Assignment-1

Anil Akyildirim, John K. Hancock, John Suh, Emmanuel Hayble-Gomes, Chunjie Nan

04/05/2020

Contents

Introduction

In this assignment, we are tasked to explore, analyze and model a data set containing information on crime for various neighborhoods of a major city. Each record has a response variable indicating whether or not the crime rate is above the median crime rate (1) or not (0). The objective is to build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels. The expectation is to provide classifications and probabilities for the evaluation data set using the binary logistic regression model.

About the Data

The data set are provided in csv format as crime-evaluation-data and crime-training-data where we will explore, prepare and create our Binary Logistic Regression models with the training data using the variables given below:

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

Load Libraries

```
library(ggplot2)
library(ggcorrplot)
library(corrplot)
library(psych)
library(dplyr)
library(tidyr)
library(caret)
library(MASS)
library(pROC)
library(glmnet)
library(mltest)
```

Load the training and evaluation data sets

I will use the training data to train the model and use the evaluation data set to test/evaluate the model.

```
crime <- read.csv("https://raw.githubusercontent.com/Emahayz/Data-621/master/crime-training-data_modifi

crime_evaluation <- read.csv("https://raw.githubusercontent.com/Emahayz/Data-621/master/crime-evaluatio
```

Data Exploration

Descriptive Statistics

We can start exploring our training data set by looking at basic descriptive statistics. Look at the training dataset structure

```
str(crime)
```

```
## 'data.frame':    466 obs. of  13 variables:
##  $ zn     : num  0 0 0 30 0 0 0 0 0 80 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : num  96.2 100 100 7.8 92.2 71.3 100 100 38.1 19.1 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ rad    : int  5 5 24 6 3 5 24 24 5 1 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

The training data set has 466 observations with 13 variables. All the variables are numeric/integer. Look at the evaluation dataset structure

```
str(crime_evaluation)
```

```
## 'data.frame':    40 obs. of  12 variables:
##  $ zn     : int  0 0 0 0 0 25 25 0 0 0 ...
##  $ indus  : num  7.07 8.14 8.14 8.14 5.96 5.13 5.13 4.49 4.49 2.89 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.469 0.538 0.538 0.538 0.499 0.453 0.453 0.449 0.449 0.445 ...
##  $ rm     : num  7.18 6.1 6.5 5.95 5.85 ...
##  $ age    : num  61.1 84.5 94.4 82 41.5 66.2 93.4 56.1 56.8 69.6 ...
##  $ dis    : num  4.97 4.46 4.45 3.99 3.93 ...
##  $ rad    : int  2 4 4 4 5 8 8 3 3 2 ...
##  $ tax    : int  242 307 307 307 279 284 284 247 247 276 ...
##  $ ptratio: num  17.8 21 21 21 19.2 19.7 19.7 18.5 18.5 18 ...
##  $ lstat  : num  4.03 10.26 12.8 27.71 8.77 ...
##  $ medv   : num  34.7 18.2 18.4 13.2 21 18.7 16 26.6 22.2 21.4 ...
```

The evaluation data set has 40 observations with 12 variables; all the bariables are numerical/integers.

Look at descriptive statistics for both datasets

```
summary(crime)
```

```
##        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         lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```r
summary(crime_evaluation)
```

```
##        zn             indus             chas           nox        
##  Min.   : 0.000   Min.   : 1.760   Min.   :0.00   Min.   :0.3850  
##  1st Qu.: 0.000   1st Qu.: 5.692   1st Qu.:0.00   1st Qu.:0.4713  
##  Median : 0.000   Median : 8.915   Median :0.00   Median :0.5380  
##  Mean   : 8.875   Mean   :11.507   Mean   :0.05   Mean   :0.5592  
##  3rd Qu.: 0.000   3rd Qu.:18.100   3rd Qu.:0.00   3rd Qu.:0.6258  
##  Max.   :90.000   Max.   :25.650   Max.   :1.00   Max.   :0.7400  
##        rm             age              dis             rad        
##  Min.   :3.561   Min.   :  6.80   Min.   :1.202   Min.   : 1.000  
##  1st Qu.:5.874   1st Qu.: 56.62   1st Qu.:2.041   1st Qu.: 4.000  
##  Median :6.143   Median : 83.25   Median :3.373   Median : 5.000  
##  Mean   :6.214   Mean   : 70.99   Mean   :3.787   Mean   : 9.775  
##  3rd Qu.:6.532   3rd Qu.: 93.10   3rd Qu.:4.527   3rd Qu.:24.000  
##  Max.   :8.247   Max.   :100.00   Max.   :9.089   Max.   :24.000  
##       tax           ptratio          lstat             medv      
##  Min.   :188.0   Min.   :14.70   Min.   : 2.960   Min.   : 8.40  
##  1st Qu.:276.8   1st Qu.:18.40   1st Qu.: 6.435   1st Qu.:16.98  
##  Median :307.0   Median :19.60   Median :11.685   Median :20.55  
##  Mean   :393.5   Mean   :19.12   Mean   :12.905   Mean   :21.88  
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:17.363   3rd Qu.:25.00  
##  Max.   :666.0   Max.   :21.20   Max.   :34.020   Max.   :50.00
```

With the descriptive statistics, we are able to see mean, standard deviation, median, min, max values.

Looking for missing values

```r
colSums(is.na(crime))
```

```
##      zn   indus    chas     nox      rm     age     dis     rad     tax 
##       0       0       0       0       0       0       0       0       0 
## ptratio   lstat    medv  target 
##       0       0       0       0
```

```r
colSums(is.na(crime_evaluation))
```

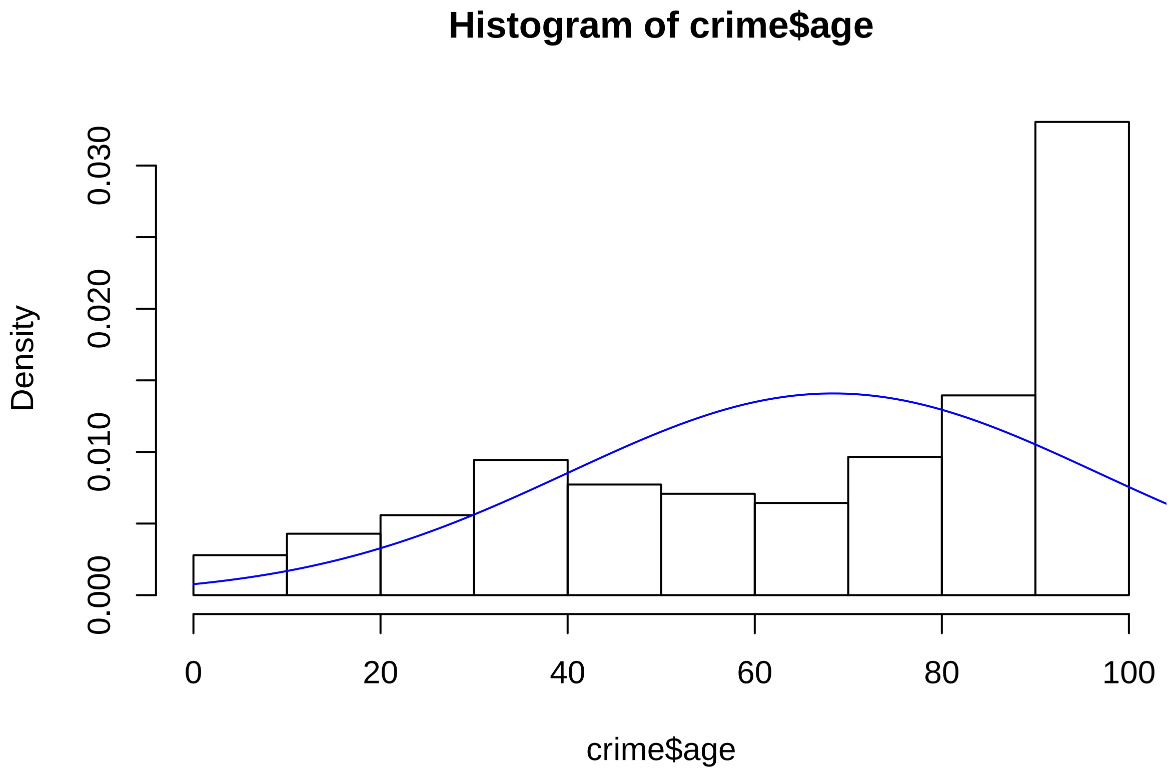
```
##      zn   indus    chas     nox      rm     age     dis     rad     tax 
##       0       0       0       0       0       0       0       0       0 
## ptratio   lstat    medv 
##       0       0       0
```

The data set shows no missing values for these data sets.

Let's look at some interesting part of the data by exploring the Age and Property Tax variables

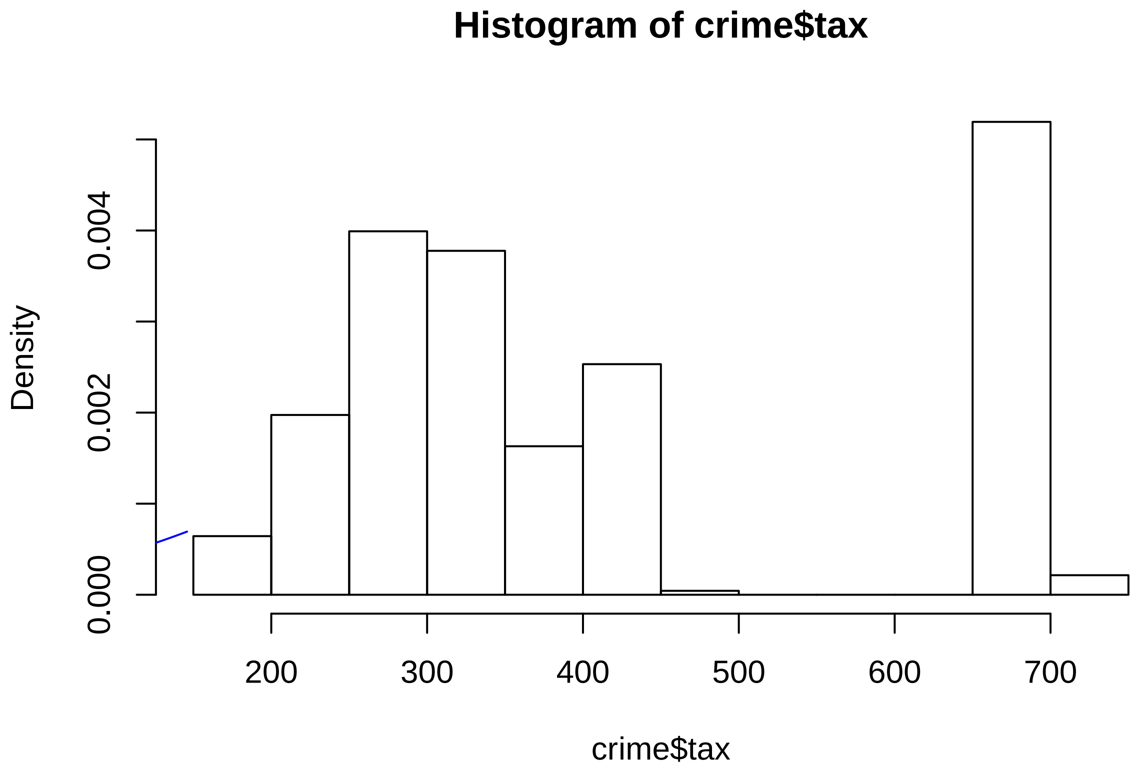
```r
mean = mean(crime$age)
  sd = sd(crime$age)

hist(crime$age, probability = TRUE)
x <- 0:146
y <- dnorm(x = x, mean = mean, sd = sd)
lines(x = x, y = y, col = "blue") #The distribution doesn't looks normal!
```



```
mean = mean(crime$tax)
  sd = sd(crime$tax)

hist(crime$tax, probability = TRUE)
x <- 0:146
y <- dnorm(x = x, mean = mean, sd = sd)
lines(x = x, y = y, col = "blue") # Doesn't looks good here too!
```



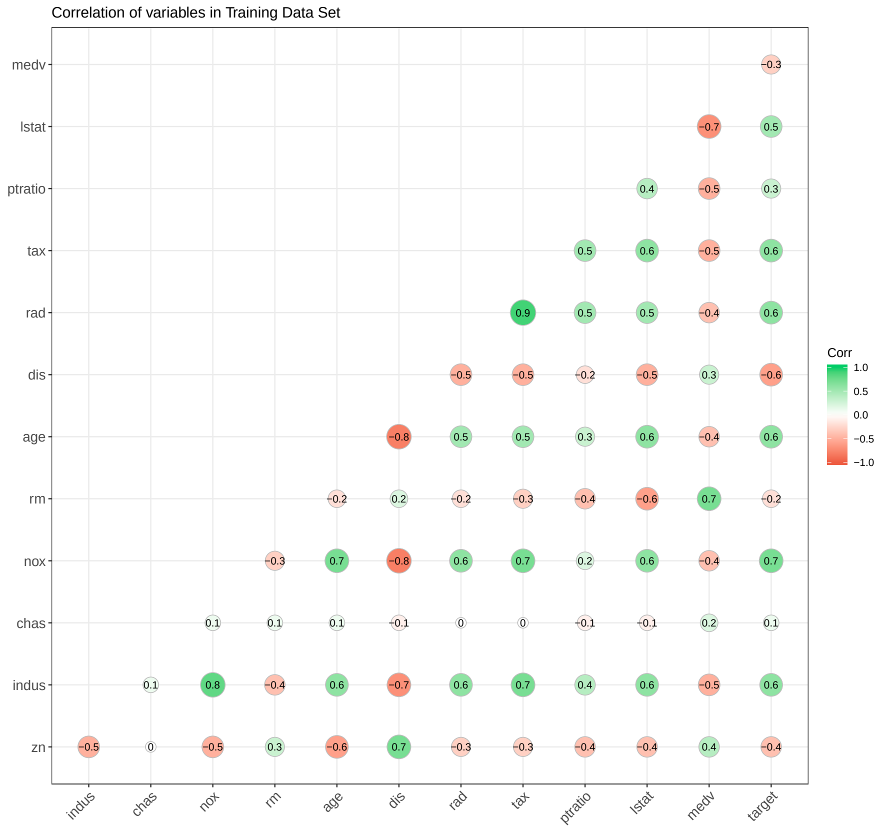
Correlation and Distribution

The approach below gives the following correlation for these variables

```
# Look at correlation between variables

corr <- round(cor(crime), 1)

ggcorrplot(corr,
           type="lower",
           lab=TRUE,
           lab_size=3,
           method="circle",
           colors=c("tomato2", "white", "springgreen3"),
           title="Correlation of variables in Training Data Set",
           ggtheme=theme_bw)
```



Since the logistic regression requires there to be little or no multicollinearity among the independent variables. The variables rad and tax have a correlation of about 90%, I will drop one of these variables for my model.

Data Preparation

In this section, we will prepare the dataset for logistic regression modeling. Logistic regression does not make many of the key assumptions of linear regression and general linear models that are based on ordinary least squares algorithms – particularly regarding linearity, normality, homoscedasticity, and measurement level.

Objective

The data for Logistic Regression doesn't have to be normally distrubuted. Hence, I will not be transforming these variables as transformation may lead to the risk of complicating the interpretation of the coefficients and Odds Ratios associated to the transformed covariates. However, I will drop the rad variable due to very high correlation with the tax variable.

```
crime1 = crime[,!(names(crime) %in% c("rad"))]
str(crime1)
```

```
## 'data.frame':    466 obs. of  12 variables:
##  $ zn     : num  0 0 0 30 0 0 0 0 0 80 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : num  96.2 100 100 7.8 92.2 71.3 100 100 38.1 19.1 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

The rad variable has been dropped to create the crime_train dataframe for building the models.

```r
table(crime1$target)
```

```
## 
##   0   1 
## 237 229
```

```r
prop.table(table(crime1$target))
```

```
## 
##         0         1 
## 0.5085837 0.4914163
```

Only 49% of the incident of crime is present in the dataset for this region.

Spliting the Crime dataset

Use 80% for training and 20% for testing the model

```r
set.seed(101)
train <- createDataPartition(y = crime1$target, p = 0.80, list = FALSE)
crime_train <- crime1[train,]
crime_test  <- crime1[-train,]
```

The Training dataset now has 373 observations with 12 variables and the testing has 93 observations with 12 variables.

Build Models

The Model is using all the variables, then applying Stepwise Variable Selection for additional Models.

Logistic Model with Stepwise Variable Selection

Using all the Eleven (11) variables for the Model with Stepwise variable selection in both direction produced three (3) models with different AIC values.

```r
crime_logit <- step(glm(target ~., data = crime_train, family = binomial(link="logit")), direction="bot
## Start:  AIC=214.84
## target ~ zn + indus + chas + nox + rm + age + dis + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - rm       1   190.92 212.92
## - age      1   192.72 214.72
## <none>         190.84 214.84
## - lstat    1   193.04 215.04
## - chas     1   195.10 217.10
## - zn       1   197.20 219.20
## - indus    1   197.42 219.42
## - ptratio  1   199.76 221.76
## - medv     1   201.38 223.38
## - tax      1   201.61 223.61
## - dis      1   204.34 226.34
## - nox      1   256.19 278.19
##
## Step:  AIC=212.92
## target ~ zn + indus + chas + nox + age + dis + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - age      1   192.88 212.88
## <none>         190.92 212.92
## - lstat    1   193.85 213.85
## + rm       1   190.84 214.84
## - chas     1   195.15 215.15
## - zn       1   197.32 217.32
## - indus    1   197.42 217.42
## - ptratio  1   199.99 219.99
## - tax      1   201.93 221.93
## - dis      1   204.65 224.65
## - medv     1   210.58 230.58
## - nox      1   256.57 276.57
##
## Step:  AIC=212.88
## target ~ zn + indus + chas + nox + dis + tax + ptratio + lstat +
##     medv
##
##           Df Deviance    AIC
## <none>         192.88 212.88
## + age      1   190.92 212.92
## + rm       1   192.72 214.72
## - chas     1   197.42 215.42
## - lstat    1   197.50 215.50
## - indus    1   199.37 217.37
## - zn       1   199.42 217.42
## - ptratio  1   201.53 219.53
## - tax      1   203.26 221.26
## - dis      1   204.89 222.89
## - medv     1   211.77 229.77
## - nox      1   269.76 287.76
```

Using this method produced three models with AIC = 214.84, AIC = 212.92 and AIC = 212.88.

```
summary(crime_logit)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + dis + tax +
##     ptratio + lstat + medv, family = binomial(link = "logit"),
##     data = crime_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.26820  -0.32972  -0.02197   0.17668   3.05407
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -38.629311   5.951419  -6.491 8.54e-11 ***
## zn           -0.056837   0.028510  -1.994 0.046196 *
## indus        -0.128005   0.054563  -2.346 0.018976 *
## chas          1.596649   0.736385   2.168 0.030142 *
## nox          45.795167   7.255224   6.312 2.75e-10 ***
## dis           0.684610   0.204146   3.354 0.000798 ***
## tax           0.005868   0.001998   2.938 0.003307 **
## ptratio       0.318371   0.110555   2.880 0.003980 **
## lstat         0.098640   0.045892   2.149 0.031602 *
## medv          0.152086   0.037498   4.056 5.00e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.31  on 372  degrees of freedom
## Residual deviance: 192.88  on 363  degrees of freedom
## AIC: 212.88
##
## Number of Fisher Scoring iterations: 8
```

The best model has 9 variables with the lowest AIC at 212.88 and converged after the eighth iteration.

Viewing the summary of the Best Model gives the following information: The Variables for this Model are all significant since they have $\alpha < 0.05$.

There are three variables that seems to be more significant in predicting crime, the variables are nox: nitrogen oxides concentration of the area, dis: weighted mean of distances to five Boston employment centers and medv: median value of owner-occupied homes of the area.

Calculating the odd ratio and the variable importance

```
exp(cbind(Odds_Ratio=coef(crime_logit)))
```

```
##               Odds_Ratio
## (Intercept) 1.673030e-17
## zn          9.447483e-01
## indus       8.798493e-01
## chas        4.936461e+00
## nox         7.737277e+19
## dis         1.982998e+00
## tax         1.005886e+00
## ptratio     1.374886e+00
## lstat       1.103669e+00
## medv        1.164260e+00
```

```
varImp(crime_logit)
```

```
##          Overall
## zn      1.993595
## indus   2.346006
## chas    2.168224
## nox     6.312027
## dis     3.353522
## tax     2.937702
## ptratio 2.879755
## lstat   2.149414
## medv    4.055855
```

The variable importance shows that nox: nitrogen oxides concentration of the area, dis: weighted mean of distances to five Boston employment centers and medv: median value of owner-occupied homes of the area are more important in predicting the incidence of crime for this region.

Evaluating the Selected Model

Calculate the Predicted Probabilities

```
prediction <- predict(crime_logit,newdata = crime_train,type="response")
roccrime   <- roc(response = crime_train$target, predictor = prediction,
                  levels=base::levels(as.factor(crime_train$target)))
```

Calculate the Metrics or Fit Statistics for the model

```
predclass <-ifelse(prediction>coords(roccrime,"best")[1],1,0)
ConfMatrix <- table(Predicted = predclass,Actual = crime_train$target)
AccuracyRate <- sum(diag(ConfMatrix))/sum(ConfMatrix)
Gini <-2*auc(roccrime)-1
metric <- data.frame(c(coords(roccrime,"best"),AUC=auc
                          (roccrime),AccuracyRate=AccuracyRate,Gini=Gini))
metric <- data.frame(rownames(metric),metric)
rownames(metric) <-NULL
names(metric) <- c("Metric","Values")
metric
```

```
##          Metric    Values
## 1     threshold 0.4152553
## 2   specificity 0.8820513
## 3  sensitivity 0.9157303
## 4          AUC 0.9589744
## 5 AccuracyRate 0.8981233
## 6         Gini 0.9179487
```

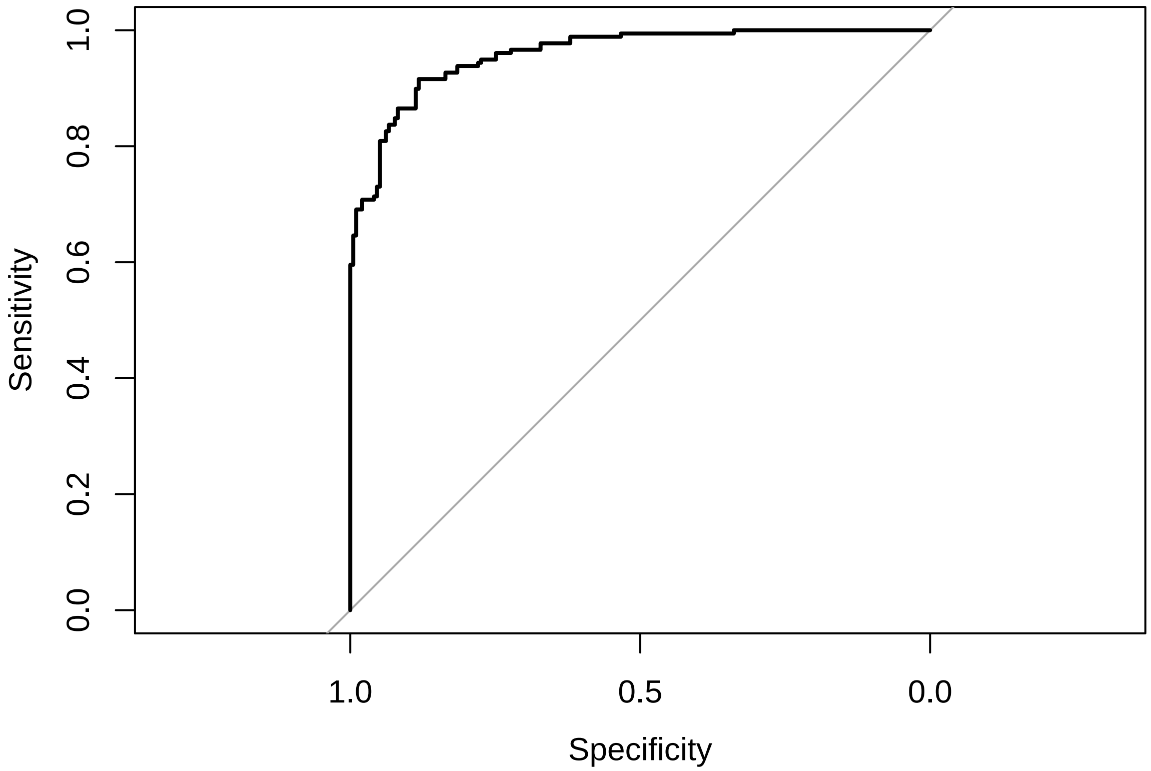
The Accuracy for the selected model is approximately 89%

Now lets view the Confusion Matrix and the ROC curve

```
ConfMatrix
```

```
##          Actual
## Predicted   0   1
##         0 172  15
##         1  23 163
```

```
plot(roccrime)
```



Using Test Dataset

```
prediction1 <- predict(crime_logit,newdata = crime_test,type="response")
roccrime1   <- roc(response = crime_test$target, predictor = prediction1,
                 levels=base::levels(as.factor(crime_test$target)))
```

```
predclass1 <-ifelse(prediction1>coords(roccrime1,"best")[1],1,0)
ConfMatrix1 <- table(Predicted = predclass1,Actual = crime_test$target)
AccuracyRate1 <- sum(diag(ConfMatrix1))/sum(ConfMatrix1)
Gini1 <-2*auc(roccrime1)-1
metric1 <- data.frame(c(coords(roccrime1,"best"),AUC=auc
                          (roccrime1),AccuracyRate=AccuracyRate1,Gini=Gini1))
metric1 <- data.frame(rownames(metric1),metric1)
rownames(metric1) <-NULL
names(metric1) <- c("Metric","Values")
metric1
```

```
##          Metric    Values
## 1    threshold 0.1785224
## 2  specificity 0.8095238
## 3  sensitivity 0.9803922
## 4          AUC 0.9607843
## 5 AccuracyRate 0.9032258
## 6         Gini 0.9215686
```

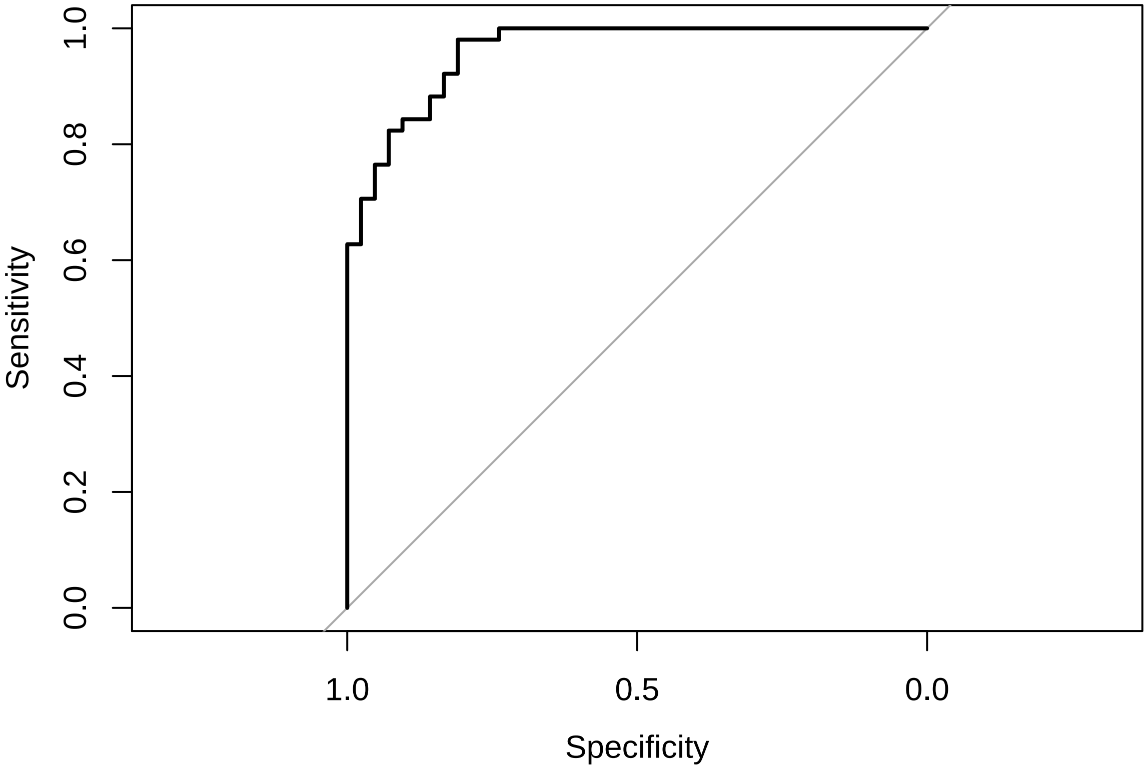
The Accuracy using the Test Data is at 87%

Viewing the Confusion Matrix using the Test data

```
ConfMatrix1
```

```
##           Actual
## Predicted  0  1
##         0 34  1
##         1  8 50
```

```
plot(roccrime1)
```



Complete Classification Metrics

This is the complete classification metrics using the Test data set for the best Model.

```
Predicted <- predclass1
Actual <- crime_test$target
classifier_metrics <- ml_test(predicted = Predicted, true = Actual, output.as.table = FALSE)
classifier_metrics
```

```
## $accuracy
## [1] 0.9032258
##
## $balanced.accuracy
##        0        1
## 0.894958 0.894958
##
## $DOR
##     0     1
## 212.5 212.5
##
## $error.rate
## [1] 0.09677419
##
## $F0.5
##         0         1
## 0.9340659 0.8833922
##
## $F1
##         0         1
## 0.8831169 0.9174312
##
## $F2
##         0         1
## 0.8374384 0.9541985
##
## $FDR
##          0          1
## 0.02857143 0.13793103
##
## $FNR
##          0          1
## 0.19047619 0.01960784
##
## $FOR
##          0          1
## 0.13793103 0.02857143
##
## $FPR
##          0          1
## 0.01960784 0.19047619
##
## $geometric.mean
##         0         1
## 0.8908708 0.8908708
##
## $Jaccard
##         0         1
## 0.7906977 0.8474576
##
## $L
##         0         1
## 41.285714  5.147059
##
## $lambda
##          0          1
## 0.19428571 0.02422145
##
## $MCC
##         0         1
## 0.8114142 0.8114142
##
## $MK
##         0         1
## 0.8334975 0.8334975
##
## $NPV
##         0         1
## 0.8620690 0.9714286
##
```

```
## $OP
##         0         1
## 0.8077641 0.8077641
##
## $precision
##         0         1
## 0.9714286 0.8620690
##
## $recall
##         0         1
## 0.8095238 0.9803922
##
## $specificity
##         0         1
## 0.9803922 0.8095238
##
## $Youden
##        0        1
## 0.789916 0.789916
```

Model Prediction

Making predictions using the evaluation data set.

```
prediction <- predict(crime_logit,newdata = crime_evaluation,type="response")
new_target <-ifelse(prediction >= 0.5, 1, 0)
eval <- data.frame(prediction, new_target)
eval
```

```
##      prediction new_target
## 1  0.1308119868          0
## 2  0.5716478532          1
## 3  0.6375187632          1
## 4  0.7163337109          1
## 5  0.1152534989          0
## 6  0.0504283170          0
## 7  0.0293859467          0
## 8  0.0272525623          0
## 9  0.0106846825          0
## 10 0.0109437045          0
## 11 0.1607285596          0
## 12 0.1368350360          0
## 13 0.8579214830          1
## 14 0.4871226941          0
## 15 0.3412435333          0
## 16 0.3123657411          0
## 17 0.1781816970          0
## 18 0.7976958627          1
## 19 0.1270709485          0
## 20 0.0001861031          0
## 21 0.0003330540          0
## 22 0.0109801884          0
## 23 0.1703090856          0
## 24 0.1923608782          0
## 25 0.2209243289          0
## 26 0.4343250221          0
## 27 0.0016133151          0
## 28 0.9997422455          1
## 29 0.9998170415          1
## 30 0.8158010639          1
## 31 0.9998949848          1
## 32 0.9998382508          1
## 33 0.9997605909          1
## 34 0.9998871085          1
## 35 0.9998100511          1
## 36 0.9995220393          1
## 37 0.9996437906          1
## 38 0.9974687044          1
## 39 0.8148915809          1
## 40 0.5117856834          1
```

```
table(new_target)
```

```
## new_target
##  0  1
## 22 18
```

```
prop.table(table(new_target))
```

```
## new_target
##    0    1
## 0.55 0.45
```

There are 22 incidence predicted as “0” No Crime and 18 incidence predicted as “1” Crime. These values corresponds to 55% and 45% respectively.

Conclusion

Evaluating the selected model using the training dataset shows that the model is performing at optimal level. The Accuracy of the model was obtained at 0.8981 which is about 89%. The model also shows that the Area Under the Curve (AUC) is 0.9589 which is 96%.

The model was able to accurately predict the True Positive (TP) at 163 and True Negative (TN) at 172. The Specificity (SP) of the Model was calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR) and was 0.8821 (88%) while the Sensitivity (SN) of the model was calculated as the number of correct positive predictions divided by the total number of positives. It is also called Recall (REC) or true positive rate (TPR) and was 0.9157 (92%).

The Type I (FP) and Type II (FN) Errors were obtained as 23 and 15 respectively. The False positive rate (FPR) was calculated as the number of incorrect positive predictions divided by the total number of negatives which is 1 – specificity (1-SP)-> 1 - 0.8821 = 0.1179 which is quite low. False negative rate (FNR) is calculated as the number of incorrect negative predictions divided by the total number of positives which is 1 – sensitivity (1-SN) -> 1 - 0.9157 = 0.0843 which is another very low number.

Evaluating the Model using a new data set the Testing data shows that the model is performing very well. The Accuracy of the model was obtained at 0.9032 which is about 90%. The model also shows that the Area Under the Curve (AUC) is 0.9608 which is 96%.

The model was able to accurately predict the True Positive (TP) at 50 and True Negative (TN) at 34. The Specificity (SP) of the Model was calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR) and was 0.8095 (81%) while the Sensitivity (SN) of the model was calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall (REC) or true positive rate (TPR) and was 0.9803 (98%).

The Type I (FP) and Type II (FN) Errors were obtained as 8 and 1 respectively. The False positive rate (FPR) was calculated as the number of incorrect positive predictions divided by the total number of negatives which is 1 – specificity (1-SP)-> 1 - 0.8095 = 0.1905 which is quite low. False negative rate (FNR) is calculated as the number of incorrect negative predictions divided by the total number of positives which is 1 – sensitivity (1-SN) -> 1 - 0.9803 = 0.0197 which is very good for a classifier of this nature.

The Classification Error rate was obtained at 0.0968 which is good and less than 0.5 for this type of classifier and the nature of data provided. The precision for No Crime and Crime was obtained as 0.9714 and 0.8621 respectively. This gives an Average precision of 0.9168 (92%) which is another good metric to determine the performance of this Model.

The F1 Score or Measure was obtained for No Crime and Crime was obtained as 0.9714 and 0.8621 respectively. This gives an Average F1 Score of 0.9167 (92%) which is also quite good for this Model.

Using the Evaluation data provided, The Model was also able to predict new target with 22 incidence as “0” No Crime and 18 incidence as “1” Crime using the available features. These predictions corresponds to 55% and 45% respectively.