DATA621 Homework3 - Crime Ratings

Jonathan Hernandez

June 20, 2018

Introduction

- This assignment looks at crime information on various neighborhood for a major city. There are several variables and the goal is to come up with a binary logistic regression model that can predict whether a particular neighborhood will be at risk for high crime levels. I will examine a few models and use the best criteria based on statistics such as ROC curves/AUC (Area under the curve) values and confusion matricies for each model. The model with the best performance and accuracy will be considered. Finally, the model will be evaluated using the training data and then I will make predictions based using the evaluation data set.

Data Preparation

- Let's first load the dataset and examine the structure and summary of the dataset
- Summary:

```
##        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
```

- Structure (number of rows and columns, variable type such as int, factor etc):

```
## 'data.frame':    466 obs. of  14 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 ...
##  $ black  : num  369 397 387 375 394 ...
##  $ 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 ...
```

- Transform tax and rad variables numberic instead of integer for computation

such as computing correlation between attributes. Later, we wil change the target and chas attributes to be factors instead of integers so they can be labeled properly.

- The summary of the dataset shows that there are no missing values and besides

the response variable target and the explanatory variable chas, all others are of the numeric type.

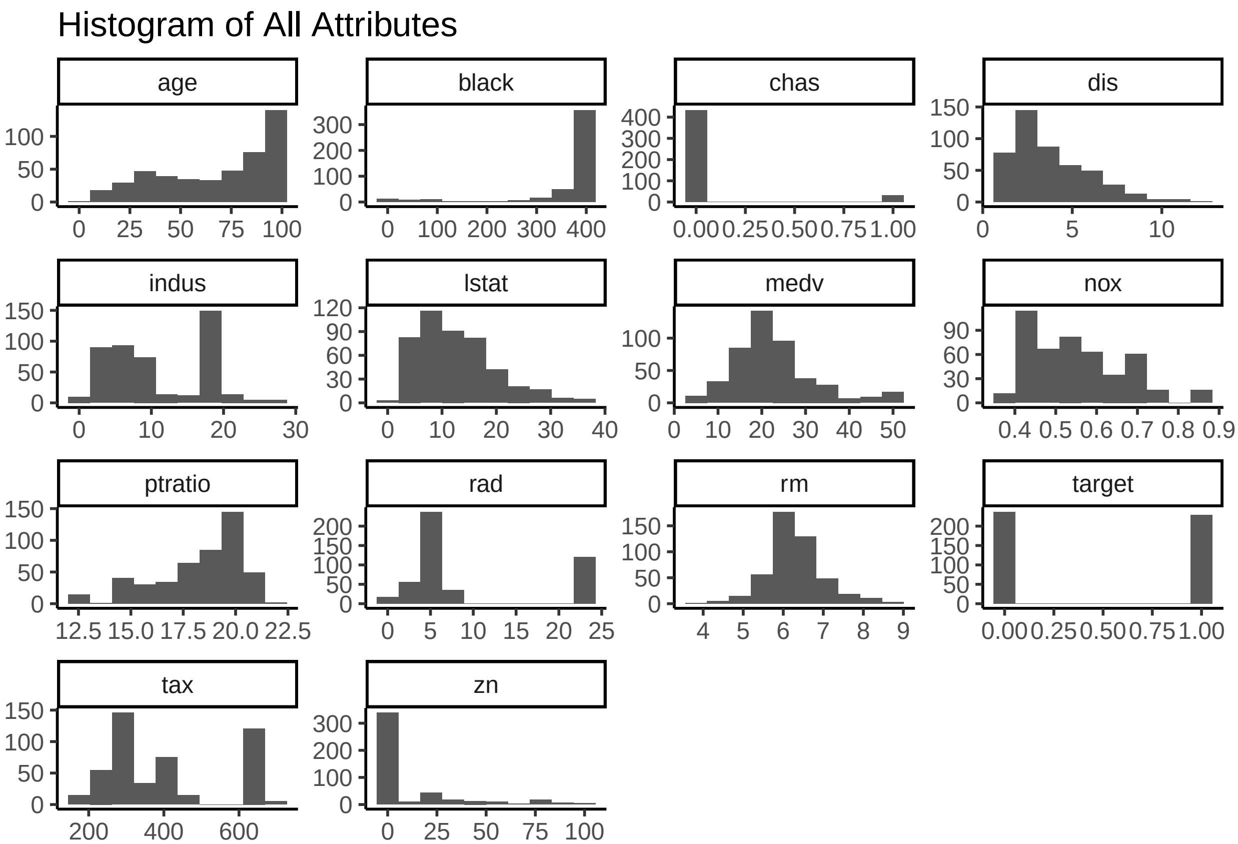
Data Exploration

- Let us make some plots like histograms and see how each variable behaves. Also,

I will plot some of the variables for each crime rate below and above the median crime rate.

- Histograms of each variable

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Loading required package: tidyr
```



- There appears to be serveral outliers in most of the variables but we will keep them for

now and use them in our models.

- Also let us see how well our data is correlated along with one another. I will

create a correlation table to see the correlation coefficients against each pair

of attributes. This will be useful for one of the models to be built later on.

```
##                 zn      indus        nox         rm        age        dis
## zn       1.0000000 -0.5382664 -0.5170452  0.3198141 -0.5725805  0.6601243
## indus   -0.5382664  1.0000000  0.7596301 -0.3927118  0.6395818 -0.7036189
## nox     -0.5170452  0.7596301  1.0000000 -0.2954897  0.7351278 -0.7688840
## rm       0.3198141 -0.3927118 -0.2954897  1.0000000 -0.2328125  0.1990158
## age     -0.5725805  0.6395818  0.7351278 -0.2328125  1.0000000 -0.7508976
## dis      0.6601243 -0.7036189 -0.7688840  0.1990158 -0.7508976  1.0000000
## rad     -0.3154812  0.6006284  0.5958298 -0.2084457  0.4603143 -0.4949919
## tax     -0.3192841  0.7322292  0.6538780 -0.2969343  0.5121245 -0.5342546
## ptratio -0.3910357  0.3946898  0.1762687 -0.3603471  0.2554479 -0.2333394
## black    0.1794150 -0.3581356 -0.3801549  0.1326676 -0.2734677  0.2938441
## lstat   -0.4329925  0.6071102  0.5962426 -0.6320245  0.6056200 -0.5075280
## medv     0.3767171 -0.4961743 -0.4301227  0.7053368 -0.3781560  0.2566948
##                rad        tax    ptratio      black      lstat       medv
## zn      -0.3154812 -0.3192841 -0.3910357  0.1794150 -0.4329925  0.3767171
## indus    0.6006284  0.7322292  0.3946898 -0.3581356  0.6071102 -0.4961743
## nox      0.5958298  0.6538780  0.1762687 -0.3801549  0.5962426 -0.4301227
## rm      -0.2084457 -0.2969343 -0.3603471  0.1326676 -0.6320245  0.7053368
## age      0.4603143  0.5121245  0.2554479 -0.2734677  0.6056200 -0.3781560
## dis     -0.4949919 -0.5342546 -0.2333394  0.2938441 -0.5075280  0.2566948
```

```
## rad      1.0000000  0.9064632  0.4714516 -0.4463750  0.5031013 -0.3976683
## tax      0.9064632  1.0000000  0.4744223 -0.4425059  0.5641886 -0.4900329
## ptratio  0.4714516  0.4744223  1.0000000 -0.1816395  0.3773560 -0.5159153
## black   -0.4463750 -0.4425059 -0.1816395  1.0000000 -0.3533659  0.3300286
## lstat    0.5031013  0.5641886  0.3773560 -0.3533659  1.0000000 -0.7358008
## medv    -0.3976683 -0.4900329 -0.5159153  0.3300286 -0.7358008  1.0000000
```

Build Models

- I will use the training data to build at least 3 different binary logistic regression

models using different models and techniques.

- Let's first with the trivial easiest model; include all variables to predict

if a city will be above or below the crime rate. No variable elimination is done.

Model 1

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + black + lstat + medv, family = "binomial",
##     data = crime)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2854  -0.1372  -0.0017   0.0020   3.4721
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -36.839521   7.028726  -5.241 1.59e-07 ***
## zn           -0.061720   0.034410  -1.794 0.072868 .
## indus        -0.072580   0.048546  -1.495 0.134894
## chas1         1.032352   0.759627   1.359 0.174139
## nox          50.159513   8.049503   6.231 4.62e-10 ***
## rm           -0.692145   0.741431  -0.934 0.350548
## age           0.034522   0.013883   2.487 0.012895 *
## dis           0.765795   0.234407   3.267 0.001087 **
## rad           0.663015   0.165135   4.015 5.94e-05 ***
## tax          -0.006593   0.003064  -2.152 0.031422 *
## ptratio       0.442217   0.132234   3.344 0.000825 ***
## black        -0.013094   0.006680  -1.960 0.049974 *
## lstat         0.047571   0.054508   0.873 0.382802
## medv          0.199734   0.071022   2.812 0.004919 **
## ---
## 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: 186.15  on 452  degrees of freedom
## AIC: 214.15
##
```

```
## Number of Fisher Scoring iterations: 9
```

Model 2

- The second model is done using backward elimination; from the above this eliminates the following variables due to having high p-values. The variables selected are the ones that have low p-values below 0.05 are significant.

- indus
- chas
- rm
- lstat

```
##
## Call:
## glm(formula = target ~ nox + age + dis + rad + tax + ptratio +
##     medv, family = "binomial", data = crime)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.01059  -0.19744  -0.01371   0.00402   3.06424
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -36.824228   5.858405  -6.286 3.26e-10 ***
## nox          42.338378   6.639207   6.377 1.81e-10 ***
## age           0.031882   0.010693   2.982 0.002867 **
## dis           0.429555   0.171849   2.500 0.012433 *
## rad           0.701767   0.139426   5.033 4.82e-07 ***
## tax          -0.008237   0.002534  -3.250 0.001153 **
## ptratio       0.376575   0.108912   3.458 0.000545 ***
## medv          0.093653   0.033556   2.791 0.005255 **
## ---
## 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: 203.45  on 458  degrees of freedom
## AIC: 219.45
##
## Number of Fisher Scoring iterations: 9
```

Model 3

- The third model is to use the idea of correlation to filter out variables with high correlation using a cutoff point of 0.75. By using this, we can see if we can get a binary classification model that can have high accuracy of calculating if a particular neighborhood will be targeted for a high crime rate or not. Variables that are highly correlated

```
## Loading required package: caret

## Loading required package: lattice

## [1] "Variables to be used:"

##  [1] "zn"      "nox"     "age"     "dis"     "tax"     "ptratio" "black"
##  [8] "lstat"   "medv"    "target"

##
## Call:
## glm(formula = target ~ zn + nox + age + dis + tax + ptratio +
##     black + lstat + medv, family = "binomial", data = crime_subset)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.26039  -0.31877  -0.01099   0.22687   3.08518
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -29.745688   5.154374  -5.771 7.88e-09 ***
## zn           -0.068459   0.029039  -2.357   0.0184 *
## nox          36.897169   5.473339   6.741 1.57e-11 ***
## age           0.021283   0.009663   2.203   0.0276 *
## dis           0.776376   0.190603   4.073 4.64e-05 ***
## tax           0.002733   0.001515   1.804   0.0713 .
## ptratio       0.228694   0.101724   2.248   0.0246 *
## black        -0.011202   0.005160  -2.171   0.0299 *
## lstat         0.050236   0.042441   1.184   0.2365
## medv          0.177577   0.036889   4.814 1.48e-06 ***
## ---
## 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: 240.50  on 456  degrees of freedom
## AIC: 260.5
##
## Number of Fisher Scoring iterations: 8
```

Model 4

- We will use the glmnet package to do feature selection on the crime dataset

and retrieve the variables most relevant to predict the response variable target.

I'm also using the concept of LASSO regularization that finds good regularization

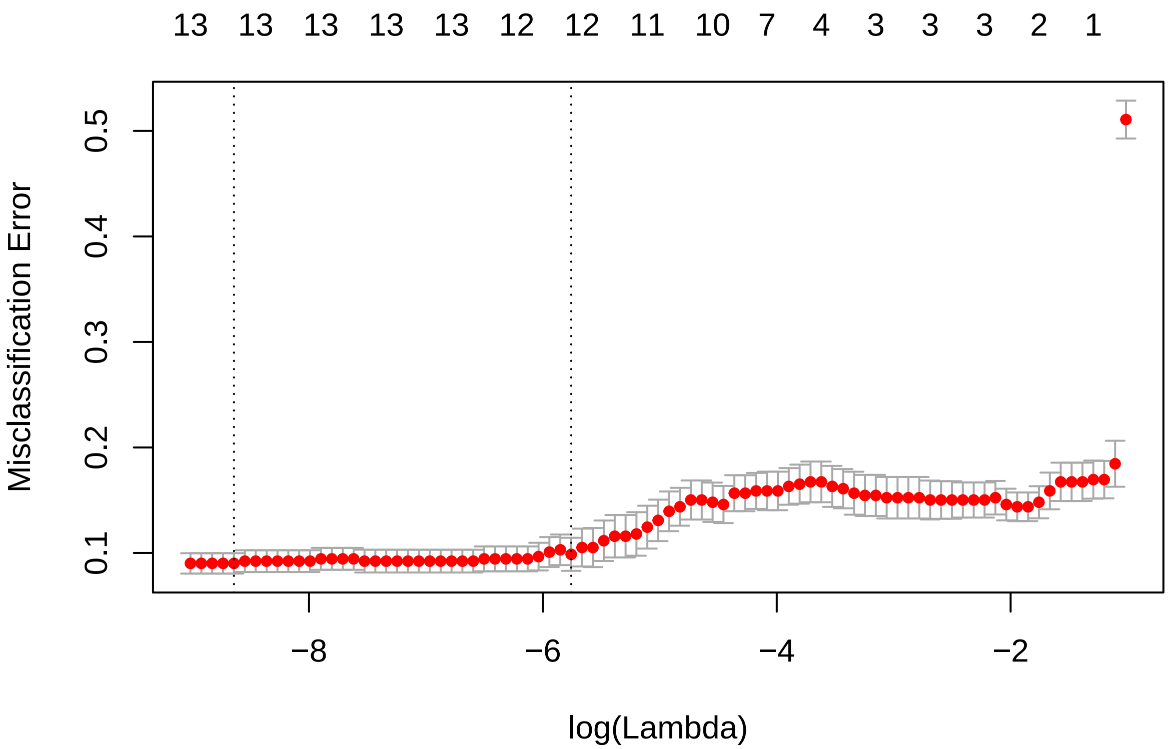
coefficent to prevent overfitting and puts a constraint on the sum of the feature values.

Adding an extra term in the logistic regression (or linear regression) is common

and usually involves adding a coefficent lambda λ the higher the λ

value is, the more bias but less overfitting.

```
## Loading required package: glmnet
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
## Loading required package: foreach
## Loaded glmnet 2.0-16
```




```
##        s0
## -25.35204
##            zn        indus         chas          nox           rm
## -0.028211488 -0.049269882  0.843264685 32.984904774  0.000000000
##           age          dis          rad          tax      ptratio
##  0.019369868  0.338561225  0.353187890 -0.003190429  0.232242728
##         black        lstat         medv
## -0.007501370  0.021490523  0.085996066
## [1] -871.6335
```

Summary of the built models

- Starting with Model 1 that is to include all variables based on

the coefficents is showing that the variable nox (nitrogen oxide concentration) has the most influence out of the other variables and says that for every concentration, all constants held constant, the probability will greatly increase and will be more likley for that town to have a high crime level.

- The second model which uses backward elimination has coefficents that have good significant p-values and has a higher AIC value. Like the first model, the coefficent for nox is high and seems to be the main variable that holds the most weight when computing the probability of a city having a high crime level. The only coefficent that is negative is the tax variable which means for all other variables held constant and for every 1% increase in tax rate, the probability of that city being a high crime level city decreases by a factory of 0.0082.

- The third model uses correlation matricies and the correlation threshold of 0.75 and uses the 2nd least variables. The AIC value is higher than the first two models. Coefficents are similar and took less time to converge to a solution (see the Fisher Scoring iterations in the summary of model 3).

- Model 4 the final model uses the concept of LASSO (Least absolute shrinkage and selection operator) to select variables and prevent the model from overfitting the data using the concept of regularization to penalize variables that would overfit the data. Variable coefficents are the similar but the rm (average number of rooms per dwelling) is 0 and is the only variable not included in the model. The AIC is low as well which makes it a good candidate as well for picking a good model.

Selecting Model

- After looking at each model, I decided to go with the one with the lowest AIC model as that rubric is used for doing feature selection and an appropriate model. I have selected model 4 (model using LASSO and regularization) as the model of choice. The reason for this is the low AIC value it has (running it shows it is negative) while the other models are positive AIC values. Also using the glmnet function with regularization is good as it helps the model to prevent overfitting.

- Confusion Matrix along with accuracy, specificity, sensitivity

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 227  25
##          1  10 204
##
##                Accuracy : 0.9249
##                  95% CI : (0.8971, 0.9471)
##     No Information Rate : 0.5086
```

```
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8496
##  Mcnemar's Test P-Value : 0.01796
##
##             Sensitivity : 0.8908
##             Specificity : 0.9578
##          Pos Pred Value : 0.9533
##          Neg Pred Value : 0.9008
##              Prevalence : 0.4914
##          Detection Rate : 0.4378
##    Detection Prevalence : 0.4592
##       Balanced Accuracy : 0.9243
##
##        'Positive' Class : 1
##
```

- F1 Score:

```
## [1] 0.9631327
```

- Classification error rate

```
## Classification error rate
##                 0.0751073
```

- Precision

```
## [1] 0.9007937
```

- Predict using the evaluation dataset

```
## Loading required package: pROC
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:glmnet':
##
##     auc
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
##    probabilities aboveCrimeLevel
## 1   0.0401878911              no
## 2   0.3510332967              no
## 3   0.3982404496              no
## 4   0.5568010397             yes
## 5   0.0630655654              no
## 6   0.0981518066              no
## 7   0.1294816917              no
## 8   0.0159707340              no
## 9   0.0090160269              no
## 10  0.0058600350              no
## 11  0.2467498620              no
## 12  0.2276392028              no
```

```
## 13  0.5610273932            yes
## 14  0.6608200358            yes
## 15  0.3813790206             no
## 16  0.1586556637             no
## 17  0.1608045508             no
## 18  0.7208701740            yes
## 19  0.0349827415             no
## 20  0.0001243210             no
## 21  0.0003177122             no
## 22  0.0554909185             no
## 23  0.0987197202             no
## 24  0.0852855416             no
## 25  0.0740784658             no
## 26  0.2756342311             no
## 27  0.0020482605             no
## 28  0.9999747906            yes
## 29  0.9999618540            yes
## 30  0.9994042555            yes
## 31  0.9999961338            yes
## 32  0.9999648979            yes
## 33  0.9999577259            yes
## 34  0.9999794856            yes
## 35  0.9999883091            yes
## 36  0.9999940835            yes
## 37  0.9999445249            yes
## 38  0.9995991342            yes
## 39  0.5109365951            yes
## 40  0.2720051271             no
```

- Appendix of R code

```
crime <- read.csv("crime-training-data.csv")
summary(crime)
str(crime)
# change to rad and tax to numerics
crime$tax <- as.numeric(crime$tax)
crime$rad <- as.numeric(crime$rad)
if (!require(dplyr)) install.packages("dplyr", dependencies = TRUE)
if (!require(ggplot2)) install.packages("ggplot2", dependencies = TRUE)
if (!require(tidyr)) install.packages("tidyr", dependencies = TRUE)

# do a histogram plot for each attribute, remove labels and have title only
ggplot(gather(crime), aes(x=value)) +
  geom_histogram(bins = 10) +
  facet_wrap(~ key, scales = "free") +
  theme_bw() +
  theme_classic() +
  ggtitle("Histogram of All Attributes") +
  theme(axis.title = element_blank())
# create correlation matrix for the crime dataset; will be used to filter columns
crime$chas <- factor(crime$chas)
crime$target <- factor(crime$target)

crime_correlation_matrix <- cor(crime[,c(1:2, 4:13)])
crime_correlation_matrix
```

```r
lm1_crime <- glm(target ~ zn + indus + chas +
                          nox + rm + age + dis + rad + tax +
                          ptratio + black + lstat + medv,
                 data = crime, family = "binomial")
summary(lm1_crime)
lm2_crime <- glm(target ~ nox + age + dis + rad + tax + ptratio + medv,
                 data = crime, family = "binomial")
summary(lm2_crime)
# load caret package
if (!require(caret)) install.packages("caret", dependencies = TRUE)

# find columns that have absolute high correlation above 0.75 as a cutoff,
# use R's caret package and the function findCorrelation
well_correlated <- findCorrelation(crime_correlation_matrix, cutoff = 0.75)

# create a crime subset that contains the variables that are not well correlated
crime_subset <- crime %>% select(-well_correlated, target)

# print variables to be used
print("Variables to be used:")
names(crime_subset)

# make the binary model out of those variables
lm3_model <- glm(target ~ zn + nox + age + dis + tax + ptratio + black + lstat +
                   medv, data = crime_subset, family = "binomial")
summary(lm3_model)
if (!require(glmnet)) install.packages("glmnet", dependencies = TRUE)
library(glmnet)

# use the cv.glmnet to do cross validation on the data to find the optimal lambda
# value that has the lowest missclassification rate in predicting target
lm4_model <- cv.glmnet(x=data.matrix(crime[,1:13]), y=crime$target,
                    family = "binomial", alpha = 1, nlambda = 100,
                    type.measure = "class")
# plot the model and show lambda values that provide minimal missclassification rate
plot(lm4_model)

# fit the model based on picking a model that has minimal lambda (penalty)
# use the lambda that is 1 standard error from the min value so features may be
# removed and less variables
fit <- glmnet(x=data.matrix(crime[,1:13]), y=crime$target, family = "binomial",
              alpha = 1, lambda = lm4_model$lambda.1se)

# output coefficents of the fitted model and which variables we will select
fit$a0
fit$beta[,1]
# compute AIC when using the GLM package for fitting a model with regression
# AIC computation is below:
k <- fit$df # number of features, degrees of freedom
loglikhood <- fit$nulldev - deviance(fit) # log-likihood
AICc <- -2*loglikhood + 2*k
AICc
# predict on the training data to compute stats
```

```r
predict_train <- predict(fit, type = "class", newx = data.matrix(crime[,1:13]),
                         s = "lambda.1se")
conf_matrix <- confusionMatrix(factor(predict_train), crime$target, positive = "1")
conf_matrix
conf_matrix_table <- conf_matrix$table
f1 <- (2 * precision(conf_matrix_table) * sensitivity(conf_matrix_table)) /
  (precision(conf_matrix_table) + specificity(conf_matrix_table))
f1
error_rate <- 1 - conf_matrix$overall['Accuracy'] # extract accuracy
names(error_rate) = "Classification error rate"
error_rate
precision(conf_matrix_table)
if(!require(pROC)) install.packages("pROC", dependencies = TRUE)
library(pROC)

# evaluation dataset to test out model
crime_eval <- read.csv("crime-evaluation-data.csv")
# probabilities
probabilities <- predict(fit, type = "response",
                         newx = data.matrix(crime_eval[,1:13]))

# classification with threshold > 0.5 neighborhood is in high crime level, not
# otherwise
above_med_crime_rate <- predict(fit, type = "class",
                                newx = data.matrix(crime_eval[,1:13]))
above_med_crime_rate <- ifelse(above_med_crime_rate >= 0.5, "yes", "no")

predictions <- data.frame(probabilities,above_med_crime_rate)
colnames(predictions) <- c("probabilities", "aboveCrimeLevel")
predictions
```