621_HW3.Rmd

Kumudini Bhave

April 10, 2017

Contents

1 Logistic Regression Model : Predicting Crime Rate For City Neighbourhoods

---

1.1 Summary

This is an R Markdown document for providing documentation for performing **Binary Logistic Regression** by practising **Data Exploration, Transformation, Analysis And Modelling and Prediction Of the Crime DataSet**

1.2 Crime DataSet

The Crime dataset of a major city depicts 466 observations across 14 variables for different neighbourhoods of the city. The response /dependant variable is the "target" which is essentially the crime rate , whether it is above the median crime rate or not.(1 if yes and 0 if not)

Predictor Variables	Definition
zn	proportion of residential land zoned for large lots (over 25000 square feet)
indus	proportion of non-retail business acres per suburb
chas	a dummy var. for whether the suburb borders the Charles River (1) or not (0)
nox	nitrogen oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted mean of distances to five Boston employment centers
rad	index of accessibility to radial highways
tax	full-value property-tax rate per $10,000
ptratio	pupil-teacher ratio by town
black	1000(Bk - 0.63)2 where Bk is the proportion of blacks by town
lstat	lower status of the population (percent)
medv	median value of owner-occupied homes in $1000s

```r
knitr::opts_chunk$set(message = FALSE, echo = TRUE)

# Library for loading CSV data
library(RCurl)

# Library for data display in tabular format

# library(DT)
library(dplyr)
# Library for plotting
library(ggplot2)
library(gridExtra)

library(corrplot)
library(e1071)
library(data.table)
library(knitr)
library(caret)
```

```r
library(pander)
library(pROC)
library(car)
library(bestglm)
```

```r
# Getting data

trdata.giturl <- "https://raw.githubusercontent.com/DataDriven-MSDA/DATA621/master/HW3/crime-training-da

evaldata.giturl <- "https://raw.githubusercontent.com/DataDriven-MSDA/DATA621/master/HW3/crime-evaluati



traindataorig <- read.csv(url(trdata.giturl))
traindata <- traindataorig

evaldataorig <- read.csv(url(evaldata.giturl))
evaldata <- evaldataorig


# View(traindata)
```

1.3 Data Exploration

Below is the summary of the predictor variables and the response variable "target" in the dataset.

Response Variable:

We find that the "target" response variable has 229 neighbourhoods with above median crime rate (i.e. value of 1) and 237 neighbourhoods with above median crime rate (i.e. value of 0)

```r
pander(table(traindata$target))
```

0	1
237	229

```r
pander(table(traindata$target)/sum(table(traindata$target)))
```

0	1
0.5086	0.4914

Because it is a binary response there are no outliers. We see that lower crime neighbourhoods and high crime neighbourhoods are pretty much equally distributed

Predictor Variables :

We have a list of Predictor variables which seem to have an impact on the response variable of "target". Some of them positively or negatively impacting. 12 are numeric and 1 is caterogical.

Since our response variable target is a two-level factor, we can take a look at a plot of each predictor, subset by target and see the relationship between the predictor and our response variable

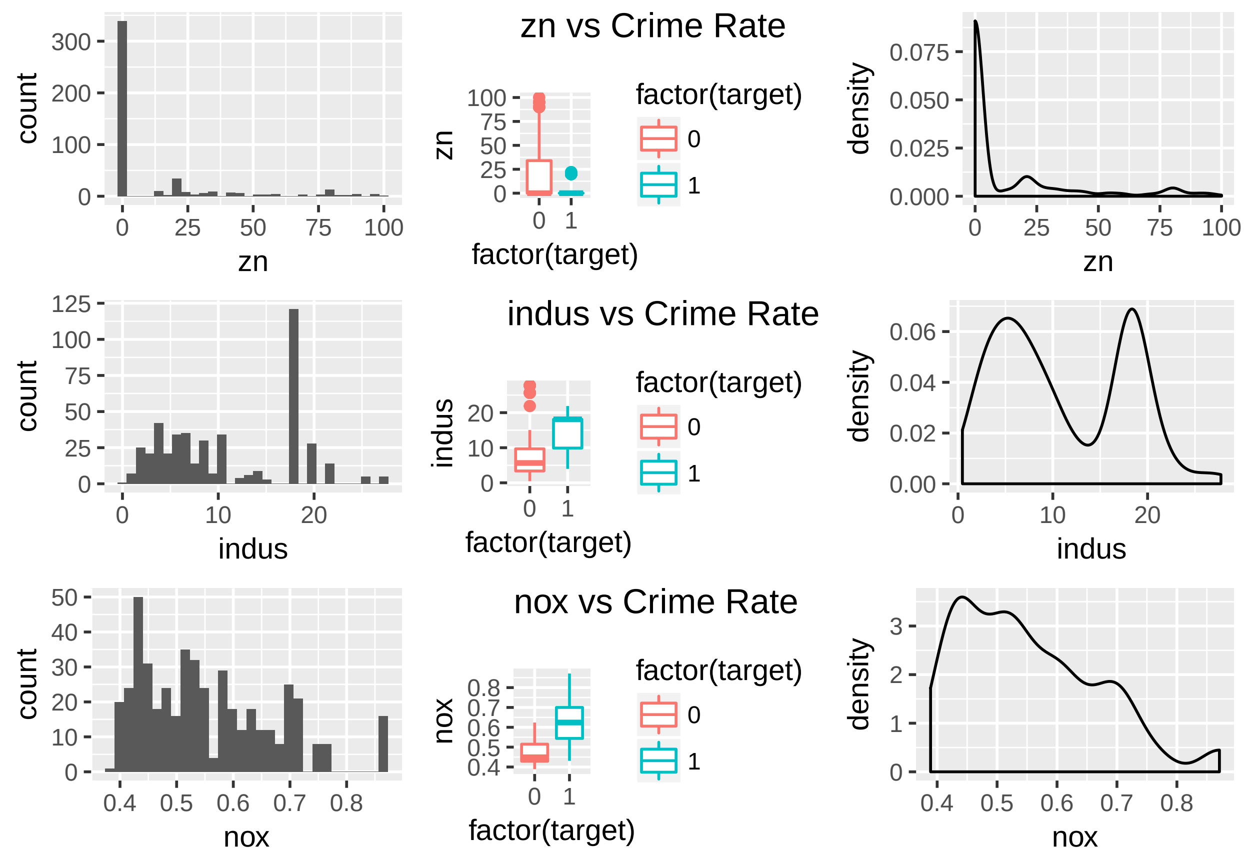
```r
summary(traindata)
```

```
##        zn             indus             chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio         black            lstat
##  Min.   :187.0   Min.   :12.6   Min.   :  0.32   Min.   : 1.730
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.:375.61   1st Qu.: 7.043
##  Median :334.5   Median :18.9   Median :391.34   Median :11.350
##  Mean   :409.5   Mean   :18.4   Mean   :357.12   Mean   :12.631
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:396.24   3rd Qu.:16.930
##  Max.   :711.0   Max.   :22.0   Max.   :396.90   Max.   :37.970
##       medv          target
```

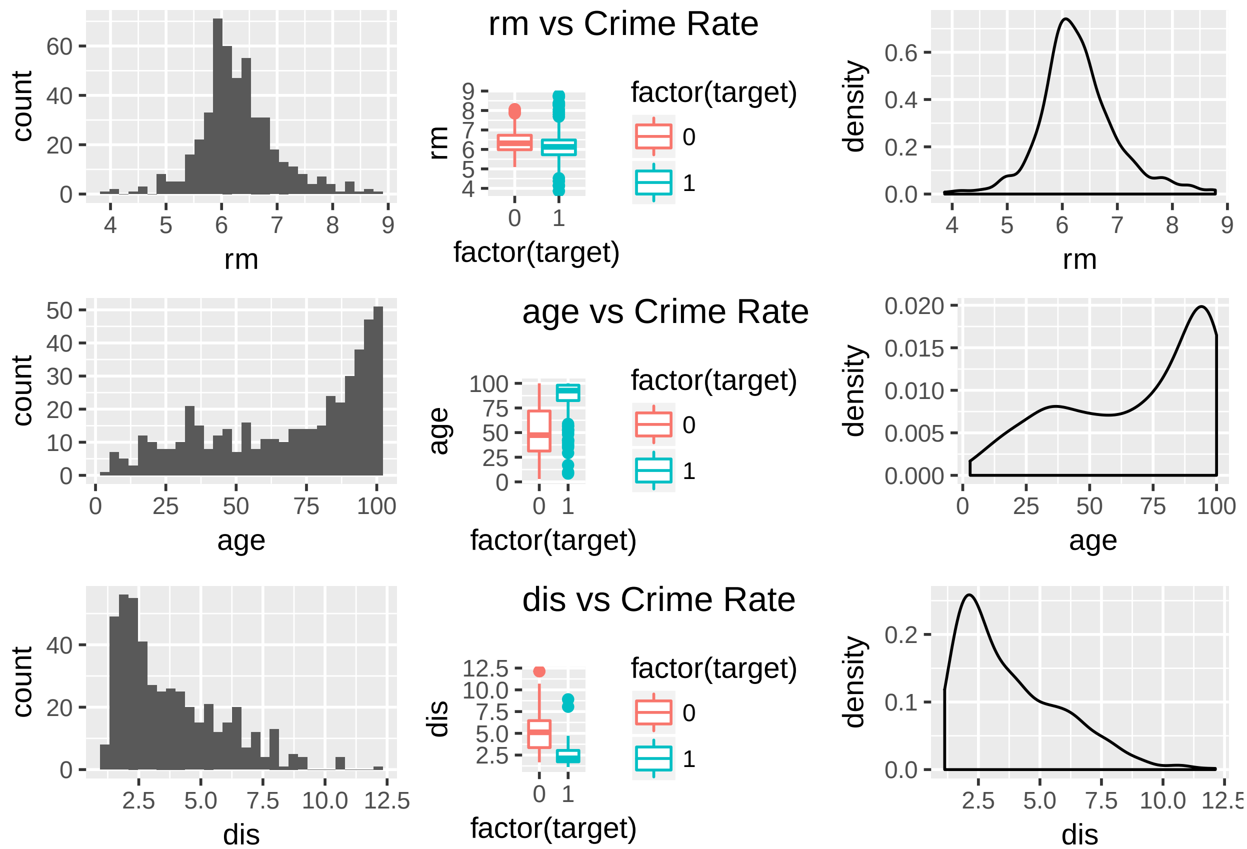
```
##  Min.   : 5.00   Min.   :0.0000
##  1st Qu.:17.02   1st Qu.:0.0000
##  Median :21.20   Median :0.0000
##  Mean   :22.59   Mean   :0.4914
##  3rd Qu.:25.00   3rd Qu.:1.0000
##  Max.   :50.00   Max.   :1.0000
```

```
znhist <- ggplot(traindata, aes(x = zn)) + geom_histogram()
indushist <- ggplot(traindata, aes(x = indus)) + geom_histogram()
noxhist <- ggplot(traindata, aes(x = nox)) + geom_histogram()
rmhist <- ggplot(traindata, aes(x = rm)) + geom_histogram()
agehist <- ggplot(traindata, aes(x = age)) + geom_histogram()
dishist <- ggplot(traindata, aes(x = dis)) + geom_histogram()
radhist <- ggplot(traindata, aes(x = rad)) + geom_histogram()
taxhist <- ggplot(traindata, aes(x = tax)) + geom_histogram()
ptratiohist <- ggplot(traindata, aes(x = ptratio)) + geom_histogram()
blackhist <- ggplot(traindata, aes(x = black)) + geom_histogram()
lstathist <- ggplot(traindata, aes(x = lstat)) + geom_histogram()
medvhist <- ggplot(traindata, aes(x = medv)) + geom_histogram()


znbox <- ggplot(traindata, aes(factor(target), zn, colour = factor(target))) + geom_boxplot() +
    ggtitle("zn vs Crime Rate\n")

indusbox <- ggplot(traindata, aes(factor(target), indus, colour = factor(target))) +
    geom_boxplot() + ggtitle("indus vs Crime Rate\n")

noxbox <- ggplot(traindata, aes(factor(target), nox, colour = factor(target))) +
    geom_boxplot() + ggtitle("nox vs Crime Rate\n")

rmbox <- ggplot(traindata, aes(factor(target), rm, colour = factor(target))) + geom_boxplot() +
    ggtitle("rm vs Crime Rate\n")

agebox <- ggplot(traindata, aes(factor(target), age, colour = factor(target))) +
    geom_boxplot() + ggtitle("age vs Crime Rate\n")

disbox <- ggplot(traindata, aes(factor(target), dis, colour = factor(target))) +
    geom_boxplot() + ggtitle("dis vs Crime Rate\n")

radbox <- ggplot(traindata, aes(factor(target), rad, colour = factor(target))) +
    geom_boxplot() + ggtitle("rad vs Crime Rate\n")

taxbox <- ggplot(traindata, aes(factor(target), tax, colour = factor(target))) +
    geom_boxplot() + ggtitle("tax vs Crime Rate\n")

ptratiobox <- ggplot(traindata, aes(factor(target), ptratio, colour = factor(target))) +
    geom_boxplot() + ggtitle("ptratio vs Crime Rate\n")

blackbox <- ggplot(traindata, aes(factor(target), black, colour = factor(target))) +
    geom_boxplot() + ggtitle("black vs Crime Rate\n")

lstatbox <- ggplot(traindata, aes(factor(target), lstat, colour = factor(target))) +
    geom_boxplot() + ggtitle("lstat vs Crime Rate\n")
```

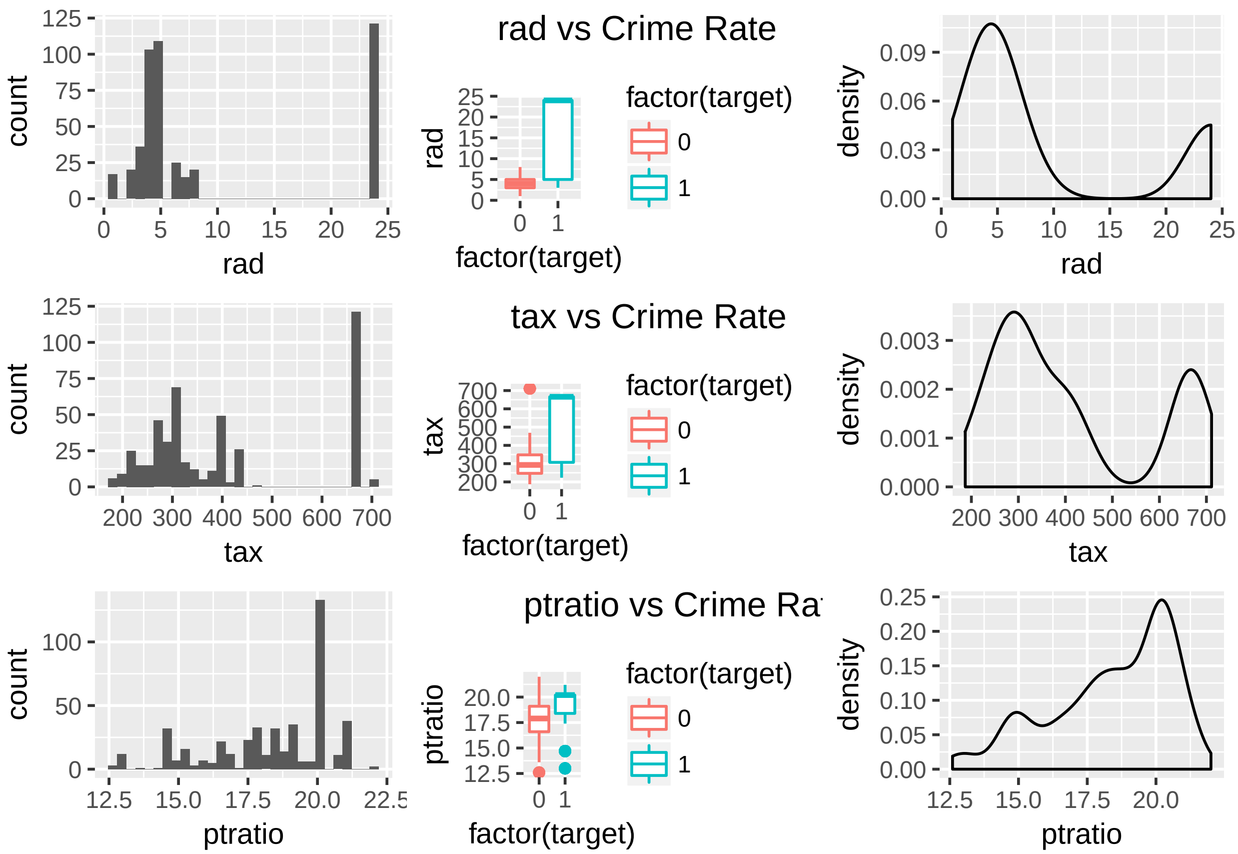
```
medvbox <- ggplot(traindata, aes(factor(target), medv, colour = factor(target))) +
    geom_boxplot() + ggtitle("mdev vs Crime Rate\n")



znden <- ggplot(traindata, aes(x = zn)) + geom_density()
indusden <- ggplot(traindata, aes(x = indus)) + geom_density()
noxden <- ggplot(traindata, aes(x = nox)) + geom_density()
rmden <- ggplot(traindata, aes(x = rm)) + geom_density()
ageden <- ggplot(traindata, aes(x = age)) + geom_density()
disden <- ggplot(traindata, aes(x = dis)) + geom_density()
radden <- ggplot(traindata, aes(x = rad)) + geom_density()
taxden <- ggplot(traindata, aes(x = tax)) + geom_density()
ptratioden <- ggplot(traindata, aes(x = ptratio)) + geom_density()

blackden <- ggplot(traindata, aes(x = black)) + geom_density()
lstatden <- ggplot(traindata, aes(x = lstat)) + geom_density()
medvden <- ggplot(traindata, aes(x = medv)) + geom_density()



grid.arrange(znhist, znbox, znden, indushist, indusbox, indusden, noxhist, noxbox,
    noxden, ncol = 3, nrow = 3)
```



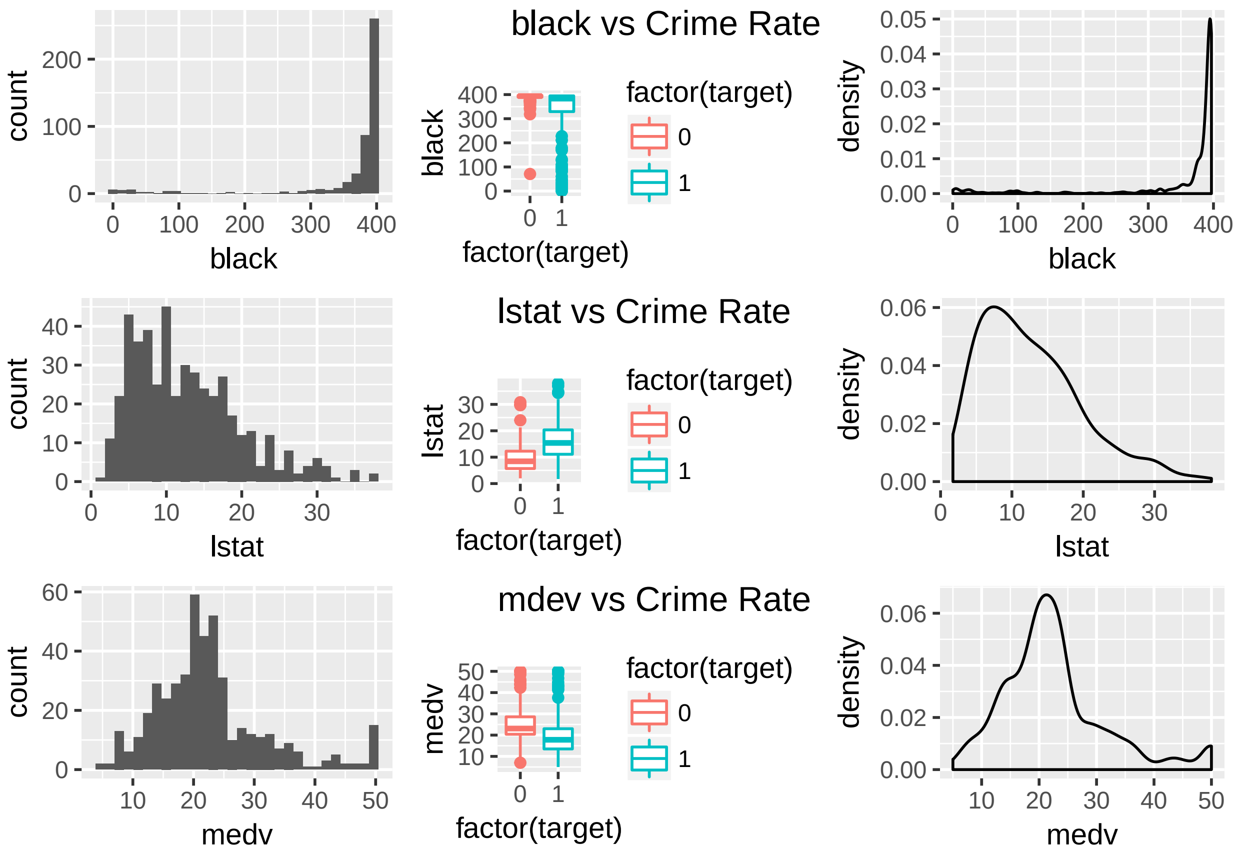

```
grid.arrange(rmhist, rmbox, rmden, agehist, agebox, ageden, dishist, disbox, disden,
    ncol = 3, nrow = 3)
```




```r
grid.arrange(radhist, radbox, radden, taxhist, taxbox, taxden, ptratiohist, ptratiobox,
    ptratioden, ncol = 3, nrow = 3)
```




```r
grid.arrange(blackhist, blackbox, blackden, lstathist, lstatbox, lstatden, medvhist,
    medvbox, medvden, ncol = 3, nrow = 3)
```



There are no missing values and the dataset is overall ok. Apart from skewness in few predictor variable, the data does not seem to be out of norm.

From the above plots , we find that the proportion of buildings built before 1940 , denoted by predictor variable "age",is pretty left skewed shows a high skew in higher crime neighbourhood .

We find similar left skewness in the number of "black" in the neighbourhood for both low and high crime neighbourhoods.

Also the weighted mean distance to Boston employment centre, "dis", is right skewed "zn", proportion for residential land zoned for large lots is also severely right skewed

1.4 Data Preparation

Since the "target" response variable and "chas" predictor variables are binary, we factor them

```
# convert chas (suburb borders the Charles river), it being a binary variable
# into a factor
traindata$chas <- factor(traindata$chas)
evaldata$chas <- factor(evaldata$chas)
```

Since there are no missing values, we don't have to do any imputations.

For "zn" , since more than 70% of the on observations have no residential land zoned for large lots, we opt to categorize "zn" into buckets of neighbourhoods with large lots zoning (values of zn >5) and no/less lots zoning (zn < = 5).

We add a new variable *znnew* which is categorical that has value of 1 for "zn" > 5 and value of 0 for "zn" <=5

Overall , we find that the predictor variables, "zn", "nox" (nitrogen oxide concentrations),"age", "dis" (distance rom employment centr), "bleack" appear to be important in prediction of the crime rate. We also see "tax", "rad" (access to highway). We would need to explore this further and handle the multicollinearity among the predictor variables if any.

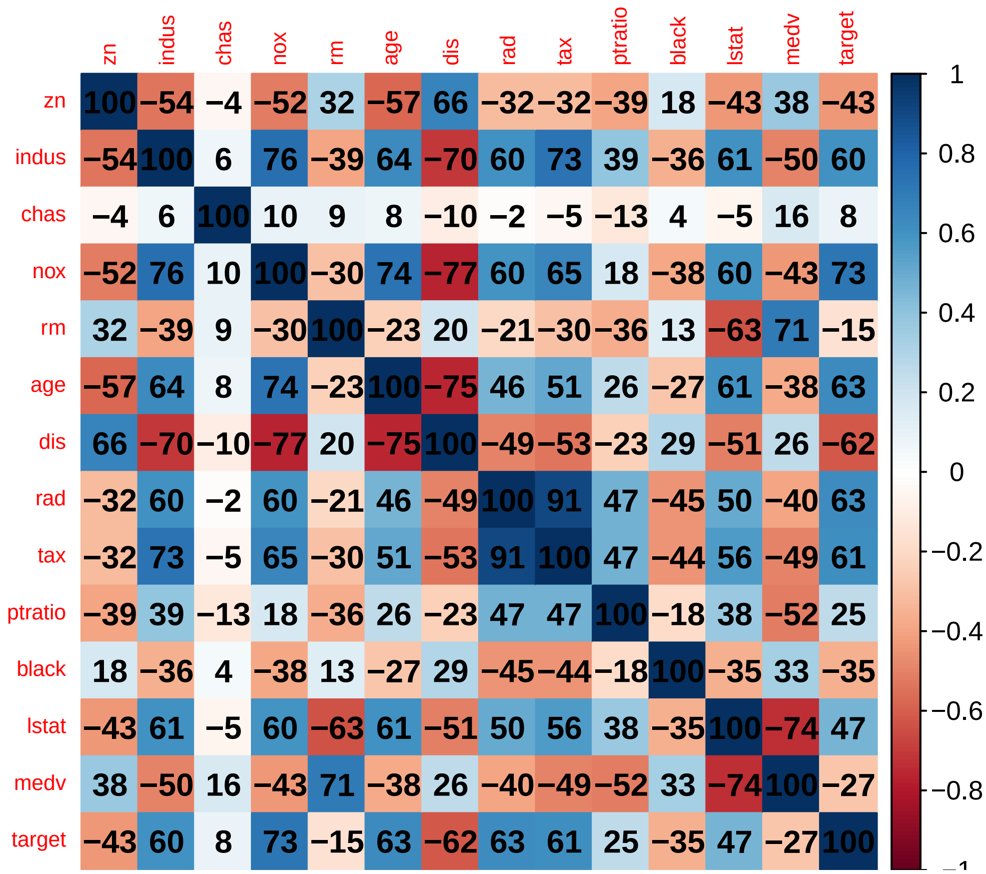
```
# convert chas (suburb borders the Charles river), it being a binary variable
# into a factor

traindata$znnew <- ifelse(traindata$zn > 5, 1, 0)
traindata$znnew <- as.factor(traindata$znnew)

# bucket the 'zn' variable
tzn <- as.data.frame(table(znnew = traindata$znnew, Target = traindata$target))
kable(tzn, align = "c")
```

znnew	Target	Freq
0	0	125
1	0	112
0	1	214
1	1	15

** Correlation Matrix**

Using the original predictor variables to find their correlation with the response variable, we have the following correlation plot.

```
cormat <- as.matrix(cor(traindataorig, use = "pairwise.complete.obs"))
corrplot(cormat, method = "color", tl.cex = 0.7, addCoef.col = "black", addCoefasPercent = TRUE)
```



From the Correlation Matrix :

We do see some correlation , among variables such as "indus" the industrialization negative effect on the median value of homes "medv".

Likewise, the industrialization and nitrogen oxide levels show a strong positive correlation.

We observe that the nitrogen oxide also has a positive correlation with the crime rate and the "rad" which is radial highways accessibility and negatively correlated to "medv" median value of home, depicting that the industrialization and high traffic areas lead to potential high nitrogen oxide emissions which can further lead to lower values of real estate and thus an increase in the crime rate

We find that the "dis" which is distance to employment centres is negatively correlated to crime rate. This is intuitive because employment centres are likely to be in areas of high unemployment which is also correlated to high crime rate. Some of these predictors appear to be correlated like industrailization and access to highway, similarly tax and indusrialization also have strong correlation.

We explore by checking some transformations for the skewed predictor variables like "age", "black", "nox", "indus". Through the trials, We do a logarithm transformation for the age , black and nox and a sqrt transformation for the indus and check if their correlation to the response variable "target" betters with the transformations.

```r
log_age_cor <- cor(traindata$target, log(traindata$age))
log_age_cor
```

```
## [1] 0.5431245
```

```r
log_black_cor <- cor(traindata$target, log(traindata$black))
log_black_cor
```

```
## [1] -0.2723165
```

```r
log_nox_cor <- cor(traindata$target, log(traindata$nox))
log_nox_cor
```

```
## [1] 0.7456989
```

```r
sqrt_indus_cor <- cor(traindata$target, sqrt(traindata$indus))
sqrt_indus_cor
```

```
## [1] 0.6166801
```

After performing trials for different transformations for handling the skewness for certain predictor variable, we find that the log(age) and log(black) do not add much significance to the correlation so we leave them as it is .

However we do see that the log(nox) and the square root of "indus" does make a slight impact and betters the correlation with the target crime rate.

We will further see how these really impact the target in our models. We add the log transformations to "nox" and square root transformation to "indus" predictor variables

```r
traindata$lognox <- log(traindata$nox)

traindata$sqrtindus <- sqrt(traindata$nox)

# View(traindata)
```

1.5 Building The Models

Now that the response and predictor variables have been studies, we further proceed by constructing different models. We will initiate with first all the variables along with the newly added "znnew", and log(nox), sqrt(indus).

Also to crossvalidate the models constructed, we verify it with splitting the train data into 70:30 ratio by randomly selecting the observation data for further analysis of models (since evaluation data lacks the target response variable)

```
set.seed(41)
randomobs <- sample(seq_len(nrow(traindata)), size = floor(0.7 * nrow(traindata)))

trainnew <- traindata[randomobs, ]
testnew <- traindata[-randomobs, ]
```

1.5.1 Model 1 : Full model with transformed predictor variables

As our first model, we construct this using all predictor variables and also include the "znnew" (categorized residential zoned lots) and "lognox" which is log(nox) (nitroden oxide concentrations) and the "sqrtindus" , which is sqrt(indus) (non-retail businees acres / industrial). In logistic regression we expect this model to have the highest predictive capacity.

```
# View(trainnew)
model1 <- glm(target ~ ., family = binomial(link = "logit"), data = trainnew)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model1)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = trainnew)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9778  -0.1671  -0.0064   0.0010   3.5471
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.943e+03  1.137e+04   0.611  0.54149
## zn          -1.051e-02  8.772e-02  -0.120  0.90466
## indus       -1.222e-01  7.157e-02  -1.708  0.08765 .
## chas1        6.354e-02  9.988e-01   0.064  0.94927
## nox          3.829e+03  5.887e+03   0.650  0.51546
## rm          -5.256e-01  9.049e-01  -0.581  0.56137
## age          3.113e-02  1.616e-02   1.926  0.05411 .
## dis          7.056e-01  3.249e-01   2.171  0.02990 *
## rad          7.615e-01  1.980e-01   3.847  0.00012 ***
## tax         -8.765e-03  3.937e-03  -2.226  0.02601 *
## ptratio      2.592e-01  1.694e-01   1.530  0.12602
## black       -8.775e-03  6.048e-03  -1.451  0.14677
## lstat        1.661e-02  7.713e-02   0.215  0.82954
## medv         9.845e-02  8.341e-02   1.180  0.23787
```

```
## znnew1      -1.943e+00  2.163e+00  -0.898  0.36892
## lognox       1.886e+03  3.128e+03   0.603  0.54653
## sqrtindus   -1.069e+04  1.718e+04  -0.622  0.53378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.15  on 325  degrees of freedom
## Residual deviance: 132.14  on 309  degrees of freedom
## AIC: 166.14
##
## Number of Fisher Scoring iterations: 10
```

```
model1.probtest <- predict(model1, newdata = testnew, type = "response")
model1.predtest <- ifelse(model1.probtest > 0.5, 1, 0)

# Confusion Matrix For Model 1

model1.cfmat <- confusionMatrix(data = model1.predtest, reference = as.factor(testnew$target),
    positive = "1")


model1cf_p1 <- as.data.frame(model1.cfmat$overall)
model1cf_p2 <- as.data.frame(model1.cfmat$byClass)
colnames(model1cf_p1) <- "Model1"
colnames(model1cf_p2) <- "Model1"

model1cf_p <- rbind(model1cf_p1, model1cf_p2)

coefficients(model1)
```

```
##   (Intercept)            zn         indus         chas1           nox
##  6.943472e+03 -1.050629e-02 -1.222318e-01  6.354365e-02  3.828755e+03
##            rm           age           dis           rad           tax
## -5.255610e-01  3.113089e-02  7.055724e-01  7.615392e-01 -8.764971e-03
##       ptratio         black         lstat          medv        znnew1
##  2.591974e-01 -8.775191e-03  1.660595e-02  9.845240e-02 -1.943361e+00
##        lognox     sqrtindus
##  1.886088e+03 -1.068802e+04
```

```
exp(model1$coefficients)
```

```
## (Intercept)          zn       indus       chas1         nox          rm
##         Inf   0.9895487   0.8849432   1.0656060         Inf   0.5912236
##         age         dis         rad         tax     ptratio       black
##   1.0316205   2.0250055   2.1415701   0.9912733   1.2958896   0.9912632
##       lstat        medv      znnew1      lognox   sqrtindus
##   1.0167446   1.1034619   0.1432218         Inf   0.0000000
```

```
# Finding Log Likelihoos, AIC and BIC

loglikm1 <- logLik(model1)
aicm1 <- AIC(model1)
bicm1 <- BIC(model1)
```

From the summary, we find that the "nox" (nitrogen oxide concentrations in environment) has quite a high positive effect on the crime rate of neighbourhood, with high levels of nox denoting high crime rate.

We do see that some of the variables like "chas" are not showing any significance. The "dis" (distance form employment centres) and "rad" (access to highways) seem to have some signigicance. We also see tax as a significant predictor.

Overall this model has **0.900 accurancy and 166.1 AIC** . The **area under curve is 0.97133** which is pretty good. **Classification Error Rate : 0.1**

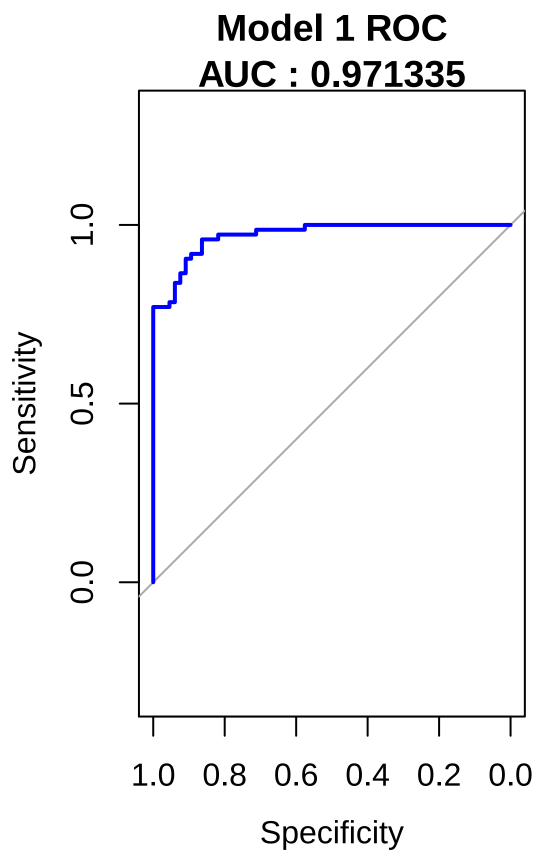
We do find some predictors with very less significance We will further work with newer reduced models by removing the less significant predictors and observe the changes.

Plotting the ROC curve for Model 1

```
model1roc <- roc(target ~ model1.probtest, data = testnew)
aucmodel1 <- round(auc(model1roc), 6)

par(mfrow = c(1, 2))

pander(ftable(model1.cfmat$table))
```

	"Reference"	"0"	"1"
"Prediction"			
"0"		60	8
"1"		6	66

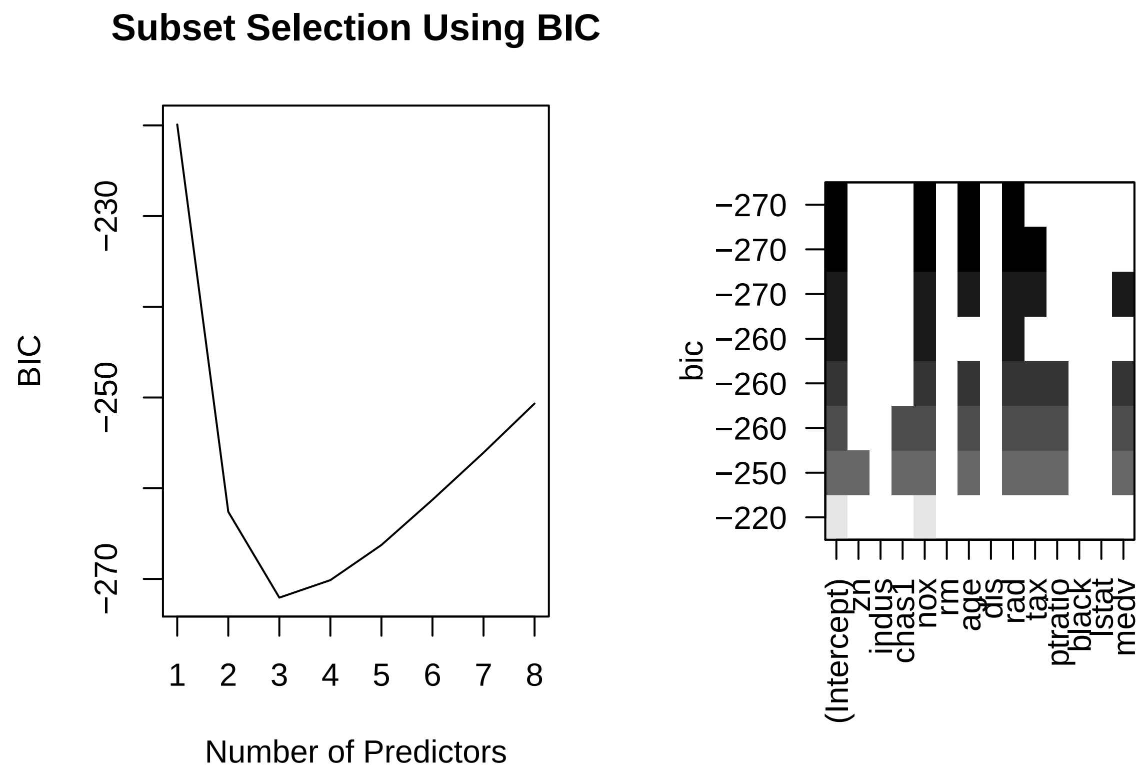
```
plot(model1roc, legacy_axes = TRUE, col = "blue", main = paste0("Model 1 ROC", "\n",
    "AUC : ", aucmodel1))
```



1.5.2 Model 2 : Bayesian Information Criterion

We create this model with the Bayesian Information Criterion (BIC) to determine the number of predictors to use and which predictors should be used. We use the original observation without the new added transformations

```
# bictrain <- dplyr::select(trainnew, -lognox,-sqrtindus,-znnew)


regfit.full <- regsubsets(factor(target) ~ . - znnew - lognox - sqrtindus, data = trainnew)
reg.summary <- summary(regfit.full)

par(mfrow = c(1, 2))

plot(reg.summary$bic, xlab = "Number of Predictors", ylab = "BIC", type = "l", main = "Subset Selection

plot(regfit.full)
```



From the plots we find the 3 predictors that minimize BIC are the "nox", "age" , and "rad" and heance we create a model with these 3 variables.

```
model2 <- glm(target ~ nox + age + rad, family = binomial, data = trainnew)

summary(model2)
```

```
##
## Call:
## glm(formula = target ~ nox + age + rad, family = binomial, data = trainnew)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.85064  -0.35055  -0.08079   0.00568   2.72758
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -16.78878    2.44367  -6.870 6.41e-12 ***
## nox          22.98021    4.63882   4.954 7.27e-07 ***
## age           0.01716    0.01091   1.573    0.116
## rad           0.56944    0.12907   4.412 1.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.15  on 325  degrees of freedom
```

```
## Residual deviance: 166.55  on 322  degrees of freedom
## AIC: 174.55
##
## Number of Fisher Scoring iterations: 8
```

```r
model2.probtest <- predict(model2, newdata = testnew, type = "response")
model2.predtest <- ifelse(model2.probtest > 0.5, 1, 0)

# Confusion Matrix For Model 2

model2.cfmat <- confusionMatrix(data = model2.predtest, reference = as.factor(testnew$target),
    positive = "1")

model2cf_p1 <- as.data.frame(model2.cfmat$overall)
model2cf_p2 <- as.data.frame(model2.cfmat$byClass)
colnames(model2cf_p1) <- "Model2"
colnames(model2cf_p2) <- "Model2"

model2cf_p <- rbind(model2cf_p1, model2cf_p2)

# Finding Log Likelihoos, AIC and BIC

loglikm2 <- logLik(model2)
aicm2 <- AIC(model2)
bicm2 <- BIC(model2)
```

We observe that "nox" , the nitrogen oxide concentrations have the strongest impact as per the coefficients depicted by this model. Also "nox" is statistically significant , as is "rad", access to highways.

We dont see "age" having so much of an impact and is statistically hardly significant.

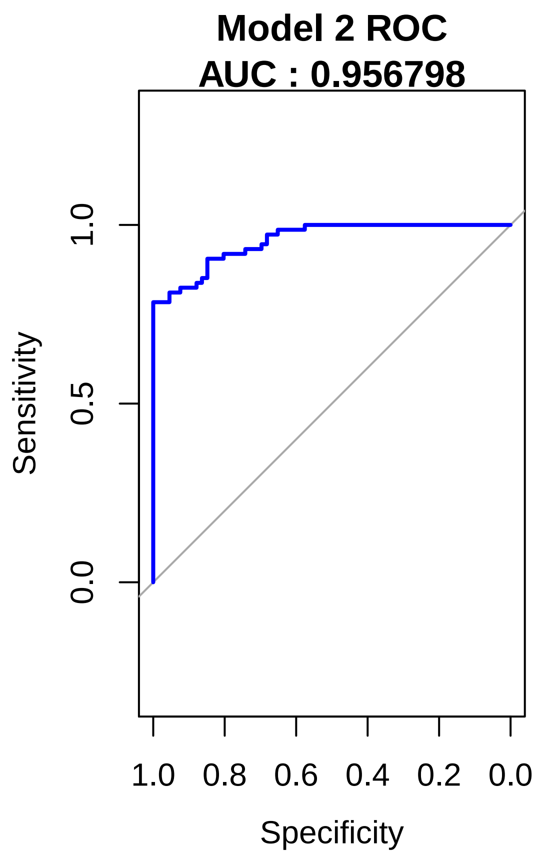
Overall this model has **0.8642 accurancy** and **174.55 AIC .** The **area under curve is 0.95679** which is pretty good. **Classification Error Rate : 0.1357**

Plotting the ROC curve for Model 2

```r
model2roc <- roc(target ~ model2.probtest, data = testnew)
aucmodel2 <- round(auc(model2roc), 6)

par(mfrow = c(1, 2))

pander(ftable(model2.cfmat$table))
```

	"Reference"	"0"	"1"
"Prediction"			
"0"		60	13
"1"		6	61

```r
plot(model2roc, legacy_axes = TRUE, col = "blue", main = paste0("Model 2 ROC", "\n",
    "AUC : ", aucmodel2))
```



1.5.3 Model 3 : Bayesian Information Criterion with Transformations

We construct this model based on the same BIC selection from Model 2 but with the transformations of applicable variables done earlier instead of the original predictor variables.

Based on the distributions, the log of "nox" has been used

```
model3 <- glm(target ~ lognox + age + rad, family = binomial, data = trainnew)

summary(model3)
```

```
##
## Call:
## glm(formula = target ~ lognox + age + rad, family = binomial,
##     data = trainnew)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.83103  -0.33224  -0.07184   0.00528   2.74148
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.53435    2.10766   1.677   0.0936 .
## lognox      12.54647    2.48814   5.043 4.59e-07 ***
## age          0.01542    0.01104   1.396   0.1626
```

```
## rad          0.57213    0.13034   4.390 1.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.15  on 325  degrees of freedom
## Residual deviance: 165.55  on 322  degrees of freedom
## AIC: 173.55
##
## Number of Fisher Scoring iterations: 8
```

```r
model3.probtest <- predict(model3, newdata = testnew, type = "response")
model3.predtest <- ifelse(model3.probtest > 0.5, 1, 0)


# Confusion Matrix For Model 3

model3.cfmat <- confusionMatrix(data = model3.predtest, reference = as.factor(testnew$target),
    positive = "1")


model3cf_p1 <- as.data.frame(model3.cfmat$overall)
model3cf_p2 <- as.data.frame(model3.cfmat$byClass)
colnames(model3cf_p1) <- "Model3"
colnames(model3cf_p2) <- "Model3"

model3cf_p <- rbind(model3cf_p1, model3cf_p2)

# Finding Log Likelihoos, AIC and BIC

loglikm3 <- logLik(model3)
aicm3 <- AIC(model3)
bicm3 <- BIC(model3)
```

The coefficient associated with nitrogen oxide concentration has decreased in value bus still shows high impact and statistically significant. There is a very slight increase in the "rad" coefficient, however the "age" seems to be as depicted in earlier model, less significant.

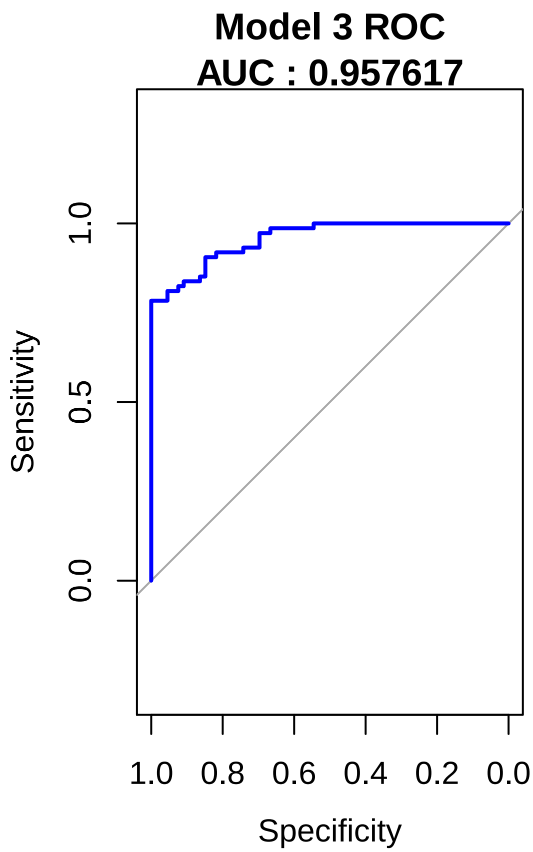
Overall this model has **0.8642 accuracy and 173.55 AIC** which is an improvement over the earlier Model 2. The **area under curve is 0.95761. Classification Error Rate : 0.1357** i.e it is same as Model 2

Plotting the ROC curve for Model 3

```r
model3roc <- roc(target ~ model3.probtest, data = testnew)
aucmodel3 <- round(auc(model3roc), 6)

par(mfrow = c(1, 2))
pander(ftable(model3.cfmat$table))
```

	"Reference"	"0"	"1"
"Prediction"			
"0"		60	13
"1"		6	61

```r
plot(model3roc, legacy_axes = TRUE, col = "blue", main = paste0("Model 3 ROC", "\n",
    "AUC : ", aucmodel3))
```



1.5.4 Model 4 : Reduced model without transformed predictor variables

As our first model, we construct this using the significant predictor variables and also get rid of the new transformed "znnew" (categorized residential zoned lots) and "lognox" which is log(nox) (nitroden oxide concentrations) and the "sqrtindus" , which is sqrt(indus) (non-retail businees acres / industrial)

Deriving from the Model 1 and correlation matrix, we remove the "rm" variable as it does not seem to be affecting target so much. Also working backwards, we remove the "chas",black" and "zn" as their significance seems to be pretty less (they have high p -values)

```r
model4 <- glm(target ~ . - znnew - lognox - sqrtindus - rm - zn - black - chas, family = binomial(link =
    data = trainnew)

summary(model4)
```

```
##
## Call:
## glm(formula = target ~ . - znnew - lognox - sqrtindus - rm -
##     zn - black - chas, family = binomial(link = "logit"), data = trainnew)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.1812  -0.1769  -0.0142   0.0015   3.1931
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.893008   7.631515  -4.965 6.86e-07 ***
## indus        -0.107358   0.057997  -1.851  0.06416 .
## nox          52.117094  10.623430   4.906 9.30e-07 ***
## age           0.028996   0.014228   2.038  0.04156 *
## dis           0.313110   0.226160   1.384  0.16622
## rad           0.781203   0.177479   4.402 1.07e-05 ***
## tax          -0.008939   0.003448  -2.592  0.00953 **
## ptratio       0.307286   0.129227   2.378  0.01741 *
## lstat        -0.008794   0.065862  -0.134  0.89379
## medv          0.039711   0.051991   0.764  0.44498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.15  on 325  degrees of freedom
## Residual deviance: 139.99  on 316  degrees of freedom
## AIC: 159.99
##
## Number of Fisher Scoring iterations: 9
```

We further update the model to get rid of the "lstat" (lower status of population) and "medv" (median home values) and industrialization "indus" predictor variables.

```
model41 <- update(model4, . ~ . - lstat - medv - indus)
summary(model41)
```

```
##
## Call:
## glm(formula = target ~ nox + age + dis + rad + tax + ptratio,
##     family = binomial(link = "logit"), data = trainnew)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.06400  -0.19689  -0.01390   0.00106   2.97644
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -29.57908    5.73156  -5.161 2.46e-07 ***
## nox          39.96591    7.94984   5.027 4.98e-07 ***
## age           0.02376    0.01235   1.924 0.054347 .
## dis           0.19124    0.20606   0.928 0.353373
## rad           0.88241    0.17369   5.080 3.77e-07 ***
## tax          -0.01071    0.00292  -3.666 0.000246 ***
## ptratio       0.24249    0.10778   2.250 0.024461 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.15  on 325  degrees of freedom
```

```
## Residual deviance: 145.32  on 319  degrees of freedom
## AIC: 159.32
##
## Number of Fisher Scoring iterations: 9
```

```r
model42 <- update(model41, . ~ . - age - dis)
summary(model42)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio, family = binomial(link = "logit"),
##     data = trainnew)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.10993  -0.20377  -0.01760   0.00073   2.79058
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -27.838609   4.615010  -6.032 1.62e-09 ***
## nox          40.563924   6.232097   6.509 7.57e-11 ***
## rad           0.873603   0.168010   5.200 2.00e-07 ***
## tax          -0.010110   0.002763  -3.659 0.000253 ***
## ptratio       0.256855   0.107651   2.386 0.017033 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.15  on 325  degrees of freedom
## Residual deviance: 149.31  on 321  degrees of freedom
## AIC: 159.31
##
## Number of Fisher Scoring iterations: 9
```

We now have the most significant predictor variables , "nox", "rad", "tax", "ptratio" in the model. And we proceed with these to analyze further.

Although the tax has a negative impact on the target, and is statistically significant, the coeefficient details that there is only 0.01 unit decrease in crime rate with every 1 unit increase in tax.

```r
model4 <- model42

model4.probtest <- predict(model4, newdata = testnew, type = "response")
model4.predtest <- ifelse(model4.probtest > 0.5, 1, 0)

# Confusion Matrix For Model 1

model4.cfmat <- confusionMatrix(data = model4.predtest, reference = as.factor(testnew$target),
    positive = "1")


model4cf_p1 <- as.data.frame(model4.cfmat$overall)
model4cf_p2 <- as.data.frame(model4.cfmat$byClass)
colnames(model4cf_p1) <- "Model4"
colnames(model4cf_p2) <- "Model4"
```

```r
model4cf_p <- rbind(model4cf_p1, model4cf_p2)

# Finding Log Likelihoos, AIC and BIC

loglikm4 <- logLik(model4)
aicm4 <- AIC(model4)
bicm4 <- BIC(model4)
```

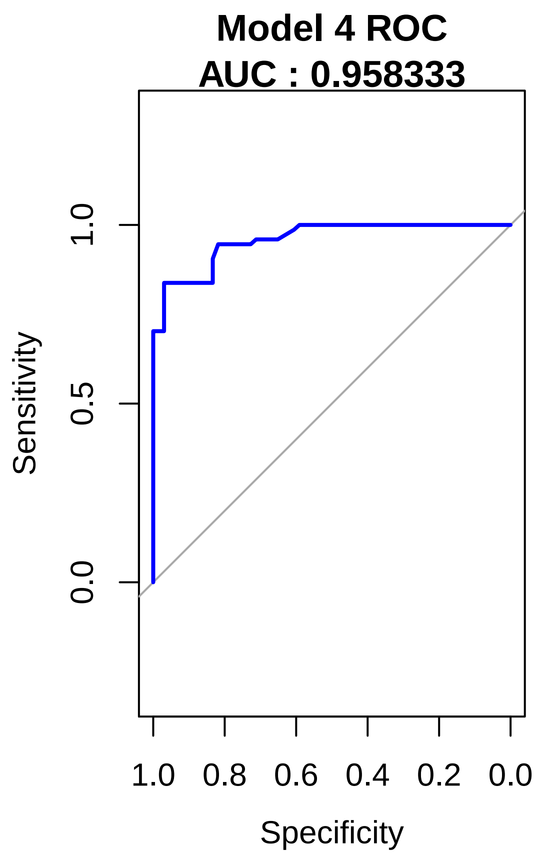
Overall this model has **0.8714 accurancy and 159.31 AIC** which is an improvement over the earlier Model 2 and Model 1 respectively. The **area under curve is 0.95833 Classification Error Rate : 0.128** which is lesser compared to Model 2 and Model 3.

Plotting the ROC curve for Model 4

```r
model4roc <- roc(target ~ model4.probtest, data = testnew)
aucmodel4 <- round(auc(model4roc), 6)

par(mfrow = c(1, 2))
pander(model4.cfmat$table)
```

	0	1
0	60	12
1	6	62

```r
plot(model4roc, legacy_axes = TRUE, col = "blue", main = paste0("Model 4 ROC", "\n",
    "AUC : ", aucmodel4))
```



```
# trainnewbestglm <- dplyr::select(trainnew, -c(znnew,lognox,sqrtindus))

# model5 <- bestglm(trainnewbestglm, IC= 'BIC', family = binomial)

# summary(model5$BestModel)


# bestglm model slow
```

1.6 Model Selection

From the four models derived above, we look at the performance of each of these through cross validation, with respect to the Accuracy, Area Under Curve, Log likelihood, the AIC(Akaike Information Criterion) and BIC. We compare the Sensitiviy, Specificity

Confusion Matrix Metrics For All Models

```
cfmatmetricsdf <- cbind(model1cf_p, model2cf_p, model3cf_p, model4cf_p)
kable(cfmatmetricsdf, caption = "Confusion Matrix Metrics For All Models")
```

Table 9: Confusion Matrix Metrics For All Models

	Model1	Model2	Model3	Model4
Accuracy	0.9000000	0.8642857	0.8642857	0.8714286
Kappa	0.7996729	0.7292345	0.7292345	0.7432763
AccuracyLower	0.8379103	0.7962034	0.7962034	0.8044311
AccuracyUpper	0.9442424	0.9162713	0.9162713	0.9219873
AccuracyNull	0.5285714	0.5285714	0.5285714	0.5285714
AccuracyPValue	0.0000000	0.0000000	0.0000000	0.0000000
McnemarPValue	0.7892680	0.1686686	0.1686686	0.2385928
Sensitivity	0.8918919	0.8243243	0.8243243	0.8378378
Specificity	0.9090909	0.9090909	0.9090909	0.9090909
Pos Pred Value	0.9166667	0.9104478	0.9104478	0.9117647
Neg Pred Value	0.8823529	0.8219178	0.8219178	0.8333333
Precision	0.9166667	0.9104478	0.9104478	0.9117647
Recall	0.8918919	0.8243243	0.8243243	0.8378378
F1	0.9041096	0.8652482	0.8652482	0.8732394
Prevalence	0.5285714	0.5285714	0.5285714	0.5285714
Detection Rate	0.4714286	0.4357143	0.4357143	0.4428571
Detection Prevalence	0.5142857	0.4785714	0.4785714	0.4857143
Balanced Accuracy	0.9004914	0.8667076	0.8667076	0.8734644

Area Under Curve Comparison For All Models

```
vif(model4)
```

```
##      nox      rad      tax  ptratio
## 2.126467 1.875034 1.583708 1.264045
```

```
AUCAll <- rbind(aucmodel1, aucmodel2, aucmodel3, aucmodel4)

LogLikAll <- rbind(loglikm1, loglikm2, loglikm3, loglikm4) %>% round(2)


AICAll <- rbind(aicm1, aicm2, aicm3, aicm4) %>% round(2)


BICAll <- rbind(bicm1, bicm2, bicm3, bicm4) %>% round(2)


comptable <- cbind(AUCAll, LogLikAll, AICAll, BICAll)

rownames(comptable) <- c("Model 1", "Model 2", "Model 3", "Model 4")
colnames(comptable) <- c("Area Under Curve", "Log Likelihood", "AIC", "BIC")
```

```r
pander(comptable, caption = "Model Comparison: AUC / Log Likelihood / AIC / BIC")
```

Table 10: Model Comparison: AUC / Log Likelihood / AIC / BIC

	Area Under Curve	Log Likelihood	AIC	BIC
Model 1	0.9713	-66.07	166.1	230.5
Model 2	0.9568	-83.27	174.6	189.7
Model 3	0.9576	-82.77	173.6	188.7
Model 4	0.9583	-74.65	159.3	178.2

```r
vif(model4)
```

```
##      nox      rad      tax  ptratio
## 2.126467 1.875034 1.583708 1.264045
```

We did see a lot of Multicollinearity in Model 1, this has been taken care though in Model 2, 3, 4 The accuracy of Model 1 although pretty good at 90% , Model 4 close to it at 87%. Model 1 also has a higher Area under curve as compared to Model 4.

Model 2 and Model 3 are pretty good in handling the multicollinearity issues, however the Accuracy and Are under curve as compared ot Model 4 is still less.

While Specificity is similar for all models , Model 1 excel in Sensitivty, followed by Model 4. Model 1 also is better is Classification Error rate , it has the least compared to all the other models. And better F1 score.

Model 4 seems to be the Model to go ahead with as it is good at the muilticollinearity with Variance Inflation Factors analyzed , all predictor score below 4 , which proves it. Also the Residual Deviance is least for Model 4. Also , as compared to all models and especially with Model 1, it has the lowest AIC and BIC and high log likelihood. Also it is parsimonous with a decent Accuracy, AUC, F1 score, predictive power.

Reference : (https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/)

1.7 Predictions on Evaluation Data

We now use this Model 4 for predicting the crime evaluation dataset.

```
pred_evaldata <- predict(model4, newdata = evaldata, type = "response")
pred_evaldata_target <- ifelse(pred_evaldata > 0.5, 1, 0)
evaldata$target <- pred_evaldata_target

pander(table(evaldata$target))
```

0	1
20	20

```
pander(table(evaldata$target)/sum(table(traindata$target)))
```

0	1
0.04292	0.04292

```
write.csv(evaldata, "predicted_crime_evaluation.csv")
# View(evaldata)
```

We find that after applying our Model 4 to evaluation crime data, the predicted values for crime neighbourhoods is 20 for low crime (valued 0) and 20 for high crime (valued 1). The results of the predicted values are stored in new file *predicted_crime_evaluation.csv*