DATA 621 Homework 3

Critical Thinking Group 1

November 07, 2021

Contents

Prepared for:

Prof. Dr. Nasrin Khansari
City University of New York, School of Professional Studies - Data 621

DATA 621 – Business Analytics and Data Mining

Home Work 3

Prepared by:

Critical Thinking Group 1

Vic Chan
Gehad Gad
Evan McLaughlin
Bruno de Melo
Anjal Hussan
Zhouxin Shi
Sie Siong Wong

Introduction

Crime has a high cost to all parts of society and it can have severe long term impact on neighborhoods. If crime rises in the neighborhood, it affects the neighborhood. Additionally, crime can even have a health cost to the community in that the perception of a dangerous neighborhood was associated with significantly lower odds of having high physical activity among both men and women. It is important to understand the propensity for crime levels of a neighborhood before investing in that neighborhood.

Statement of the Problem

The purpose of this report is to develop a binary logistic regression model to determine if the neighborhood will be at risk for high crime level.

Data Exploration

Let's take a look to the first few rows of our train data set

```
##    zn indus chas   nox    rm   age    dis rad tax ptratio lstat medv target
## 1   0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7  3.70 50.0      1
## 2   0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 26.82 13.4      1
## 3   0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 18.85 15.4      1
## 4  30  4.93    0 0.428 6.393   7.8 7.0355   6 300    16.6  5.19 23.7      0
## 5   0  2.46    0 0.488 7.155  92.2 2.7006   3 193    17.8  4.82 37.9      0
## 6   0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9  7.67 26.5      0
## 7   0 18.10    0 0.693 5.453 100.0 1.4896  24 666    20.2 30.59  5.0      1
## 8   0 18.10    0 0.693 4.519 100.0 1.6582  24 666    20.2 36.98  7.0      1
## 9   0  5.19    0 0.515 6.316  38.1 6.4584   5 224    20.2  5.68 22.2      0
## 10 80  3.64    0 0.392 5.876  19.1 9.2203   1 315    16.4  9.25 20.9      0
```

Looks like all the columns are numerical. The target variable is a binary variable indicating if the crime rate above the median rate (1) or not (0)

Means

Column means of our train data set are as follows:

```
##           zn        indus         chas          nox           rm          age
##  11.57725322  11.10502146   0.07081545   0.55431052   6.29067382  68.36759657
##          dis          rad          tax      ptratio        lstat         medv
##   3.79569292   9.53004292 409.50214592  18.39849785  12.63145923  22.58927039
##       target
##   0.49141631
```

Standard Deviation

Now let's take a look at the standard deviation of our predictor variables:

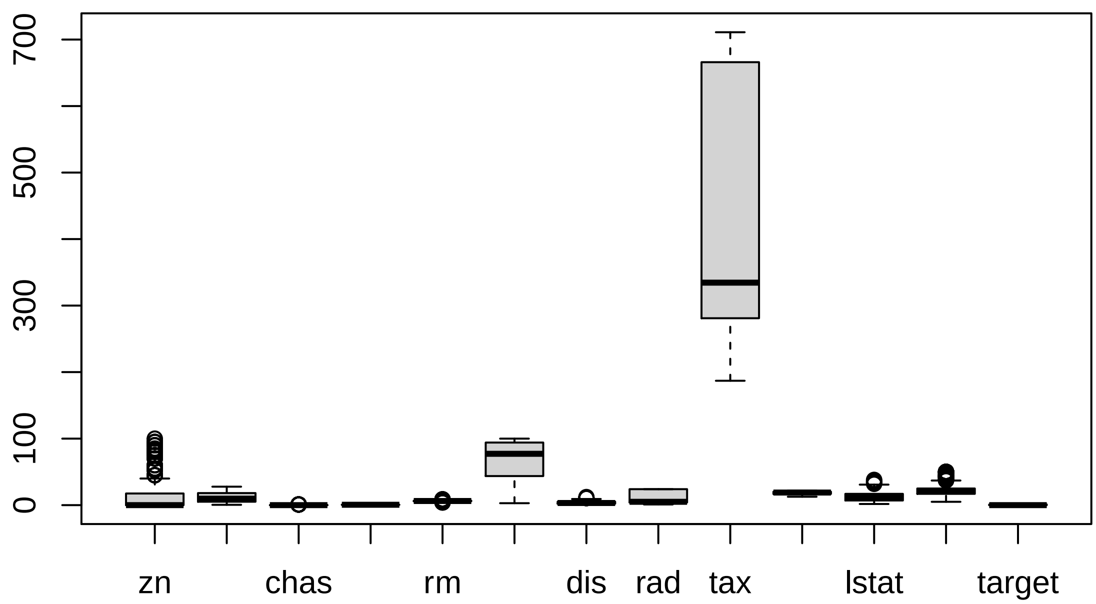
```
##          zn       indus        chas         nox          rm         age 
##  23.3646511   6.8458549   0.2567920   0.1166667   0.7048513  28.3213784 
##         dis         rad         tax     ptratio       lstat        medv 
##   2.1069496   8.6859272 167.9000887   2.1968447   7.1018907   9.2396814 
##      target 
##   0.5004636
```

Median Value

Let's take a look at the median value of our predictor variables:

```
##        zn     indus      chas       nox        rm       age       dis       rad 
##   0.00000   9.69000   0.00000   0.53800   6.21000  77.15000   3.19095   5.00000 
##       tax   ptratio     lstat      medv    target 
## 334.50000  18.90000  11.35000  21.20000   0.00000
```

Bar chart or box plot

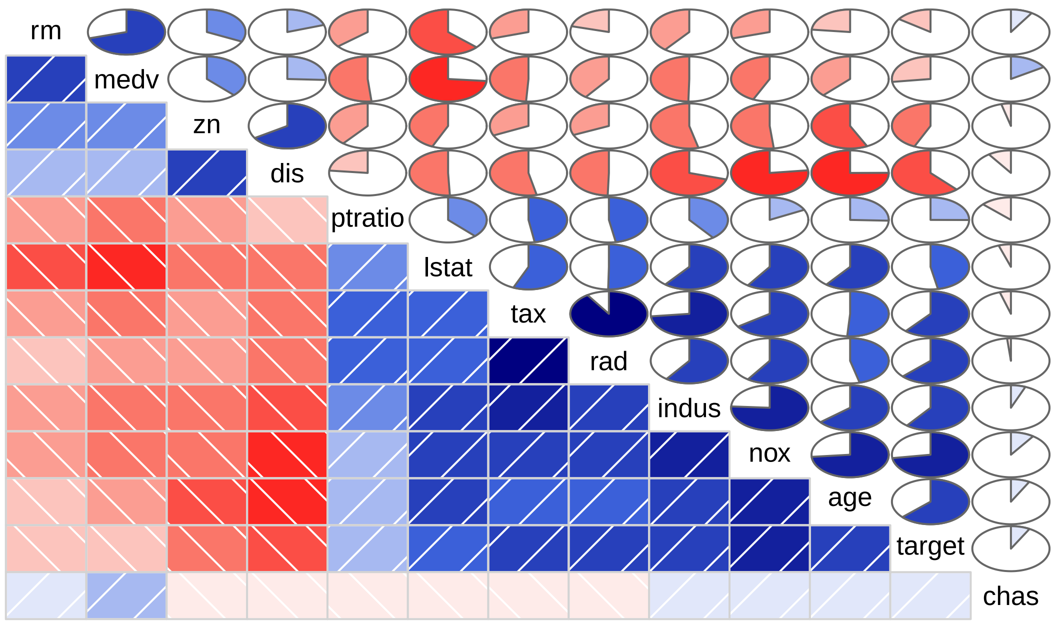


Correlation matrix

```
##                 zn       indus        chas         nox          rm         age
## zn       1.00000000 -0.53826643 -0.04016203 -0.51704518  0.31981410 -0.57258054
## indus   -0.53826643  1.00000000  0.06118317  0.75963008 -0.39271181  0.63958182
```

```
## chas    -0.04016203  0.06118317  1.00000000  0.09745577  0.09050979  0.07888366
## nox     -0.51704518  0.75963008  0.09745577  1.00000000 -0.29548972  0.73512782
## rm       0.31981410 -0.39271181  0.09050979 -0.29548972  1.00000000 -0.23281251
## age     -0.57258054  0.63958182  0.07888366  0.73512782 -0.23281251  1.00000000
## dis      0.66012434 -0.70361886 -0.09657711 -0.76888404  0.19901584 -0.75089759
## rad     -0.31548119  0.60062839 -0.01590037  0.59582984 -0.20844570  0.46031430
## tax     -0.31928408  0.73222922 -0.04676476  0.65387804 -0.29693430  0.51212452
## ptratio -0.39103573  0.39468980 -0.12866058  0.17626871 -0.36034706  0.25544785
## lstat   -0.43299252  0.60711023 -0.05142322  0.59624264 -0.63202445  0.60562001
## medv     0.37671713 -0.49617432  0.16156528 -0.43012267  0.70533679 -0.37815605
## target  -0.43168176  0.60485074  0.08004187  0.72610622 -0.15255334  0.63010625
##                 dis         rad         tax    ptratio       lstat       medv
## zn       0.66012434 -0.31548119 -0.31928408 -0.3910357 -0.43299252  0.3767171
## indus   -0.70361886  0.60062839  0.73222922  0.3946898  0.60711023 -0.4961743
## chas    -0.09657711 -0.01590037 -0.04676476 -0.1286606 -0.05142322  0.1615653
## nox     -0.76888404  0.59582984  0.65387804  0.1762687  0.59624264 -0.4301227
## rm       0.19901584 -0.20844570 -0.29693430 -0.3603471 -0.63202445  0.7053368
## age     -0.75089759  0.46031430  0.51212452  0.2554479  0.60562001 -0.3781560
## dis      1.00000000 -0.49499193 -0.53425464 -0.2333394 -0.50752800  0.2566948
## rad     -0.49499193  1.00000000  0.90646323  0.4714516  0.50310125 -0.3976683
## tax     -0.53425464  0.90646323  1.00000000  0.4744223  0.56418864 -0.4900329
## ptratio -0.23333940  0.47145160  0.47442229  1.0000000  0.37735605 -0.5159153
## lstat   -0.50752800  0.50310125  0.56418864  0.3773560  1.00000000 -0.7358008
## medv     0.25669476 -0.39766826 -0.49003287 -0.5159153 -0.73580078  1.0000000
## target  -0.61867312  0.62810492  0.61111331  0.2508489  0.46912702 -0.2705507
##              target
## zn      -0.43168176
## indus    0.60485074
## chas     0.08004187
## nox      0.72610622
## rm      -0.15255334
## age      0.63010625
## dis     -0.61867312
## rad      0.62810492
## tax      0.61111331
## ptratio  0.25084892
## lstat    0.46912702
## medv    -0.27055071
## target   1.00000000
```

visualize the data in correlation matrices

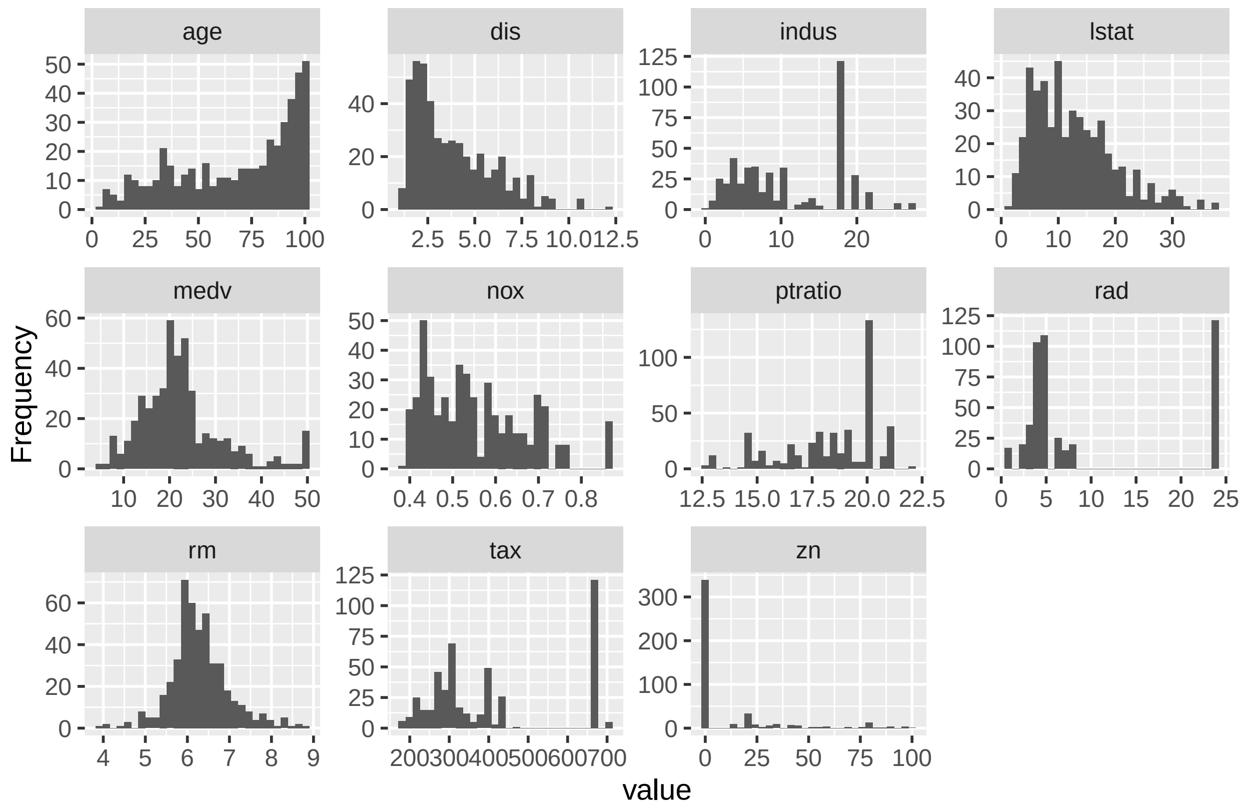


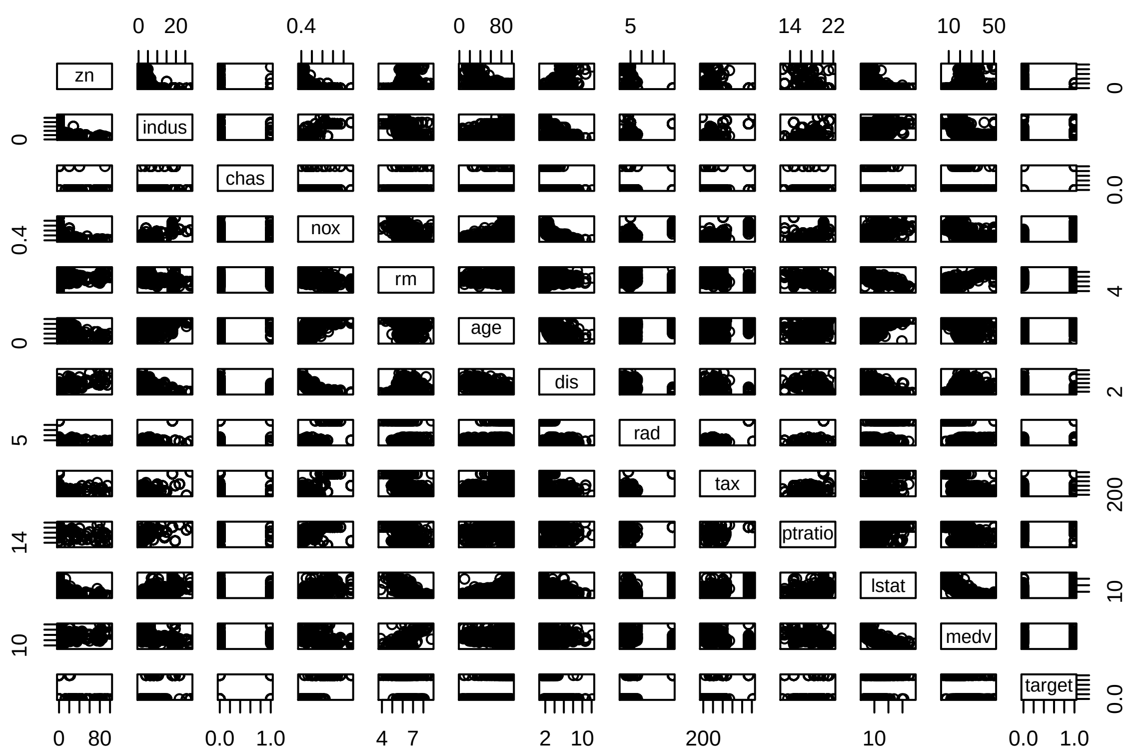
Compare Target in Training

We make sure there are no issues with an inappropriate distribution of the target variable in our training data.

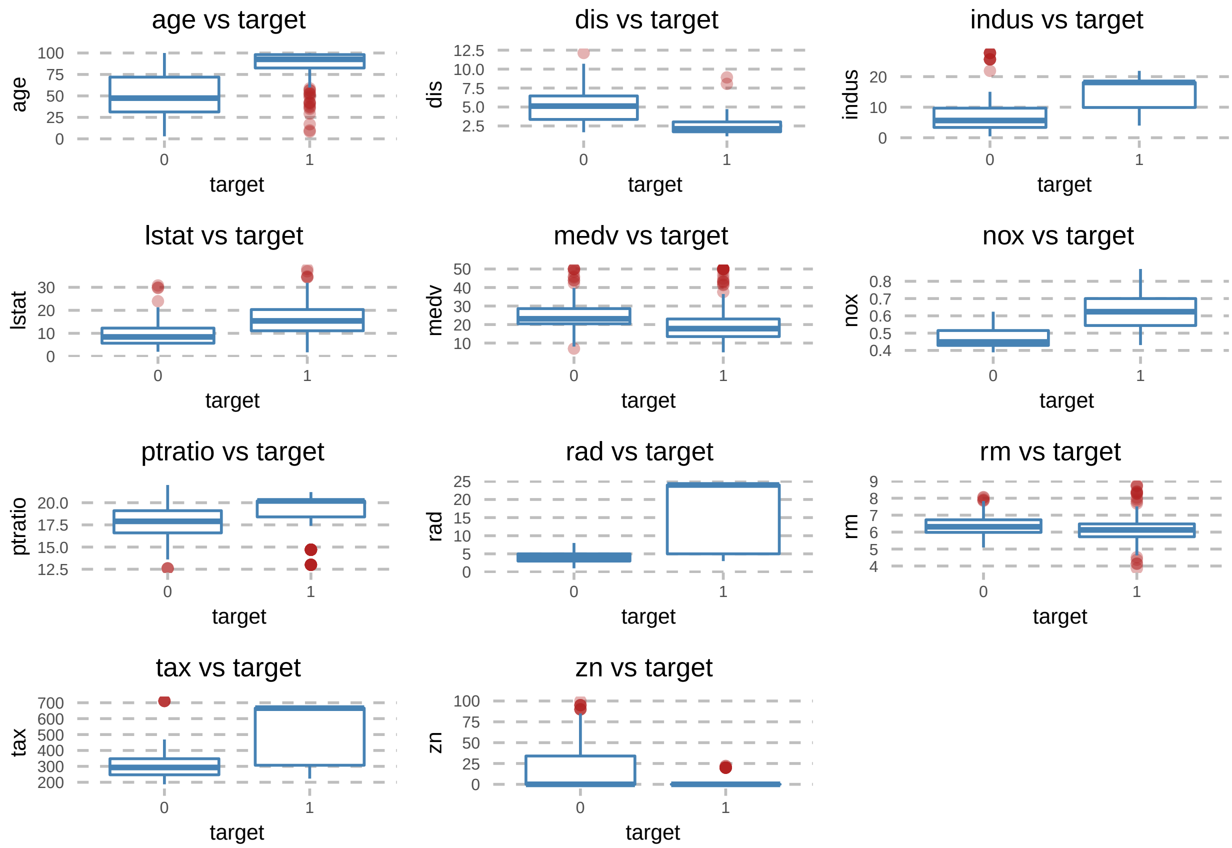
Var1	Freq
0	237
1	229

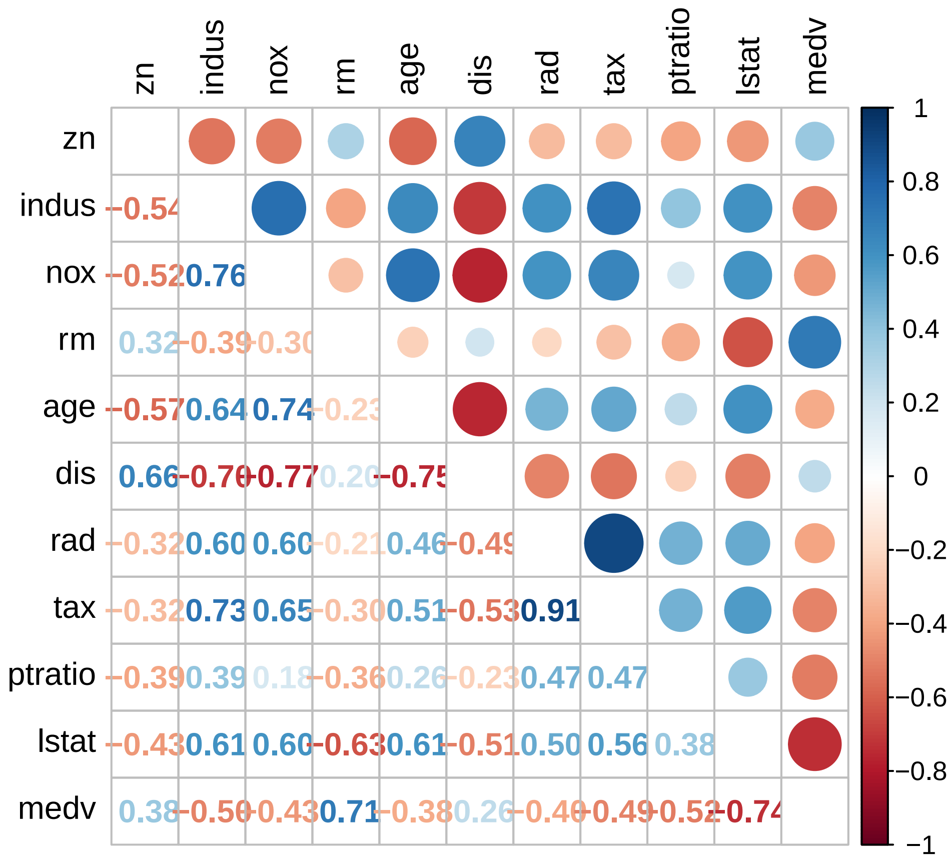
Histogram of Variables





Now that we have a basic familiarity with our data, we can analyze the relationship between the numeric variables we've brought in and the target variable. We can employ boxplots and a correlation matrix to quickly analyze this, including paired plots of the numeric feature variables.





There are a couple of items to note in the above graphics. First, in the boxplots, we observe many outliers, which could impact our regression, limiting its predictive value. Age, nox, and dis all appear to be highly correlated with our target, and numerous other features appear to have some weaker correlative relationship. Now that we've assessed the relationship between our features and the target, we can take a quick look, through our correlation matrix, at the relationship between the variables themselves. Our correlation matrix makes clear that multicollinearity is a potential issue within our observations, and we need to keep this in mind as we create and select our models.

Data Preparation

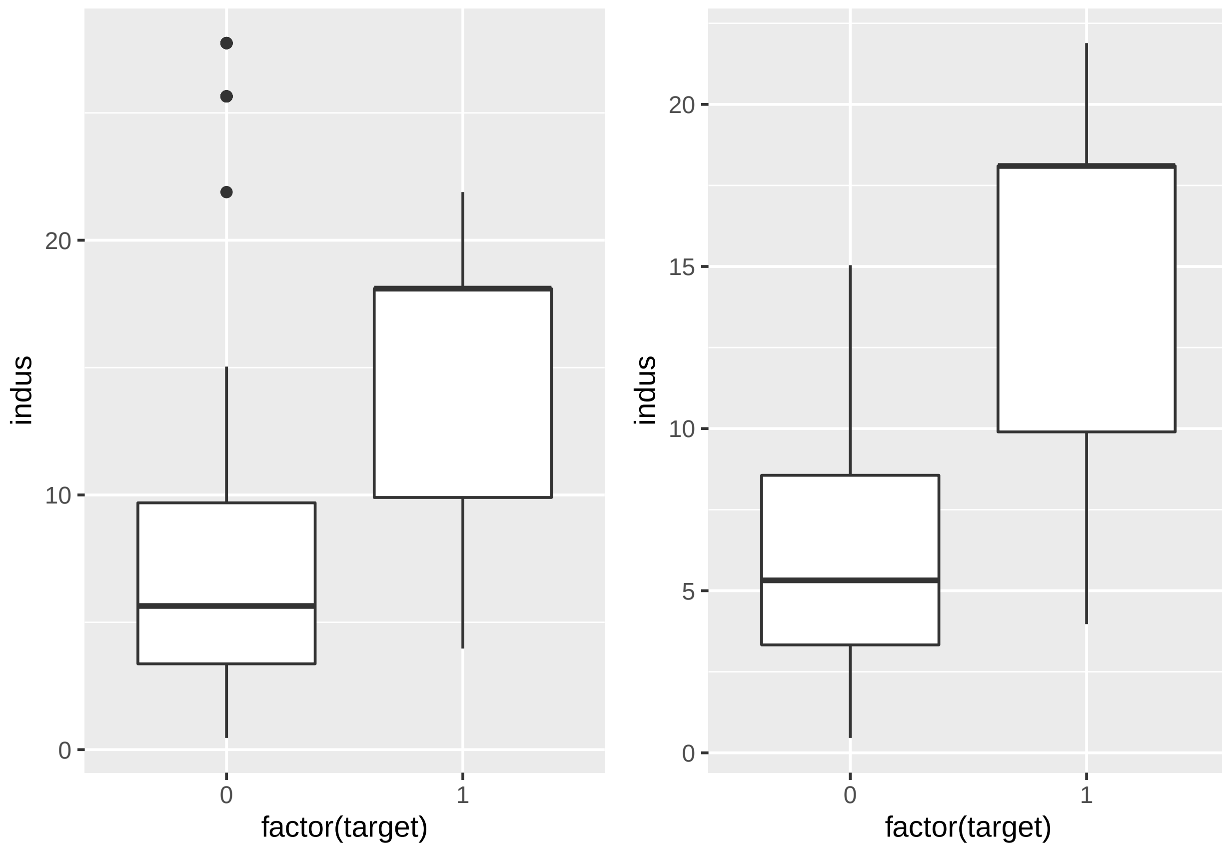
Looking at the results from our chas variable it doesn't seem to be needed here so we can remove it.

```
test_url <- 'https://raw.githubusercontent.com/ahussan/DATA_621_Group1/main/HW3/crime-evaluation-data_mo
test <- read.csv(test_url, header=TRUE)
trainchas <- as.factor(train$chas)
train$chas <- NULL
traintarget <- as.factor(train$target)
train$target <- traintarget
testchas <- as.factor(test$chas)
test$chas <- NULL
```

Indus

We see a lot of outliers in the indus variable, so we'll removed the rows which indus is greater than 20 and target is 0.

```
attach(train)
p0 <- ggplot(train, aes(factor(target), indus)) + geom_boxplot()
train <- train[-which(target==0 & indus > 20),]
p1 <- ggplot(train, aes(factor(target), indus)) + geom_boxplot()
grid.arrange(p0, p1,ncol=2,nrow=1)
```

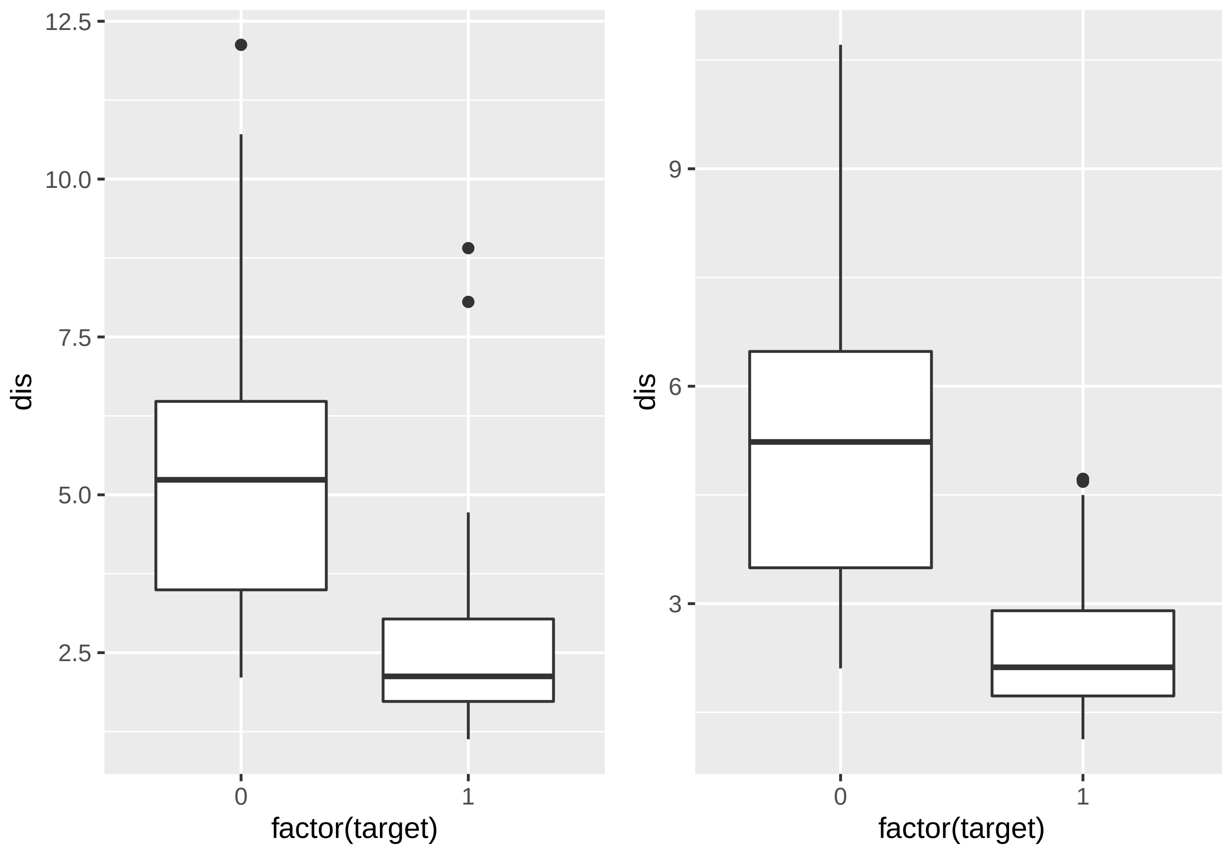



```
detach(train)
```

Dis

Dis also has some outliers so we'll remove rows where dis was greater than 11 and target was 0, and where dis was greater than 7.5 and target was 1.

```
attach(train)
p0 <- ggplot(train, aes(factor(target), dis)) + geom_boxplot()
train <- train[-which(target==0 & dis > 11),]
train <- train[-which(target==1 & dis > 7.5),]
p1 <- ggplot(train, aes(factor(target), dis)) + geom_boxplot()
grid.arrange(p0, p1, ncol=2,nrow=1)
```




```r
detach(train)
```

Data Summary

Let's take a quick look at what variables we have remaining.

```r
names(train)
```

```
##  [1] "zn"      "indus"   "nox"     "rm"      "age"     "dis"     "rad"
##  [8] "tax"     "ptratio" "lstat"   "medv"    "target"  "dataset"
```

```r
dim(train)
```

```
## [1] 452  13
```

Build Models

Model 1 - All Variables

First we will be creating a model with all the variables in the original dataset to create a baseline for other models. Based on the p-values results from this model we will be able to eliminate variables with large p-values

```r
m1 = glm(target ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, data=train, fa
summary(m1)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + rm + age + dis + rad +
##     tax + ptratio + lstat + medv, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3349  -0.0799   0.0000   0.0012   3.1976
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -38.029594   7.489999  -5.077 3.83e-07 ***
## zn           -0.048542   0.035405  -1.371 0.170361
## indus         0.199630   0.089639   2.227 0.025944 *
## nox          45.790310   8.194404   5.588 2.30e-08 ***
## rm           -0.511858   0.813749  -0.629 0.529341
## age           0.035180   0.014517   2.423 0.015378 *
## dis           0.246166   0.253439   0.971 0.331397
## rad           0.926188   0.200821   4.612 3.99e-06 ***
## tax          -0.022971   0.006322  -3.634 0.000279 ***
## ptratio       0.501234   0.162414   3.086 0.002028 **
## lstat         0.098382   0.057781   1.703 0.088630 .
## medv          0.162372   0.077550   2.094 0.036281 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 626.60  on 451  degrees of freedom
## Residual deviance: 157.53  on 440  degrees of freedom
## AIC: 181.53
##
## Number of Fisher Scoring iterations: 9
```

We can see that variables *indus*, *nox*, *age*, *rad*, *tax*, *ptratio*, *lstat*, and *medv* have p values that are close to and or smaller than 0.05 which will be used in the next model

Model 2 - Hand Pick Model

```r
m2 = glm(target ~ indus + nox + age + rad + tax + ptratio + lstat + medv, data=train, family=binomial)
summary(m2)
```

```
##
## Call:
## glm(formula = target ~ indus + nox + age + rad + tax + ptratio +
##     lstat + medv, family = binomial, data = train)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.35458  -0.09486   0.00002   0.00127   2.85518
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.745051   6.273620  -6.016 1.78e-09 ***
## indus         0.228533   0.085607   2.670 0.007595 **
## nox          43.209019   7.453316   5.797 6.74e-09 ***
## age           0.027807   0.010963   2.537 0.011196 *
## rad           0.914150   0.187390   4.878 1.07e-06 ***
## tax          -0.024103   0.005802  -4.154 3.26e-05 ***
## ptratio       0.518734   0.142931   3.629 0.000284 ***
## lstat         0.116872   0.050616   2.309 0.020944 *
## medv          0.110928   0.042294   2.623 0.008721 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 626.60  on 451  degrees of freedom
## Residual deviance: 160.86  on 443  degrees of freedom
## AIC: 178.86
##
## Number of Fisher Scoring iterations: 9
```

Model 3 - Backward Step Model

We will now build a model using backwards selection in order to compare if using backwards selection is better than hand picking values to create a model In order to create the the backward step model we will be using the *MASS* package which includes the *stepAIC* function. The backward step requires us to pass a model which contains all of the predictors. Then the function will fit all the models which contains all but one of the predictors and will then pick the best model using AIC

```r
m3 = stepAIC(m1, direction='backward', trace=FALSE)
summary(m3)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + rad + tax + ptratio +
##     lstat + medv, family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.41614  -0.08644   0.00002   0.00128   3.08111
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -35.751993   6.402055  -5.584 2.34e-08 ***
## zn           -0.045663   0.032925  -1.387   0.1655
## indus         0.207458   0.086225   2.406   0.0161 *
## nox          42.641987   7.490375   5.693 1.25e-08 ***
## age           0.027458   0.011021   2.491   0.0127 *
```

```
## rad           0.939777   0.193632   4.853 1.21e-06 ***
## tax          -0.024982   0.005905  -4.230 2.33e-05 ***
## ptratio       0.457404   0.147306   3.105   0.0019 **
## lstat         0.116084   0.050509   2.298   0.0215 *
## medv          0.107888   0.042022   2.567   0.0102 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 626.60  on 451  degrees of freedom
## Residual deviance: 158.64  on 442  degrees of freedom
## AIC: 178.64
##
## Number of Fisher Scoring iterations: 9
```

Model 4 - Forward Step Model

We can use the same stepAIC function to build the fourth model. The forward selection approach starts from the null model and adds a variable that improves the model the most, one at a time, until the stopping criterion is met. We can see the result is different compared to the backward selection approach. We can see that the result is same as the saturated model m1.

```r
m4 = stepAIC(m1, direction='forward', trace=FALSE)
summary(m4)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + rm + age + dis + rad +
##     tax + ptratio + lstat + medv, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3349  -0.0799   0.0000   0.0012   3.1976
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -38.029594   7.489999  -5.077 3.83e-07 ***
## zn           -0.048542   0.035405  -1.371 0.170361
## indus         0.199630   0.089639   2.227 0.025944 *
## nox          45.790310   8.194404   5.588 2.30e-08 ***
## rm           -0.511858   0.813749  -0.629 0.529341
## age           0.035180   0.014517   2.423 0.015378 *
## dis           0.246166   0.253439   0.971 0.331397
## rad           0.926188   0.200821   4.612 3.99e-06 ***
## tax          -0.022971   0.006322  -3.634 0.000279 ***
## ptratio       0.501234   0.162414   3.086 0.002028 **
## lstat         0.098382   0.057781   1.703 0.088630 .
## medv          0.162372   0.077550   2.094 0.036281 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 626.60  on 451  degrees of freedom
## Residual deviance: 157.53  on 440  degrees of freedom
## AIC: 181.53
##
## Number of Fisher Scoring iterations: 9
```

Model 5 - Stepwise step Model

We also can use the same stepAIC function to build the fifth model using stepwise regression. The stepwise regression method involves adding or removing potential explanatory variables in succession and testing for statistical significance after each iteration. At the very last step stepAIC as shown in the summary table has produced the optimal set of features $\{zn, nox, age, dis, rad, ptratio, medv\}$. This is exactly same result as the backward step model.

```r
m5 = stepAIC(m1, direction='both', trace=FALSE)
summary(m5)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + rad + tax + ptratio +
##     lstat + medv, family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.41614  -0.08644   0.00002   0.00128   3.08111
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -35.751993   6.402055  -5.584 2.34e-08 ***
## zn           -0.045663   0.032925  -1.387   0.1655
## indus         0.207458   0.086225   2.406   0.0161 *
## nox          42.641987   7.490375   5.693 1.25e-08 ***
## age           0.027458   0.011021   2.491   0.0127 *
## rad           0.939777   0.193632   4.853 1.21e-06 ***
## tax          -0.024982   0.005905  -4.230 2.33e-05 ***
## ptratio       0.457404   0.147306   3.105   0.0019 **
## lstat         0.116084   0.050509   2.298   0.0215 *
## medv          0.107888   0.042022   2.567   0.0102 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 626.60  on 451  degrees of freedom
## Residual deviance: 158.64  on 442  degrees of freedom
## AIC: 178.64
##
## Number of Fisher Scoring iterations: 9
```

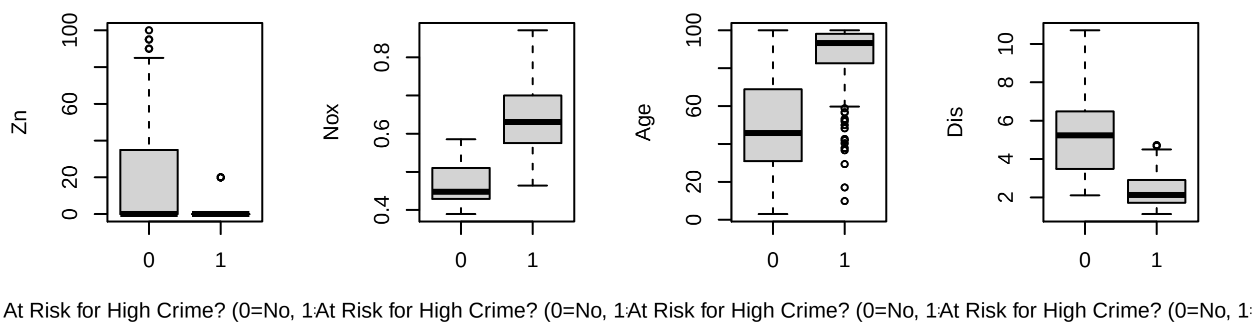
The analysis of deviance table shows further confirms that dropping these 4 variables $\{indus, chas, rm, lstat\}$ either in model 3 or 5 are statistically insignificant and can be dropped.

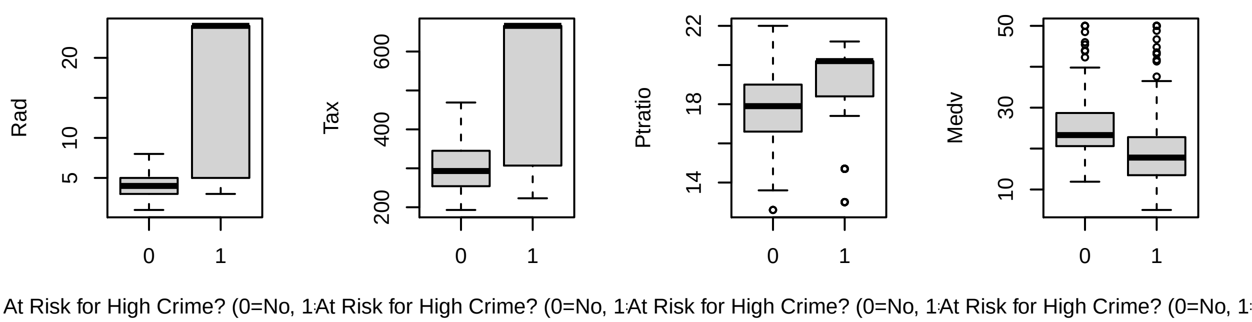
```r
anova(m5,m1, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ zn + indus + nox + age + rad + tax + ptratio + lstat +
##     medv
## Model 2: target ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio +
##     lstat + medv
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       442     158.64
## 2       440     157.53  2   1.1042   0.5757
```

Model 6 - Transformed Predictors Model

We do a box plot for each predictors used in model m5 to check skewness. Out of the 8 predictors, these 5 predictors {*zn*, *nox*, *rad*, *tax*, *Ptratio*} are quite skewed. Thus, we shall include log of these predictors in our logistic regression model m5 or m3.





Now, we create a new model to include those log transformed predictors. We can see from the summary table the impact of including transformed predictors give lower deviance and lower AIC.

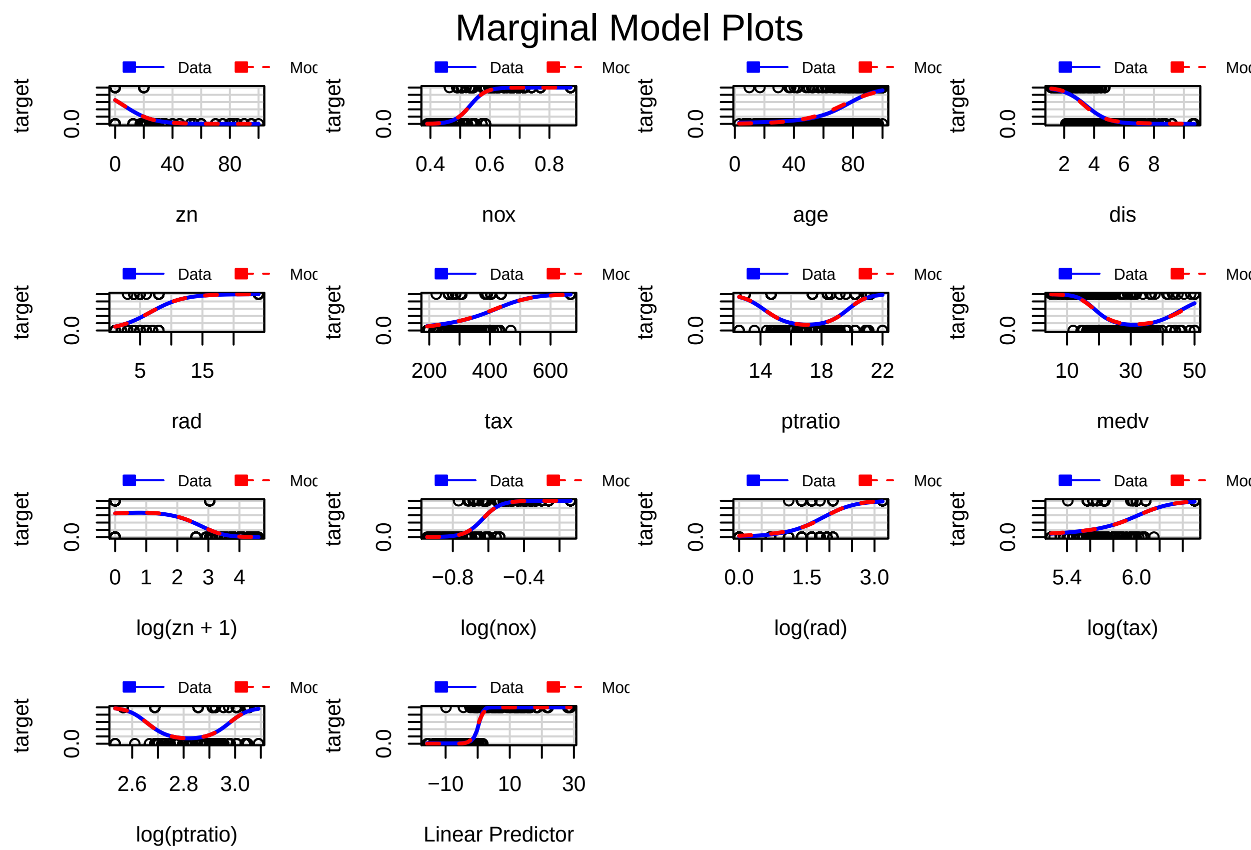
```r
m6 = glm(target~zn+nox+age+dis+rad+tax+ptratio+medv+log(zn+1)+log(nox)+log(rad)+log(tax)+log(ptratio),f
summary(m6)
```

```
##
```

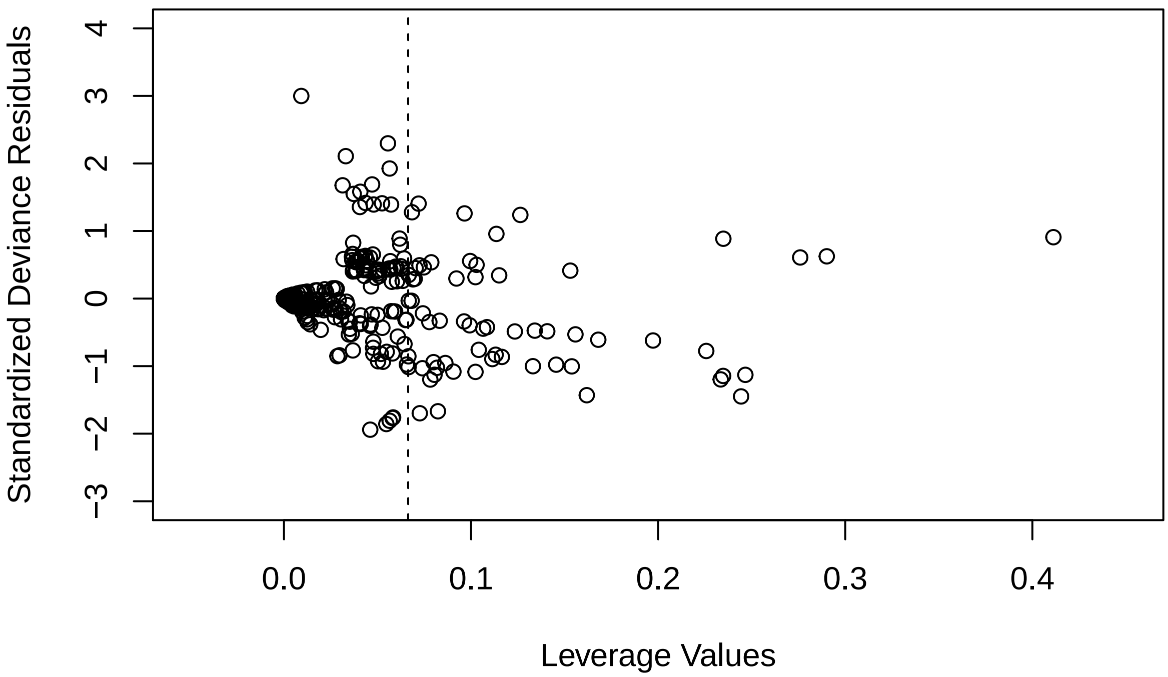
```
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv + log(zn + 1) + log(nox) + log(rad) + log(tax) + log(ptratio),
##     family = binomial(), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8952  -0.0908   0.0000   0.0384   4.4420
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -209.10978  135.24624  -1.546 0.122070
## zn             -0.03922    0.23346  -0.168 0.866596
## nox           254.65099  120.20726   2.118 0.034138 *
## age             0.03277    0.01316   2.490 0.012765 *
## dis             0.30124    0.34788   0.866 0.386525
## rad             1.17299    0.29826   3.933  8.4e-05 ***
## tax            -0.14262    0.03858  -3.697 0.000218 ***
## ptratio         6.20826    2.28035   2.722 0.006479 **
## medv            0.04257    0.05208   0.817 0.413667
## log(zn + 1)    -1.08368    1.65588  -0.654 0.512825
## log(nox)     -112.92548   63.64789  -1.774 0.076026 .
## log(rad)       -0.32827    1.10659  -0.297 0.766731
## log(tax)       40.22883   12.16325   3.307 0.000942 ***
## log(ptratio) -105.88067   41.59776  -2.545 0.010917 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 626.60  on 451  degrees of freedom
## Residual deviance: 141.42  on 438  degrees of freedom
## AIC: 169.42
##
## Number of Fisher Scoring iterations: 11
```

Next we can check if the logistic regression model m6 is adequate or not by doing marginal model plots. From the figure below, it shows both the loess estimate curve and the fitted values curve are in agreement, and that indicates the model m6 is a valid model.

```r
mmps(m6, layout=c(4,4))
```



We can further check the validity of model m6 by plotting leverage values versus standardized deviance. The average leverage is equal to $(p + 1)/n = (14 + 1)/466 = 0.032$. The p value here is the number of predictors from m6 including the intercept. So the usual cut-off is, 0.064, equal to twice the average leverage value. There are number of high leverage points can be seen in the figure below and can be removed at the data preparation step.



Select Models

we will compare various metrics for all six models. We check models' confusion matrix, accuracy, classification error rate, precision, sensitivity, specificity, F1 score, and AUC.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Accuracy	0.9181416	0.9269912	0.9247788	0.9181416	0.9247788	0.9424779
Class. Error Rate	0.0818584	0.0730088	0.0752212	0.0818584	0.0752212	0.0575221
Sensitivity	0.9162996	0.9162996	0.9207048	0.9162996	0.9207048	0.9251101
Specificity	0.9200000	0.9377778	0.9288889	0.9200000	0.9288889	0.9600000
Precision	0.9203540	0.9369369	0.9288889	0.9203540	0.9288889	0.9589041
F1	0.9183223	0.9265033	0.9247788	0.9183223	0.9247788	0.9417040
AUC	0.9811650	0.9802252	0.9814978	0.9811650	0.9814978	0.9863142

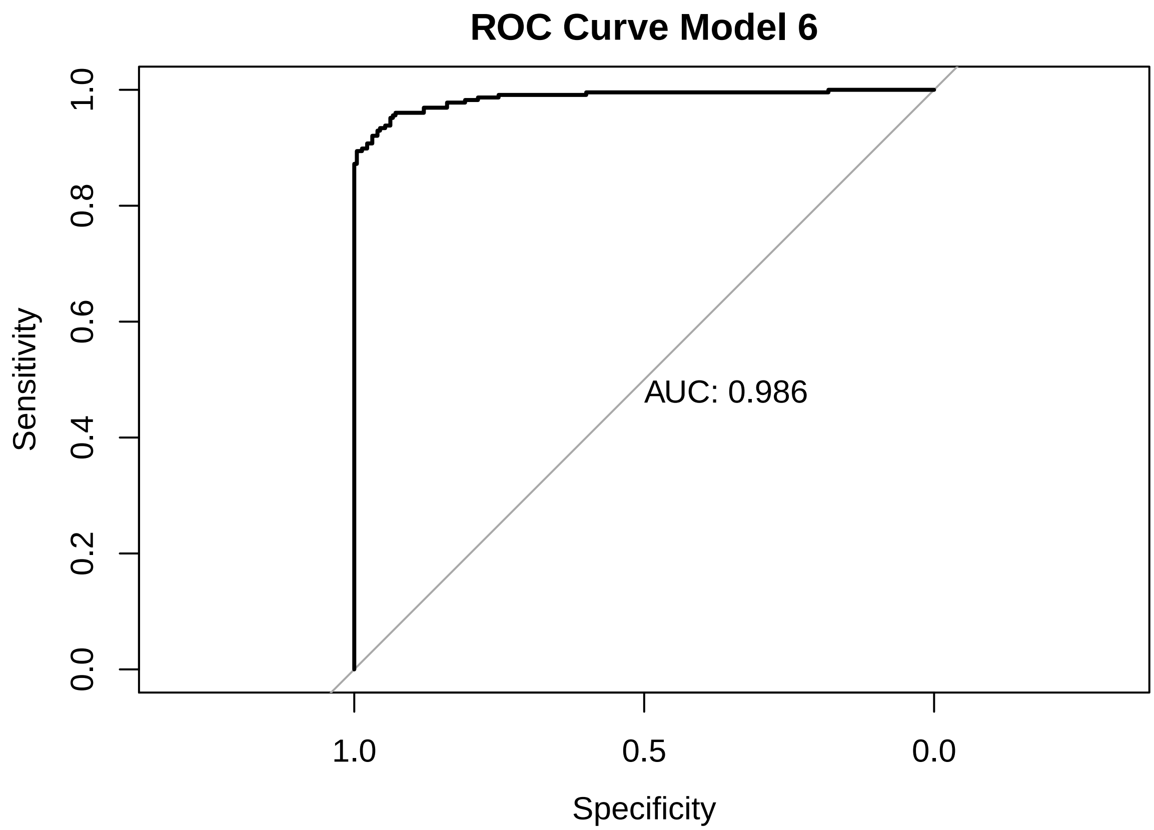
Model 6 performs the highest in all metrics except Class. Error Rate.

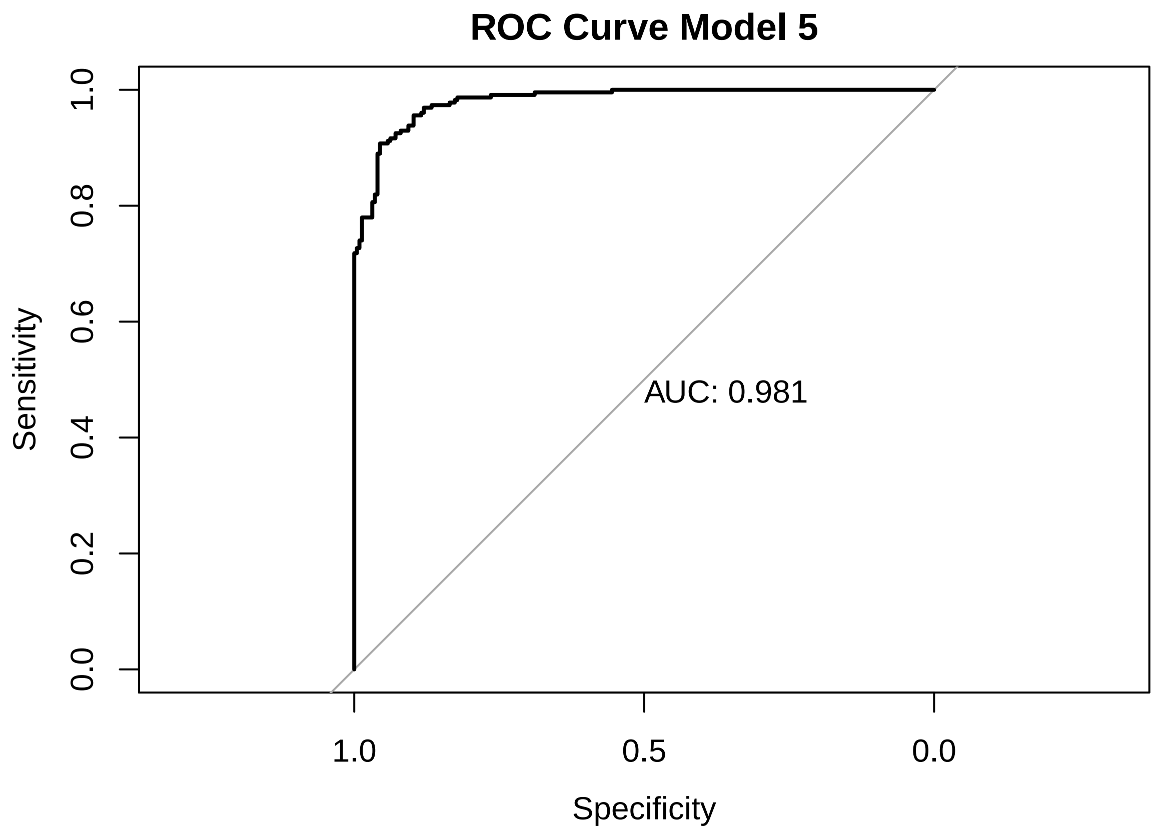
Model 1 and 4 perform the same.

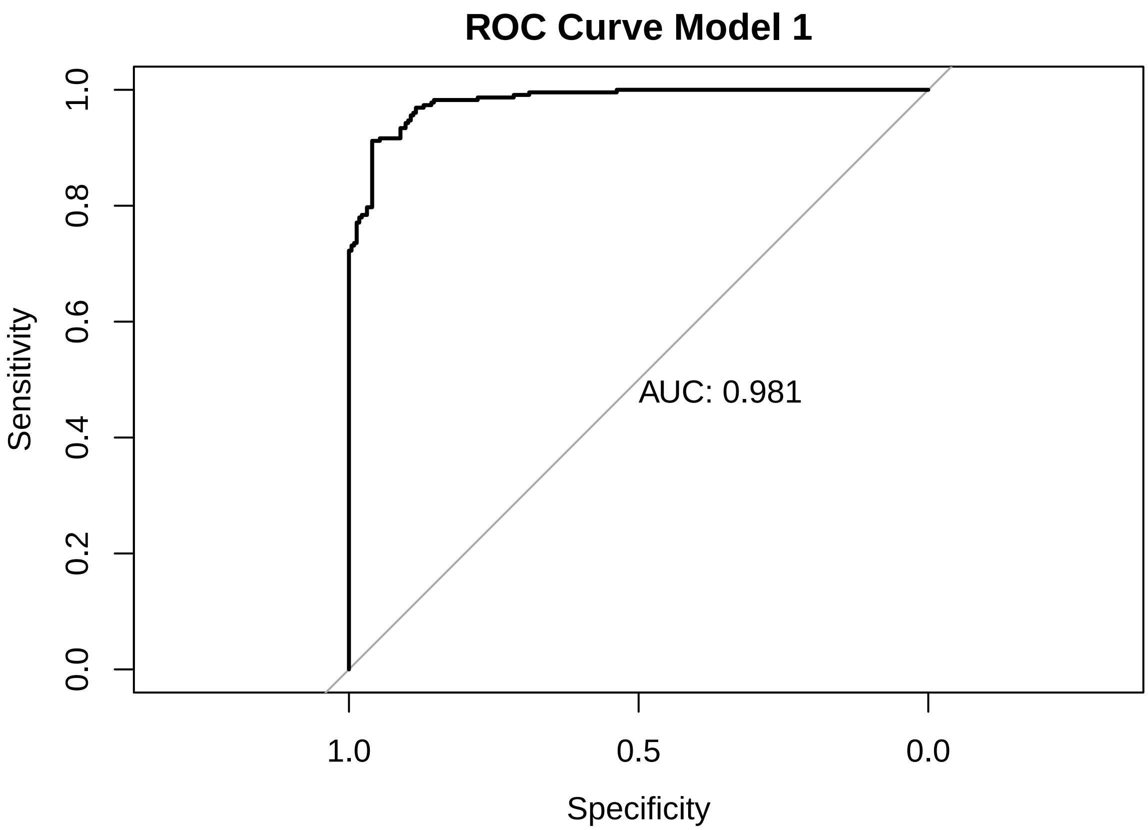
Model 3 and 5 perform the same.

Model 5 is pretty much close to all other metrics.

Let's look at the roc curve to help us make the best selection.


ROC Curve Model 6
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
AUC: 0.986
1.0
0.5
0.0
Specificity


ROC Curve Model 5
AUC: 0.981
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity
1.0
0.5
0.0



As we can see, the model 6 is the best model. Let's now using the evaluation dataset to evaluate the model.

```
##    zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv TARGET
## 1   0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7      0
## 2   0  8.14    0 0.538 6.096 84.5 4.4619   4 307    21.0 10.26 18.2      1
## 3   0  8.14    0 0.538 6.495 94.4 4.4547   4 307    21.0 12.80 18.4      1
## 4   0  8.14    0 0.538 5.950 82.0 3.9900   4 307    21.0 27.71 13.2      1
## 5   0  5.96    0 0.499 5.850 41.5 3.9342   5 279    19.2  8.77 21.0      0
## 6  25  5.13    0 0.453 5.741 66.2 7.2254   8 284    19.7 13.15 18.7      0
## 7  25  5.13    0 0.453 5.966 93.4 6.8185   8 284    19.7 14.44 16.0      0
## 8   0  4.49    0 0.449 6.630 56.1 4.4377   3 247    18.5  6.53 26.6      0
## 9   0  4.49    0 0.449 6.121 56.8 3.7476   3 247    18.5  8.44 22.2      0
## 10  0  2.89    0 0.445 6.163 69.6 3.4952   2 276    18.0 11.34 21.4      0
```

```
write.csv(evaluation$TARGET,paste0(getwd(),"/Evaluation_Target.csv"),row.names = FALSE)
```

Appendix

```
train <- read.csv("https://raw.githubusercontent.com/ahussan/DATA_621_Group1/main/HW3/crime-training-da
```

```
print(head(train, 10))
```

```r
print(colMeans(train))
```

```r
print(apply(train, 2, sd))
```

```r
print(apply(train, 2, median))
```

```r
boxplot(train, use.cols = TRUE)
```

```r
train.cor = cor(train)
print(train.cor)
```

```r
corrgram(train, order=TRUE, lower.panel=panel.shade,
  upper.panel=panel.pie, text.panel=panel.txt,
  main="visualize the data in correlation matrices ")
```

```r
knitr::kable(table(train$target))
```

```r
plot_histogram(train)
relationships <- train
relationships$chas <- NULL
pairs(train %>% select_if(is.numeric))
```

```r
#convert features to factor and add a dataset feature
train$chas <- as.factor(train$chas)
train$target <- as.factor(train$target)
train$dataset <- 'train'
plotfontsize <- 8
train_int_names <- train %>% select_if(is.numeric)
int_names <- names(train_int_names)
for (i in int_names) {
  assign(paste0("var_",i), ggplot(train, aes_string(x = train$target, y = i)) +
          geom_boxplot(color = 'steelblue',
                       outlier.color = 'firebrick',
                       outlier.alpha = 0.35) +
#scale_y_continuous
          labs(title = paste0(i,' vs target'), y = i, x= 'target') +
          theme_minimal() +
          theme(
            plot.title = element_text(hjust = 0.45),
            panel.grid.major.y =  element_line(color = "grey", linetype = "dashed"),
            panel.grid.major.x = element_blank(),
            panel.grid.minor.y = element_blank(),
            panel.grid.minor.x = element_blank(),
            axis.ticks.x = element_line(color = "grey"),
            text = element_text(size=plotfontsize)
          ))
}
gridExtra::grid.arrange(var_age, var_dis, var_indus,var_lstat,
                        var_medv,var_nox,var_ptratio,var_rad,
                        var_rm, var_tax, var_zn, nrow=4)
numeric_values <- train %>% select_if(is.numeric)
```

```
train_cor <- cor(numeric_values)
corrplot.mixed(train_cor, tl.col = 'black', tl.pos = 'lt')
```

```
test_url <- 'https://raw.githubusercontent.com/ahussan/DATA_621_Group1/main/HW3/crime-evaluation-data_m
test <- read.csv(test_url, header=TRUE)
trainchas <- as.factor(train$chas)
train$chas <- NULL
traintarget <- as.factor(train$target)
train$target <- traintarget
testchas <- as.factor(test$chas)
test$chas <- NULL
```

```
attach(train)
p0 <- ggplot(train, aes(factor(target), indus)) + geom_boxplot()
train <- train[-which(target==0 & indus > 20),]
p1 <- ggplot(train, aes(factor(target), indus)) + geom_boxplot()
grid.arrange(p0, p1,ncol=2,nrow=1)
detach(train)
```

```
attach(train)
p0 <- ggplot(train, aes(factor(target), dis)) + geom_boxplot()
train <- train[-which(target==0 & dis > 11),]
train <- train[-which(target==1 & dis > 7.5),]
p1 <- ggplot(train, aes(factor(target), dis)) + geom_boxplot()
grid.arrange(p0, p1, ncol=2,nrow=1)
detach(train)
```

```
names(train)
dim(train)
```

```
m1 = glm(target ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, data=train, fa
summary(m1)
```

```
m2 = glm(target ~ indus + nox + age + rad + tax + ptratio + lstat + medv, data=train, family=binomial)
summary(m2)
```

```
m3 = stepAIC(m1, direction='backward', trace=FALSE)
summary(m3)
```

```
m4 = stepAIC(m1, direction='forward', trace=FALSE)
summary(m4)
```

```
m5 = stepAIC(m1, direction='both', trace=FALSE)
summary(m5)
```

```
anova(m5,m1, test="Chi")
```

```
par(mfrow=c(2,4))
boxplot(zn~target, ylab="Zn",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
boxplot(nox~target, ylab="Nox",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
```

```r
boxplot(age~target, ylab="Age",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
boxplot(dis~target, ylab="Dis",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
boxplot(rad~target, ylab="Rad",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
boxplot(tax~target, ylab="Tax",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
boxplot(ptratio~target, ylab="Ptratio",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
boxplot(medv~target, ylab="Medv",xlab="At Risk for High Crime? (0=No, 1=Yes)", data=train)
```

```r
m6 = glm(target~zn+nox+age+dis+rad+tax+ptratio+medv+log(zn+1)+log(nox)+log(rad)+log(tax)+log(ptratio),fa
summary(m6)
```

```r
mmps(m6, layout=c(4,4))
```

```r
hvalues <- influence(m6)$hat
stanresDeviance <- residuals(m6)/sqrt(1-hvalues)
plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",xlab="Leverage Values",ylim=c(-3,4)
abline(v=2*15/length(train$target),lty=2)
```

```r
# comparing all models using various measures
CM1 <- confusionMatrix(as.factor(as.integer(fitted(m1) > .5)), as.factor(m1$y), positive = "1")
CM2 <- confusionMatrix(as.factor(as.integer(fitted(m2) > .5)), as.factor(m2$y), positive = "1")
CM3 <- confusionMatrix(as.factor(as.integer(fitted(m3) > .5)), as.factor(m3$y), positive = "1")
CM4 <- confusionMatrix(as.factor(as.integer(fitted(m4) > .5)), as.factor(m4$y), positive = "1")
CM5 <- confusionMatrix(as.factor(as.integer(fitted(m5) > .5)), as.factor(m5$y), positive = "1")
CM6 <- confusionMatrix(as.factor(as.integer(fitted(m6) > .5)), as.factor(m6$y), positive = "1")
```

```r
Roc1 <- roc(train$target,  predict(m1, train, interval = "prediction"))
Roc2 <- roc(train$target,  predict(m2, train, interval = "prediction"))
Roc3 <- roc(train$target,  predict(m3, train, interval = "prediction"))
Roc4 <- roc(train$target,  predict(m4, train, interval = "prediction"))
Roc5 <- roc(train$target,  predict(m5, train, interval = "prediction"))
Roc6 <- roc(train$target,  predict(m6, train, interval = "prediction"))
```

```r
metrics1 <- c(CM1$overall[1], "Class. Error Rate" = 1 - as.numeric(CM1$overall[1]), CM1$byClass[c(1, 2,
metrics2 <- c(CM2$overall[1], "Class. Error Rate" = 1 - as.numeric(CM2$overall[1]), CM2$byClass[c(1, 2,
metrics3 <- c(CM3$overall[1], "Class. Error Rate" = 1 - as.numeric(CM3$overall[1]), CM3$byClass[c(1, 2,
metrics4 <- c(CM4$overall[1], "Class. Error Rate" = 1 - as.numeric(CM4$overall[1]), CM4$byClass[c(1, 2,
metrics5 <- c(CM5$overall[1], "Class. Error Rate" = 1 - as.numeric(CM5$overall[1]), CM5$byClass[c(1, 2,
metrics6 <- c(CM6$overall[1], "Class. Error Rate" = 1 - as.numeric(CM6$overall[1]), CM6$byClass[c(1, 2,
```

```r
kable(cbind(metrics1, metrics2, metrics3, metrics4, metrics5, metrics6), col.names = c("Model 1", "Model
  kable_styling(full_width = T)
```

```r
# plotting roc curve of model 6
plot(roc(train$target,  predict(m6, train, interval = "prediction")), print.auc = TRUE, main='ROC Curve
```

```r
# plotting roc curve of model 5
plot(roc(train$target,  predict(m5, train, interval = "prediction")), print.auc = TRUE, main='ROC Curve
```

```r
# plotting roc curve of model 4
plot(roc(train$target,  predict(m4, train, interval = "prediction")), print.auc = TRUE, main='ROC Curve
```

```r
evaluation <- read.csv("https://raw.githubusercontent.com/ahussan/DATA_621_Group1/main/HW3/crime-evalua
evaluation$TARGET <- predict(m6, evaluation, type="response")
evaluation$TARGET <- ifelse(evaluation$TARGET > 0.5, 1, 0)
print(head(evaluation,10))
```

```r
write.csv(evaluation$TARGET,paste0(getwd(),"/Evaluation_Target.csv"),row.names = FALSE)
```