CRIME RATE PREDICTOR

Binary Logistic Regression Model

Rafal Decowski

CUNY | DATA MINING

Objective

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.

Contents

DATA EXPLORATION

Dataset

The data set contains information on crime for various neighborhoods of a major city. It has 466 cases across 12 predictor variables and one response variable. All variables are numerical (or bool) and all cases are complete (no missing values.) The dataset contains several *proportions* or *ratios* as units. These composite variables are engineered and not in their raw format, but the unit is somewhat standardized.

VARAIBLE	DESCRIPTION
ZN	Proportion of residential land zoned for large lots
INDUS	Proportion of non-retail business acres per suburb
CHAS	A dummy var. for whether the suburb borders the Charles River
NOX	Nitrogen oxides concentration
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
MEDV	Median value of owner-occupied homes in $1000s
TARGET	Whether the crime rate is above the median crime rate

Descriptive Statistics

Descriptive statistics help us identify variations, ranges, distributions, missing values and more with a simple summary table. This will later help us drive decisions on transformations, normalizations and general data cleansing. The table below tells me that there are no missing values but there seem to be some outliers due to a significant mean and median differences. It also highlights that the dataset contains almost the same number of high and low crime rate cases.

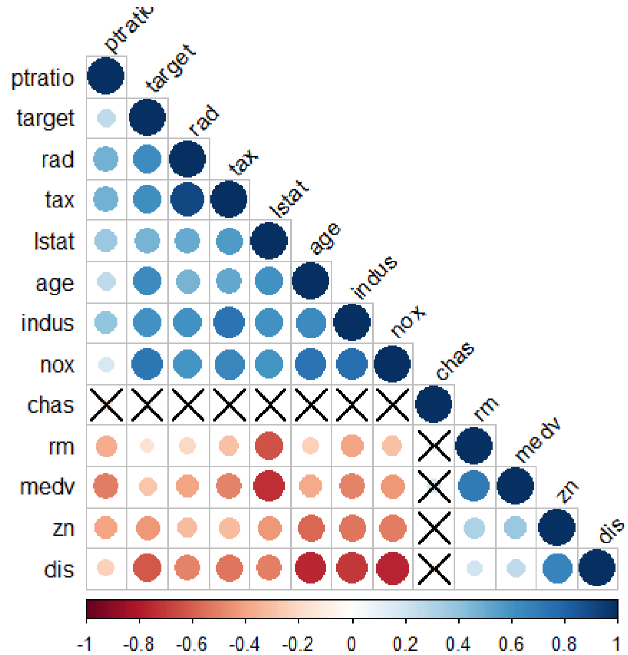
	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	LSTAT	MEDV	TARGET
MIN.	0	0.46	0	0.39	3.86	2.9	1.13	1	187	12.6	1.73	5	0
1ST QU.	0	5.15	0	0.45	5.89	43.88	2.1	4	281	16.9	7.04	17.02	0
MEDIAN	0	9.69	0	0.54	6.21	77.15	3.19	5	334.5	18.9	11.35	21.2	0
MEAN	11.58	11.11	0.07	0.55	6.29	68.37	3.8	9.53	409.5	18.4	12.63	22.59	0.49
3RD QU.	16.25	18.1	0	0.62	6.63	94.1	5.21	24	666	20.2	16.93	25	1
MAX.	100	27.74	1	0.87	8.78	100	12.13	24	711	22	37.97	50	1
SD	23.36	6.85	0.26	0.12	0.7	28.32	2.11	8.69	167.9	2.2	7.1	9.24	0.5

Correlation

The correlation helps us highlight predictor variables that have a strong relationship with the target variable. It helps us narrow down the important ones and discard the ones that do not significantly affect the prediction results.

VARIABLE	CORRELATION
NOX	0.73
AGE	0.63
RAD	0.63
TAX	0.61
INDUS	0.6
LSTAT	0.47
PTRATIO	0.25
CHAS	0.08
RM	-0.15
MEDV	-0.27
ZN	-0.43
DIS	-0.62

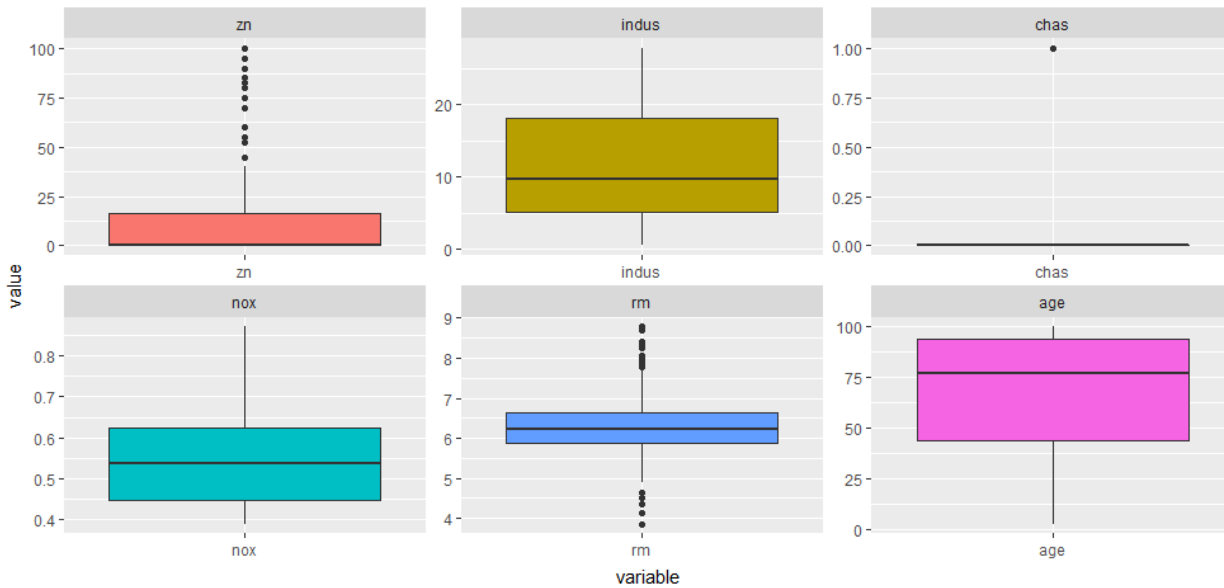
The image below shows positive (blue) and negative(red) correlation between all variables. The crossed-out fields are rejected by a 95% confidence level.
Domain knowledge makes this chart more significant as it helps form more advanced hypotheses and see how variables are related.

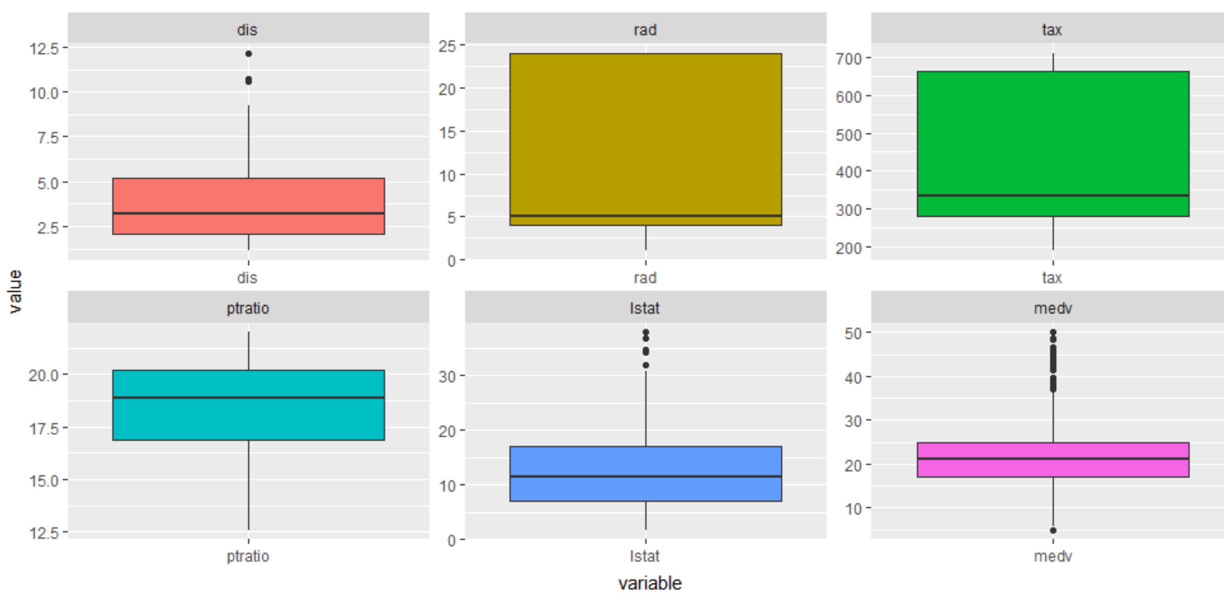


Boxplots

The boxplots below help us bring the descriptive statistics from the previous section into neat visuals. We can easily determine ranges, medians and outliers. Variables with a high number of outliers may need additional cleansing and transformations which may help with improving accuracy of models.

It seems there aren't too many outliers and they are only visible for some variables. It suggests that handling them may not bring too much value for this dataset.





DATA PREPARATION

The crimes data, even though it is small, all cases are complete, and the variables are of numerical nature (some are categorical but represented as 1's and 0s.) Since there were no missing values it made data handling much easier which usually translates into a greater accuracy of models.

At this point domain knowledge often is the most powerful as it helps with deriving new features, grouping or partitioning existing features into more informative categories or 'buckets.' As I am not an expert on crime and the original dataset contains a mix of variables that cover a variety of topics such as education (student-teacher ratio), economic value (property tax), and they already seem '*composite'* (ratios), I will refrain from engineering new ones.

Applied transformations:

- Firstly, all variables were converted from a mix of numerical and characters types to **all** numerical.

- The set was split into two, the original and one without the target variable for plotting.

- Considering the variables use different units, it is may be helpful to apply scaling to the entire dataset to bring values for every variable within the range of 0 to 1.

Model Building

Model 1

This model uses the original dataset with all available variables and no other transformation besides the type conversion to numerical. The reason why I decided to do this because the dataset itself appears to be high quality with reasonable variables.

```
Call:
glm(formula = target ~ ., family = binomial(), data = crimes)

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
```

Model 2

This model is an extension of the first one. I applied a stepwise approach using the built in function *stepAIC()* in both directions. This helped me narrow down the dataset from the 12 original variables to 8 (zn + nox + age + dis + rad + tax + ptratio + medv.) No additional transformations were applied.

```
Call:
glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
    medv, family = binomial(), data = crimes)

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
```

Model 3

This is the only model that uses scaled data. I applied a function that converts values every variable to represent them within the 0 to 1 range. It also uses handpicked variables based on the p-values of the first model. When designing this model, it was my expectation that it will perform best out of all 3.

```
Call:
glm(formula = target ~ nox + rad + dis + ptratio + medv + age +
    tax, family = binomial(), data = crimes_l10)

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)  -36.824      5.858  -6.286 3.26e-10 ***
nox           36.877      5.783   6.377 1.81e-10 ***
rad           16.842      3.346   5.033 4.82e-07 ***
dis            5.209      2.084   2.500 0.012433 *
ptratio        8.285      2.396   3.458 0.000545 ***
medv           4.683      1.678   2.791 0.005255 **
age            3.188      1.069   2.982 0.002867 **
tax           -5.856      1.802  -3.250 0.001153 **
---
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 Selection

Performance of models can be measured in many ways. I an external package called *caret* to tap into metrics that will help me identify the best performing model.

By running confusionMatrix() function on each of the models we can classify outcomes of our predictions into 4 buckets – True Positive, True Negative, False Positive and False Negative and at the same time calculate multiple metrics.

I extracted the data from the function above and put it into a new dataframe for easier model comparison. The table below shows overall accuracies and their ranges. We can easily determine that all 3 models behave similarly.

	MODEL1	MODEL2	MODEL3
ACCURACY	0.96	0.96	0.96
KAPPA	0.92	0.91	0.91
ACCURACY LOWER	0.94	0.94	0.94
ACCURACY UPPER	0.97	0.97	0.97
ACCURACY NULL	0.51	0.51	0.51
ACCURACY P-VALUE	0	0	0
MCNEMAR P-VALUE	0.52	0.75	0.75

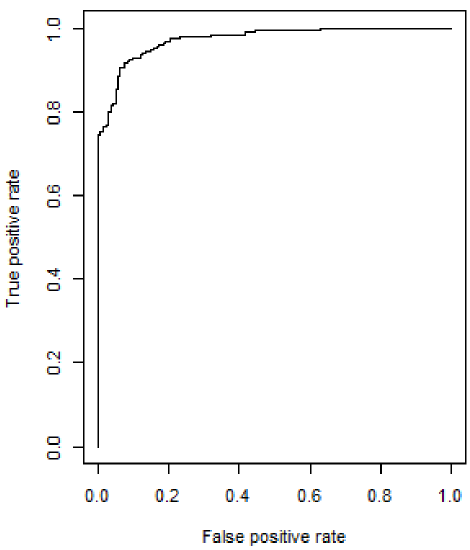
There are several additional metrics that can be extracted from the function and they are the following:

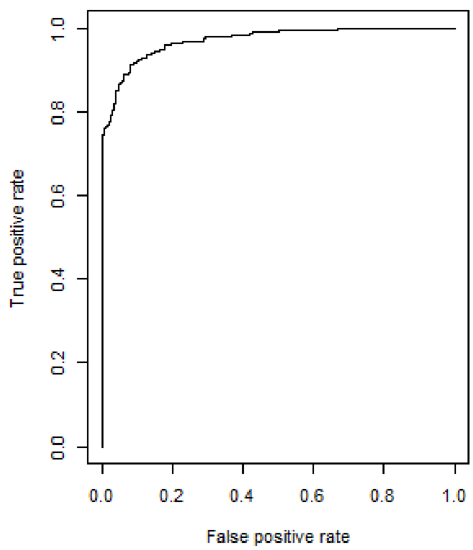
	MODEL1	MODEL2	MODEL3
SENSITIVITY	0.95	0.95	0.95
SPECIFICITY	0.96	0.96	0.96
POS PRED VALUE	0.96	0.96	0.96
NEG PRED VALUE	0.95	0.95	0.95
PRECISION	0.96	0.96	0.96
RECALL	0.95	0.95	0.95
F1	0.96	0.96	0.96
PREVALENCE	0.49	0.49	0.49
DETECTION RATE	0.47	0.47	0.47
DETECTION PREVALENCE	0.49	0.49	0.49
BALANCED ACCURACY	0.96	0.96	0.96

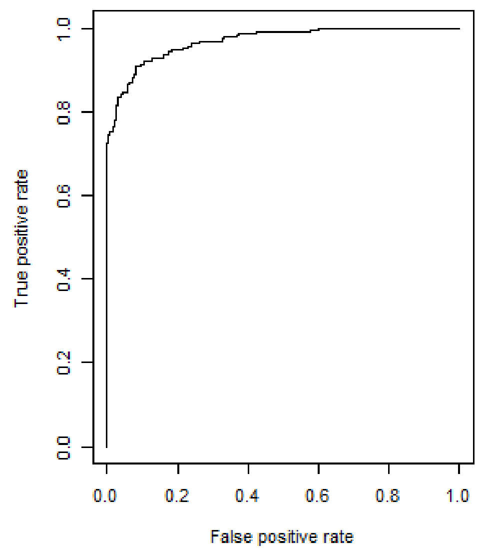
Once again, all 3 models tie in every single category.

Another great way to compare models is to determine their *Receiver Operating Characteristic* (ROC) and the Area Under the Curve (AUC). Package pROC provides a function that quickly calculated the AUC and also plots the results.

	MODEL1	MODEL2	MODEL3
AUC	0.9738	0.9719	0.9693







Summary

The dataset we worked with has proven to be great for building a crime ratio predicting model. It contained several variables with strong relationships to the target variable. We built 3 models which included all variables, only selected ones and performed a scaling to bring the values into a standardized range. The performance metrics outlined above indicated that all 3 scored high, with minor differences. Since the accuracy of the models did not differ so much, the selected model will be driven by other factors. **Model 2** is the winner as it did used only 8 variables (vs. 12 in model 1) and did not undergo any additional transformations (vs. model 3 – scaling). Further tuning of the model would include dropping at least one more variable (zn seems to be the least valuable), as well as outlier handling, other transformations to derive more advanced features. With accuracy score of 96% and other metric scores just as high, we can trust this model will help us predict whether the neighborhood will be at risk for high crime levels.

Crime Rate Predictor

Rafal Decowski

```
library(dplyr)
library(tidyr)
library(knitr)
library(stringr)
library(reshape2)

library(ggplot2)
library(corrplot)


########################################################
# Loading data and simple transformations
########################################################
crimes <- read.csv2('D:\\Rafal\\CUNY\\621\\hw\\hw3\\crime-training-data_modified.csv', sep=',', strings

# Convert to numerical type
crimes <- mutate_all(crimes, function(x) as.numeric(as.character(x)))

# Drop the target column for plotting
crimes_no_target <- crimes %>% select(-one_of('target'))



########################################################
# Descriptive statistics
########################################################

stats <- do.call(cbind, lapply(crimes, summary))

# Calculate standard deviation
d <- t(as.data.frame(sapply(crimes, function(x) sd(x))))
row.names(d) <- 'SD'
stats <- rbind(stats, d)
kable(stats)



########################################################
# Boxplots
########################################################
ggplot(data = melt(crimes_no_target[,1:6]), aes(x=variable, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  theme(legend.position="none") +
  facet_wrap( ~ variable, scales="free")

ggplot(data = melt(crimes_no_target[,7:12]), aes(x=variable, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  theme(legend.position="none") +
```

```r
  facet_wrap( ~ variable, scales="free")




##########################################################
# Correlation
##########################################################

cormat <- cor(crimes)
res1 <- cor.mtest(cormat, conf.level = .95)
corrplot(cormat, type = "lower", order = "hclust", tl.col = "black", tl.srt = 45, p.mat = res1$p, sig.le

# Target Correlation
target_corr <- cor(crimes)['target',]



##########################################################
# Logistic regression models
##########################################################
library(MASS)
library(scales)

# Model 1 contains all variables and is performed on the original dataset with no transformations
model1 <- glm (target ~ ., data = crimes, family = binomial())

# Adjusting model 1 based on the stepwise suggestions to create model 2
step_m1 <- stepAIC(model1, direction="both")
step_m1$anova
model2 <- glm(target ~ zn + nox + age + dis + rad + tax + ptratio + medv, data = crimes,   family=binomi


# Normalization/Scaling the values and handpicking variables
crimes_l10 <- data.frame(lapply(crimes, function(x) scale(x, center = FALSE, scale = max(x, na.rm = TRU
model3 <- glm(target ~ nox + rad + dis + ptratio + medv + age + tax, data = crimes_l10, family=binomial


summary(model1)
summary(model2)
summary(model3)


predict1 <- predict(model1, type = 'response')
predict2 <- predict(model2, type = 'response')
predict3 <- predict(model3, type = 'response')


##########################################################
# Measuring Performance
##########################################################
library(caret)
```

```
# Create Vectors for the predicted values
c1 <- c(crimes$target, predict1 > 0.5)
c2 <- c(crimes$target, predict2 > 0.5)
c3 <- c(crimes$target, predict3 > 0.5)
pred_df <- data.frame(crimes$target, c1, c2, c3)

# Measuring Performance
cm1 <- confusionMatrix(factor(pred_df$c1),factor(pred_df$crimes.target), positive = '1')
cm2 <- confusionMatrix(factor(pred_df$c2),factor(pred_df$crimes.target), positive = '1')
cm3 <- confusionMatrix(factor(pred_df$c3),factor(pred_df$crimes.target), positive = '1')

performance_measures1 <- round(data.frame(cm1$overall, cm2$overall, cm3$overall),2)
names(performance_measures1) <- c('model1', 'model2', 'model3')
performance_measures2 <- round(data.frame(cm1$byClass, cm2$byClass, cm3$byClass),2)
names(performance_measures2) <- c('model1', 'model2', 'model3')



########################################################
# ROC
########################################################
library(pROC)
par(mfrow=c(1, 3))
roc(crimes$target ~ predict1, crimes, plot=TRUE)
roc(crimes$target ~ predict2, crimes, plot=TRUE)
roc(crimes$target ~ predict3, crimes, plot=TRUE)







########################################################
# Predictions
########################################################

crimes_eval <- read.csv2('D:\\Rafal\\CUNY\\621\\hw\\hw3\\crime-evaluation-data_modified.csv', sep=',', 
crimes_eval <- mutate_all(crimes_eval, function(x) as.numeric(as.character(x)))



predictions <- predict(object = model2, crimes_eval, type = 'response')
target <- c(predictions > 0.5)

crimes_eval$predicted_prob <- round(predictions,2)
crimes_eval$target <- target

write.csv(crimes_eval, 'D:\\Rafal\\CUNY\\621\\hw\\hw3\\crime-predicted.csv')
```