DATA 621 - Homework 3

Fall 2020 - Business Analytics and Data Mining

Mohamed Thasleem, Kalikul Zaman

11/01/2020

Contents

Introduction

In this homework assignment, you will 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).

1. Data Download

Librarires and download the classification output data set

```
library(tidyverse)
library(psych)
library(corrplot)
library(RColorBrewer)
library(knitr)
library(MASS)
library(caret)
library(kableExtra)
library(ResourceSelection)
library(pROC)
```

Variable Name	Short Description
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
lstat	lower status of the population (percent)
medv	median value of owner-occupied homes in $1000s
target	whether the crime rate is above the median crime rate (1) or not (0)

```
path <- "https://raw.githubusercontent.com/mohamedthasleem/DATA621/master/HW3"
crime_train <- read.csv(paste0(path,"/crime-training-data_modified.csv"))
crime_test <- read.csv(paste0(path,"/crime-evaluation-data_modified.csv"))
```

2. Data Exploration

The dataset contains 13 variables and 466 observations with no missing values. The variable chas, is a dummy variable and the rest are numerical variables. Finding the mean, standard deviation, skewness and other information for statistical analysis.

Based on the histogram plot below,the varialbe medv, and rm are normally distributed and bi-modal distribution of the variables indus, rad and tax.

The following plots show how predictors are distributed between a positive target variable (areas with crime rates higher than the median, i.e. blue) and a negative target variable (areas with crime rates below the median, i.e. red). What we are looking for is variables that show way to split data into two groups.

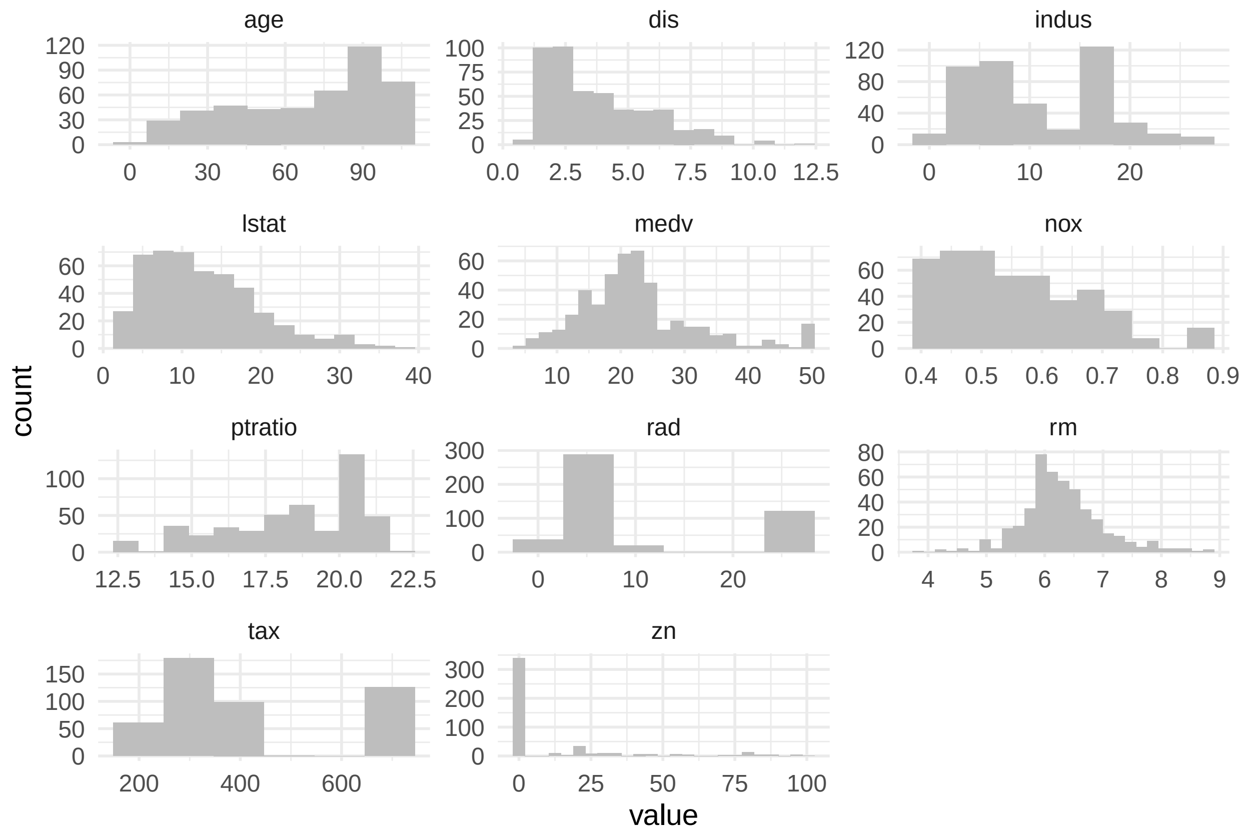
```
# Statistics
crime_train %>%
  mutate(chas = as.factor(chas), target = as.factor(target)) %>% glimpse() %>% describe()
```

```
## Rows: 466
## Columns: 13
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 100, 20...
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5.19, ...
## $ chas    <fct> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ nox     <dbl> 0.605, 0.871, 0.740, 0.428, 0.488, 0.520, 0.693, 0.693, 0.5...
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519, 6.3...
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38.1, 19...
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896, 1.6...
## $ rad     <int> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5, 5, 2...
## $ tax     <int> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330, 398,...
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, 16.4,...
```

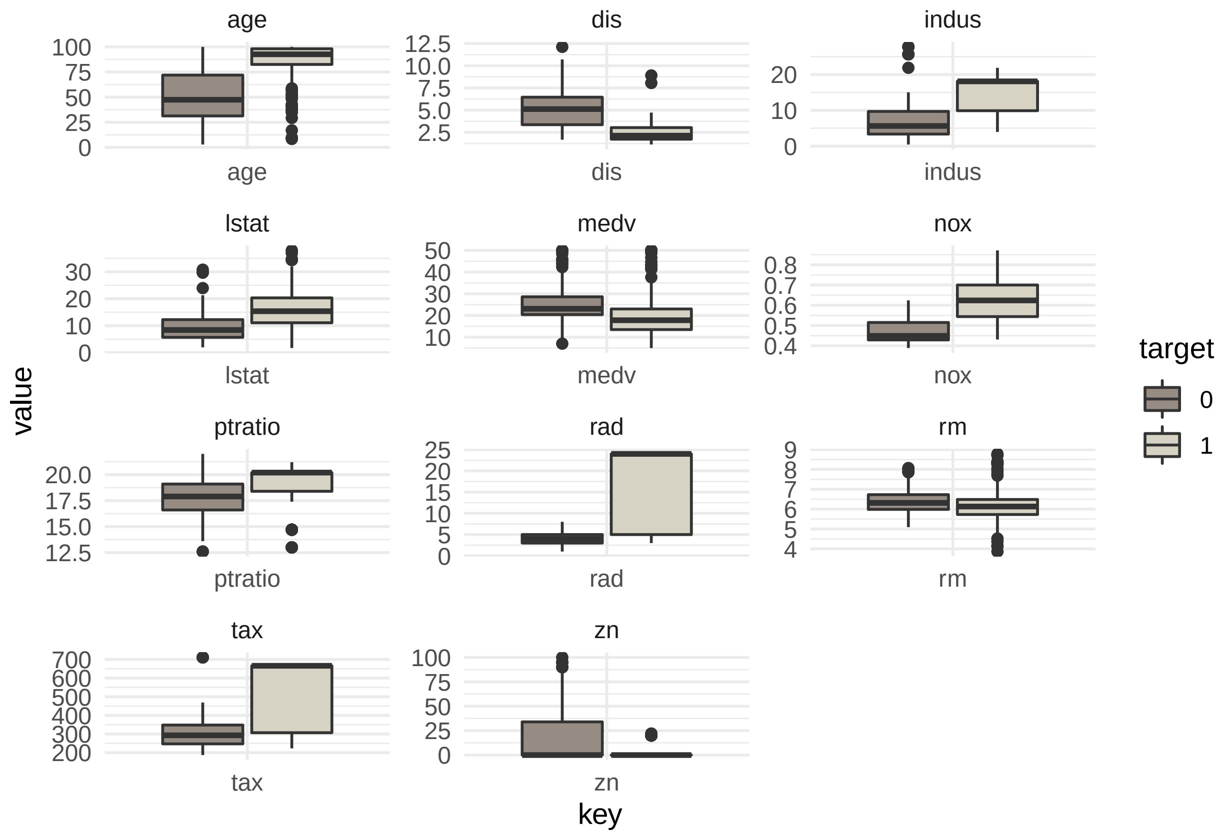
```
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5.68, 9...
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20.9, 2...
## $ target  <fct> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,...
```

```
##         vars   n   mean     sd median trimmed    mad    min    max  range  skew
## zn         1 466  11.58  23.36   0.00    5.35   0.00   0.00 100.00 100.00  2.18
## indus      2 466  11.11   6.85   9.69   10.91   9.34   0.46  27.74  27.28  0.29
## chas*      3 466   1.07   0.26   1.00    1.00   0.00   1.00   2.00   1.00  3.34
## nox        4 466   0.55   0.12   0.54    0.54   0.13   0.39   0.87   0.48  0.75
## rm         5 466   6.29   0.70   6.21    6.26   0.52   3.86   8.78   4.92  0.48
## age        6 466  68.37  28.32  77.15   70.96  30.02   2.90 100.00  97.10 -0.58
## dis        7 466   3.80   2.11   3.19    3.54   1.91   1.13  12.13  11.00  1.00
## rad        8 466   9.53   8.69   5.00    8.70   1.48   1.00  24.00  23.00  1.01
## tax        9 466 409.50 167.90 334.50  401.51 104.52 187.00 711.00 524.00  0.66
## ptratio   10 466  18.40   2.20  18.90   18.60   1.93  12.60  22.00   9.40 -0.75
## lstat     11 466  12.63   7.10  11.35   11.88   7.07   1.73  37.97  36.24  0.91
## medv      12 466  22.59   9.24  21.20   21.63   6.00   5.00  50.00  45.00  1.08
## target*   13 466   1.49   0.50   1.00    1.49   0.00   1.00   2.00   1.00  0.03
##         kurtosis   se
## zn          3.81 1.08
## indus      -1.24 0.32
## chas*       9.15 0.01
## nox        -0.04 0.01
## rm          1.54 0.03
## age        -1.01 1.31
## dis         0.47 0.10
## rad        -0.86 0.40
## tax        -1.15 7.78
## ptratio    -0.40 0.10
## lstat       0.50 0.33
## medv        1.37 0.43
## target*    -2.00 0.02
```

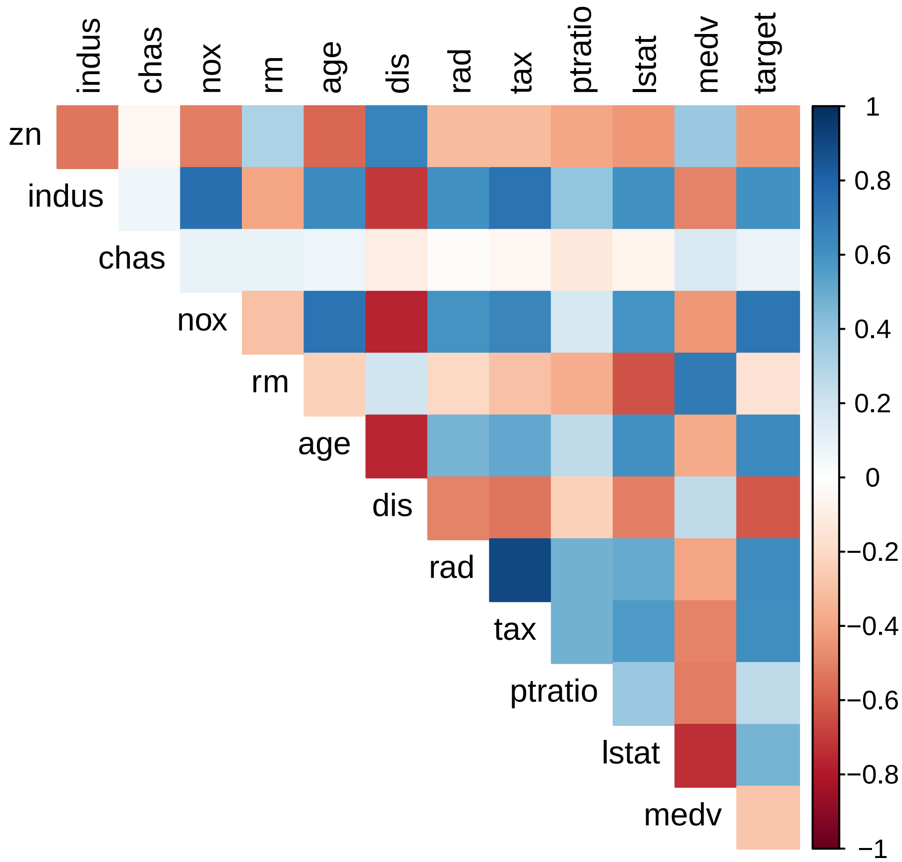
```r
# Distribution of the varaibles
crime_train %>%
  gather(key, value, -c(target, chas)) %>%
  ggplot(aes(value)) +
  geom_histogram(binwidth = function(x) 2 * IQR(x) / (length(x)^(1/3)), fill="gray") +
  facet_wrap(~ key, scales = 'free', ncol = 3) +
  theme_minimal()
```




```r
# Box-plot
crime_train %>%
  dplyr::select(-chas) %>%
  gather(key, value, -target) %>%
  mutate(key = factor(key),
         target = factor(target)) %>%
  ggplot(aes(x = key, y = value)) +
  geom_boxplot(aes(fill = target)) +
  facet_wrap(~ key, scales = 'free', ncol = 3) +
  scale_fill_manual(values=c("#968c83", "#d6d2c4")) +
  theme_minimal()
```




```r
crime_train %>%
  cor(.) %>%
  corrplot(., method = "color", type = "upper", tl.col = "black", diag = FALSE)
```




```r
# Top correlation
kable(sort(cor(dplyr::select(crime_train, target, everything()))[,1], decreasing = T),
      col.names = c("Correlation")) %>% kable_styling(full_width = F)
```

	Correlation
target	1.0000000
nox	0.7261062
age	0.6301062
rad	0.6281049
tax	0.6111133
indus	0.6048507
lstat	0.4691270
ptratio	0.2508489
chas	0.0800419
rm	-0.1525533
medv	-0.2705507
zn	-0.4316818
dis	-0.6186731

3. Data Preparation

The variance inflation factor which quantifies the extent of correlation between one predictor and the other predictors in a model. Some of the variables are skewed, have outliers or follow a bi-modal distribution.

By doing this analysism, we can remove the high scrore values also we transform some of the variables to account for its variances with respect to target variable.

```
# Multicollinear variables
kable((car::vif(glm(target ~. , data = crime_train))), col.names = c("VIF Score")) %>%
  kable_styling(full_width = F)
```

	VIF Score
zn	2.324259
indus	4.120699
chas	1.090265
nox	4.505049
rm	2.354788
age	3.134015
dis	4.240618
rad	6.781354
tax	9.217228
ptratio	2.013109
lstat	3.649059
medv	3.667370

```
# Transoformation of the variables.
crime_train_trans <- crime_train %>%
  dplyr::select(-tax) %>%
  mutate(age = log(age), lstat = log(lstat), zn = zn^2, rad = rad^2, nox = I(nox^2))
```

4. Build Models

Three different models were built to see the best performance

Model 1 - All Varaibles

Model 2 - with transformed variables

Model 3 - Stepwise Selection variables

Model 1

We use all original variables. Out of 7 1n 12 variables have statistically significant p-values. In the goodness-of-fit test, the null hypothesis is rejected due to low p-value.

```
#model 1 with all original variables
model1 <- glm(target ~ ., family = "binomial", crime_train)
summary(model1)
```

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = crime_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8464  -0.1445  -0.0017   0.0029   3.4665
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas          0.910765   0.755546   1.205  0.22803
## nox          49.122297   7.931706   6.193 5.90e-10 ***
## rm           -0.587488   0.722847  -0.813  0.41637
## age           0.034189   0.013814   2.475  0.01333 *
## dis           0.738660   0.230275   3.208  0.00134 **
## rad           0.666366   0.163152   4.084 4.42e-05 ***
## tax          -0.006171   0.002955  -2.089  0.03674 *
## ptratio       0.402566   0.126627   3.179  0.00148 **
## lstat         0.045869   0.054049   0.849  0.39608
## medv          0.180824   0.068294   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

```r
#fit test
hoslem.test(crime_train$target, fitted(model1))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  crime_train$target, fitted(model1)
## X-squared = 17.741, df = 8, p-value = 0.02326
```

Model 2

We will use our transformed variables, but same results as of Model 1. Moreover, the p-value is low again thus this model's goodness of fit null hypothesis is rejected as well.

Since the transformed variables yielded a model that performs worse than the model with original variables, we will apply a box-cox transformation to all the variables to see if it performs better. As seen previously, most of our dataset has many skewed variables. When an attribute has a normal distribution but is shifted, this is called a skew. The distribution of an attribute can be shifted to reduce the skew and make it more normal The Box Cox transform can perform this operation (assumes all values are positive). Even though this model took less Fisher Scoring iterations than other models, it too yielded similar results and low p-value as the other two models.

```r
#model 2 with transformed variables.
model2 <- glm(target ~ ., family = "binomial", crime_train_trans)
summary(model2)
```

```
##
## Call:
```

```
## glm(formula = target ~ ., family = "binomial", data = crime_train_trans)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0433  -0.2233   0.0000   0.0000   3.2207
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -28.050185   5.743773  -4.884 1.04e-06 ***
## zn           -0.003025   0.001538  -1.966 0.049279 *
## indus        -0.111100   0.045632  -2.435 0.014904 *
## chas          1.341454   0.722182   1.858 0.063240 .
## nox          44.473984   7.183977   6.191 5.99e-10 ***
## rm           -0.356816   0.652810  -0.547 0.584664
## age           0.896847   0.639678   1.402 0.160907
## dis           0.661694   0.207061   3.196 0.001395 **
## rad           0.048170   0.012613   3.819 0.000134 ***
## ptratio       0.342741   0.111882   3.063 0.002188 **
## lstat         0.435860   0.649096   0.671 0.501911
## medv          0.149548   0.059824   2.500 0.012425 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 203.77  on 454  degrees of freedom
## AIC: 227.77
##
## Number of Fisher Scoring iterations: 10
```

```r
#fit test
hoslem.test(crime_train$target, fitted(model1))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  crime_train$target, fitted(model1)
## X-squared = 17.741, df = 8, p-value = 0.02326
```

```r
# boxcox transformation use caret package
crime_boxcox <- preProcess(crime_train, c("BoxCox"))
cb_transformed <- predict(crime_boxcox, crime_train)
model <- glm(target ~ ., family = "binomial", cb_transformed)
summary(model)
```

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = cb_transformed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9381  -0.1116  -0.0010   0.1137   3.4325
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  21.134102  38.495758   0.549 0.583007
## zn           -0.022244   0.026852  -0.828 0.407433
## indus        -0.002008   0.216566  -0.009 0.992603
## chas          0.945998   0.761805   1.242 0.214316
## nox          14.172248   2.240335   6.326 2.52e-10 ***
## rm           -2.330063   2.813401  -0.828 0.407556
## age           0.012105   0.003914   3.093 0.001984 **
## dis           3.390172   0.868215   3.905 9.43e-05 ***
## rad           3.152839   0.733173   4.300 1.71e-05 ***
## tax         -16.176693  20.445106  -0.791 0.428812
## ptratio       0.025318   0.007169   3.532 0.000413 ***
## lstat        -0.051425   0.445840  -0.115 0.908173
## medv          2.461332   0.856713   2.873 0.004066 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 196.79  on 453  degrees of freedom
## AIC: 222.79
##
## Number of Fisher Scoring iterations: 8
```

```r
#fit model
hoslem.test(crime_train$target, fitted(model))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  crime_train$target, fitted(model)
## X-squared = 31.722, df = 8, p-value = 0.0001045
```

Model 3

Finally for our third model, we will use the stepwise selection from the MASS package. This model yields the best performance so far. It has the lowest AIC Score and all of the variables have significant p-value. As such we will select this model to make prediction

```r
# model 3 stepwise selection of variables
model3 <- stepAIC(model1, direction = "both", trace = FALSE)
summary(model3)
```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv, family = "binomial", data = crime_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## zn           -0.068648   0.032019  -2.144  0.03203 *
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## age           0.032950   0.010951   3.009  0.00262 **
## dis           0.654896   0.214050   3.060  0.00222 **
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## medv          0.110472   0.035445   3.117  0.00183 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

```r
# goodness of fit test
hoslem.test(crime_train$target, fitted(model3))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  crime_train$target, fitted(model3)
## X-squared = 11.714, df = 8, p-value = 0.1644
```

```r
# comparing all models using different measures
c1 <- confusionMatrix(as.factor(as.integer(fitted(model1) > .5)),
                      as.factor(model1$y), positive = "1")
c2 <- confusionMatrix(as.factor(as.integer(fitted(model2) > .5)),
                      as.factor(model2$y), positive = "1")
c3 <- confusionMatrix(as.factor(as.integer(fitted(model3) > .5)),
                      as.factor(model3$y), positive = "1")

roc1 <- roc(crime_train$target,  predict(model1, crime_train,
                                         interval = "prediction"))
roc2 <- roc(crime_train$target,  predict(model2, crime_train,
                                         interval = "prediction"))
roc3 <- roc(crime_train$target,  predict(model3, crime_train,
                                         interval = "prediction"))
```

5. Select Models

We have compared various metrics for all three models. We calculate all three models' accuracy, classification error rate, precision, sensitivity, specificity, F1 score, AUC, and confusion matrix. Even though model 1 performs better in every metrics, the difference is very small. We will pick model 3 with stepwise variable selection because it has the lowest AIC score and all variables have high p-values

Model 3 Scrores best in AIC Score

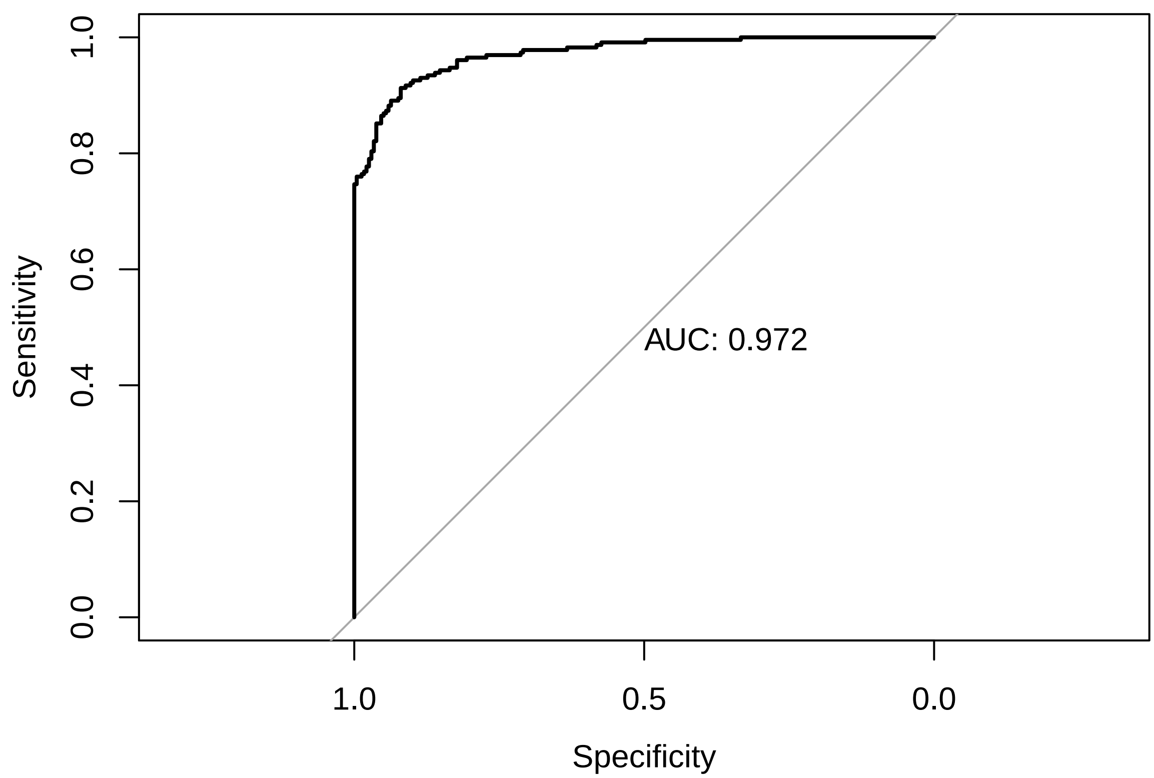
```r
## 4. Select Model

metrics1 <- c(c1$overall[1], "Class. Error Rate" = 1 - as.numeric(c1$overall[1]),
              c1$byClass[c(1, 2, 5, 7)], AUC = roc1$auc)
metrics2 <- c(c2$overall[1], "Class. Error Rate" = 1 - as.numeric(c2$overall[1]),
              c2$byClass[c(1, 2, 5, 7)], AUC = roc2$auc)
metrics3 <- c(c3$overall[1], "Class. Error Rate" = 1 - as.numeric(c3$overall[1]),
              c3$byClass[c(1, 2, 5, 7)], AUC = roc3$auc)

kable(cbind(metrics1, metrics2, metrics3), col.names = c("Model 1", "Model 2", "Model 3")) %>%
  kable_styling(full_width = T)
```

	Model 1	Model 2	Model 3
Accuracy	0.9163090	0.9055794	0.9120172
Class. Error Rate	0.0836910	0.0944206	0.0879828
Sensitivity	0.9039301	0.8733624	0.9039301
Specificity	0.9282700	0.9367089	0.9198312
Precision	0.9241071	0.9302326	0.9159292
F1	0.9139073	0.9009009	0.9098901
AUC	0.9737623	0.8977576	0.9719382

```r
# plotting roc curve of model 3
plot(roc(crime_train$target,  predict(model3, crime_train, interval = "prediction")),
     print.auc = TRUE)
```




```r
# prepare evaualtion dataset
crime_test <- crime_test %>%
  mutate(chas = as.factor(chas))

# prediction
predict <- predict(model3, crime_test, interval = "prediction")
eval <- table(as.integer(predict > .5))
eval
```

```
##
##  0  1
## 21 19
```

Model 3 (stepwise) Scrores best in AIC Score considered as a best model.