  1st Qu.:1.0000   1st Qu.:1.000   1st Qu.:25.35   1st Qu.:2.000  
##  Median :1.0000   Median :1.000   Median :33.70   Median :3.000  
##  Mean   :0.8074   Mean   :1.424   Mean   :34.51   Mean   :2.887  
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:42.55   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :2.000   Max.   :67.00   Max.   :4.000  
##   time.served     max.sentence   multiple.offenses     crime      
##  Min.   :0.000   Min.   : 1.00   Min.   :0.0000    Min.   :1.000  
##  1st Qu.:3.250   1st Qu.:12.00   1st Qu.:0.0000    1st Qu.:1.000  
##  Median :4.400   Median :12.00   Median :1.0000    Median :2.000  
##  Mean   :4.198   Mean   :13.06   Mean   :0.5363    Mean   :2.059  
##  3rd Qu.:5.200   3rd Qu.:15.00   3rd Qu.:1.0000    3rd Qu.:3.000  
##  Max.   :6.000   Max.   :18.00   Max.   :1.0000    Max.   :4.000  
##     violator     
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.1156  
##  3rd Qu.:0.0000  
##  Max.   :1.0000
```

Before building a model, we need to conver unordered factors in datase into factor variables. This can easily be done with `as.factor()` as follows:

```
parole$state=as.factor(parole$state)
parole$crime=as.factor(parole$crime)
```

Next, we can devide our dataset into training and testing set (70/30 ratio). In order to enable that our results are reproducible, we firstly set the seed.

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

```
## Warning: package 'caTools' was built under R version 3.2.1
```

```
set.seed(144)
split = sample.split(parole$violator, SplitRatio = 0.7)
train = subset(parole, split == TRUE)
test = subset(parole, split == FALSE)
```

We can now develop our first logistic regression model

```
model1 = glm(violator~., data=train, family="binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = violator ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7041  -0.4236  -0.2719  -0.1690   2.8375
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -4.2411574  1.2938852  -3.278  0.00105 **
## male               0.3869904  0.4379613   0.884  0.37690
## race               0.8867192  0.3950660   2.244  0.02480 *
## age               -0.0001756  0.0160852  -0.011  0.99129
## state2             0.4433007  0.4816619   0.920  0.35739
## state3             0.8349797  0.5562704   1.501  0.13335
## state4            -3.3967878  0.6115860  -5.554 2.79e-08 ***
## time.served       -0.1238867  0.1204230  -1.029  0.30359
## max.sentence       0.0802954  0.0553747   1.450  0.14705
## multiple.offenses  1.6119919  0.3853050   4.184 2.87e-05 ***
## crime2             0.6837143  0.5003550   1.366  0.17180
## crime3            -0.2781054  0.4328356  -0.643  0.52054
## crime4            -0.0117627  0.5713035  -0.021  0.98357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 340.04  on 472  degrees of freedom
## Residual deviance: 251.48  on 460  degrees of freedom
## AIC: 277.48
##
## Number of Fisher Scoring iterations: 6
```

From `summary` command we can observe the signifact variables (for which p-value is under 0.05). We can then caluclate predictions

```
pred = predict(model1, newdata=test, type="response")
```

Maximum predicted probability of violation can be easily calculated

```
max(pred)
```

```
## [1] 0.9072791
```

Accuracy, sensitivity and specificity of the model are obtained from confusion matrix:

```
table(test$violator, pred>0.5)
```

```
##
##     FALSE TRUE
##   0   167   12
##   1    11   12
```

```
ACC=(167+12)/(167+12+11+12)
ACC
```

```
## [1] 0.8861386
```

```
SE=12/(11+12)
SE
```

```
## [1] 0.5217391
```

```
SP=167/(167+12)
SP
```

```
## [1] 0.9329609
```

We can now compary the accuracy of our model with baseline predictions:

```
table(test$violator) #baseline
```

```
##
##   0   1
## 179  23
```

```
179/(179+23)
```

```
## [1] 0.8861386
```

One of the way of improving our model is evident from the confusion matrix. The parole board assigns more cost to a false negative than a false positive, and should therefore use a logistic regression cutoff less than 0.5. Althoough the model is likely of value to the board, and using a different logistic regression cutoff is likely to improve the model's value.

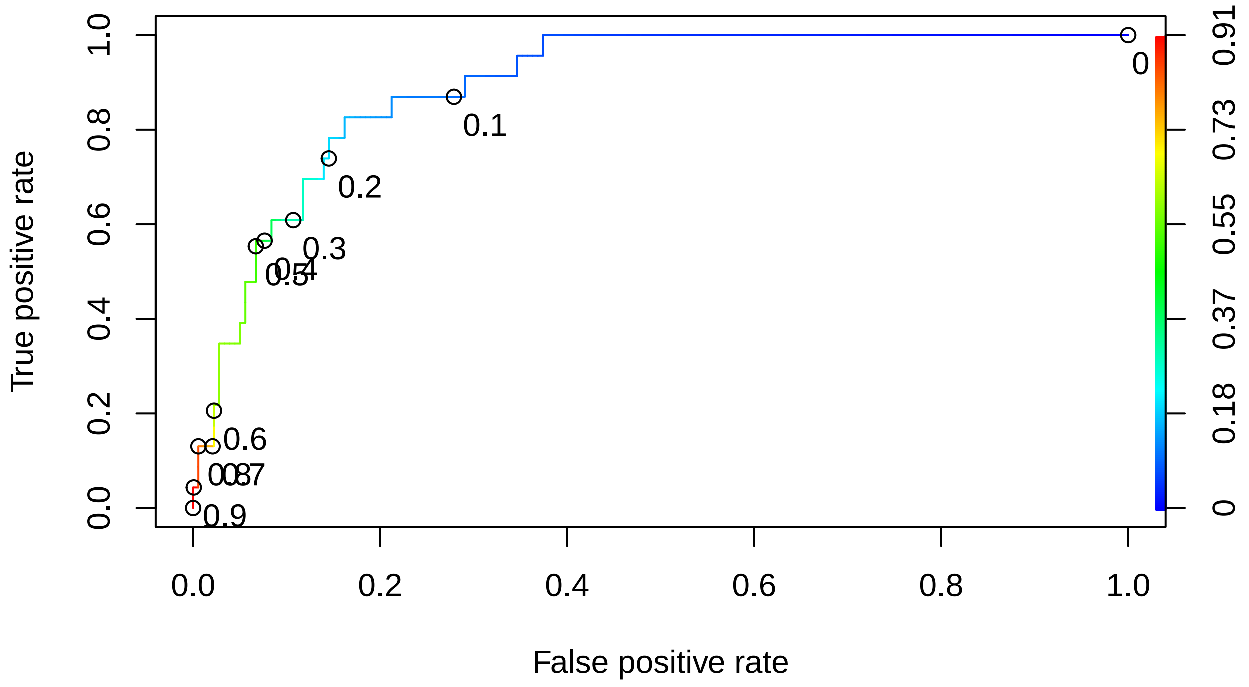
If the board used the model for parole decisions, a negative prediction would lead to a prisoner being granted parole, while a positive prediction would lead to a prisoner being denied parole. The parole board would experience more regret for releasing a prisoner who then violates parole (a negative prediction that is actually positive, or false negative) than it would experience for denying parole to a prisoner who would not have violated parole (a positive prediction that is actually negative, or false positive).Decreasing the cutoff leads to more positive predictions, which increases false positives and decreases false negatives. Meanwhile, increasing the cutoff leads to more negative predictions, which increases false negatives and decreases false positives. The parole board assigns high cost to falsenegatives, and therefore should decrease the cutoff.

Finally, we can plot ROCR (ratio between false positives and true positives), and calculate AUC (Area Under the Curve) using `ROCR` package:

```r
library("ROCR")
```

```
## Warning: package 'ROCR' was built under R version 3.2.1

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.2.1

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

```r
ROCpred = prediction(pred, test$violator)
ROCRperf=performance(ROCpred, "tpr", "fpr")

plot(ROCRperf, colorize=T, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
```



```r
AUC = as.numeric(performance(ROCpred, "auc")@y.values)
AUC
```

```
## [1] 0.8945834
```