CUNY MSDS HW3 - Binary Logistic Regression

Nicholas Schettini

CUNY School of Professional Studies

Table of Contents

Abstract

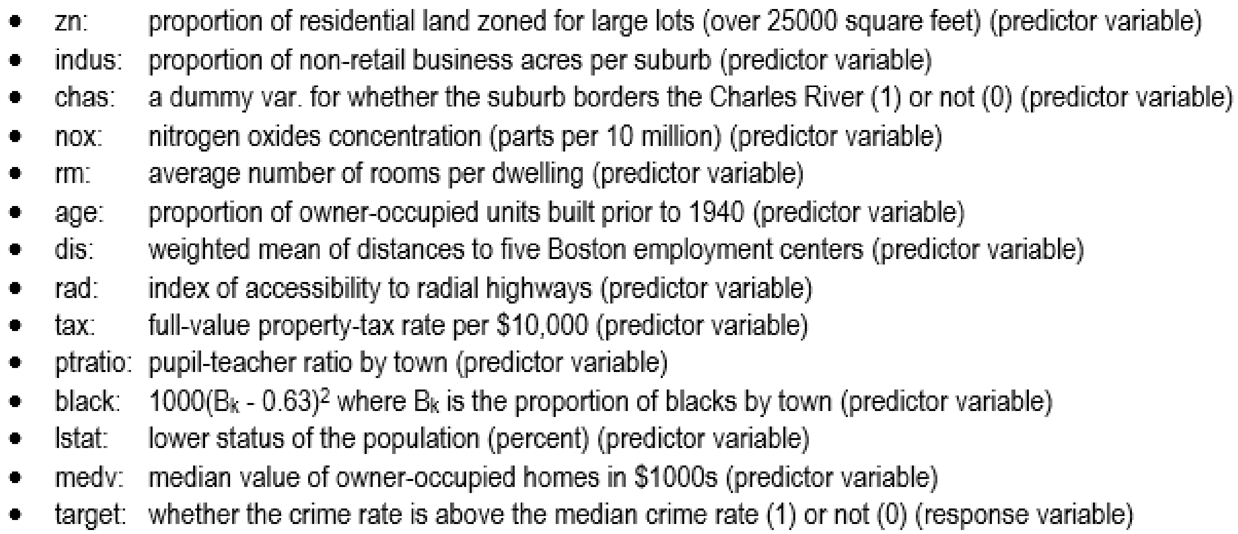
In this research assignment, we investigated data on crime in various neighborhoods of a major city. The data consisted of 13 variables, with 'target' as the response variable. The research included 4 overall groups: data exploration, data preparation, creating models, and selecting the best model. The data was visualized using multiple methods, including histograms and boxplots. The data was prepped by adding different transformations to help manipulate certain variables to become more normalized. Different models were created based on different approaches (for example, backwards elimination), and finally the best model was selected. The research shows that certain variables from within the dataset set were better predictors of crime for a major city than others, and that the selected model was able to predict crime based on an evaluation dataset. The best fitting model determined that, out of the evaluation dataset with a 0.3 threshold, the model predicted that there were 12 observations below the median crime rate, and about 28 above the median.

Keywords: R, crime, prediction, modeling, logistic binary regression

Overview

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).

Your 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. You will provide classifications and probabilities for the evaluation data set using your binary logistic regression model. You can only use the variables given to you (or variables that you derive from the variables provided). Below is a short description of the variables of interest in the data set:



- 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: $1000(B_k - 0.63)^2$ where B_k 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)

Data Exploration

The dataset contains 13 variables related to housing, transportation, the environment, education, and crime. For this data exploration, we will be focusing on a binary logistic regression for the response variable **target**, which can be either a 0 or a 1. A 1 indicates that the crime rate is above the median.

Based on the statistics below, it seems that the variables are not large enough to warrent any variable transformations (based on the kurtosis and skew). The data also show that there are no NA values.

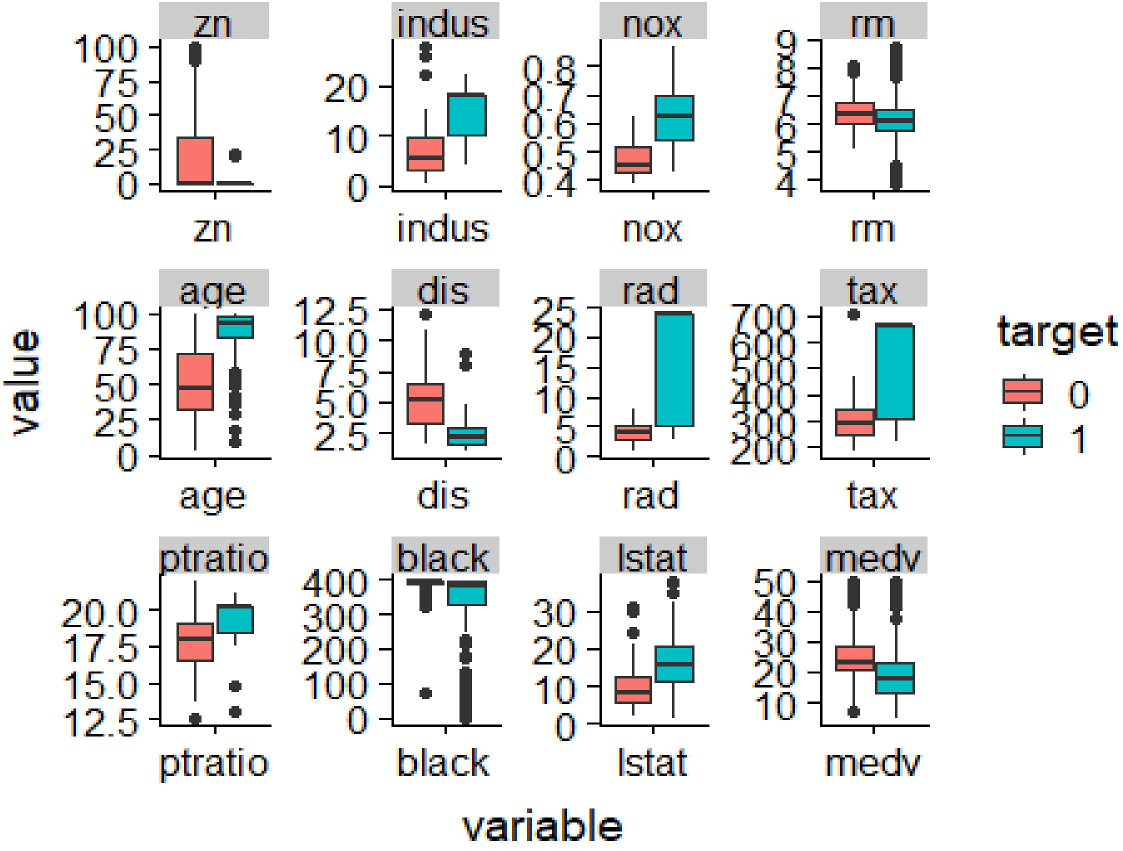
```
##         vars   n  mean    sd median trimmed   mad   min    max  range
## zn         1 466 11.58 23.36   0.00    5.35  0.00  0.00 100.00 100.00
## indus      2 466 11.11  6.85   9.69   10.91  9.34  0.46  27.74  27.28
## chas       3 466  0.07  0.26   0.00    0.00  0.00  0.00   1.00   1.00
## nox        4 466   0.55   0.12   0.54    0.54   0.13   0.39   0.87   0.48
## rm         5 466   6.29   0.70   6.21    6.26   0.52   3.86   8.78   4.92
## age        6 466  68.37  28.32  77.15   70.96  30.02   2.90 100.00  97.10
## dis        7 466   3.80   2.11   3.19    3.54   1.91   1.13  12.13  11.00
## rad        8 466   9.53   8.69   5.00    8.70   1.48   1.00  24.00  23.00
## tax        9 466 409.50 167.90 334.50  401.51 104.52 187.00 711.00 524.00
## ptratio   10 466  18.40   2.20  18.90   18.60   1.93  12.60  22.00   9.40
## black     11 466 357.12  91.32 391.34  383.51   8.24   0.32 396.90 396.58
## lstat     12 466  12.63   7.10  11.35   11.88   7.07   1.73  37.97  36.24
## medv      13 466  22.59   9.24  21.20   21.63   6.00   5.00  50.00  45.00
## target    14 466   0.49   0.50   0.00    0.49   0.00   0.00   1.00   1.00
##          skew kurtosis   se na_count
## zn       2.18     3.81 1.08        0
## indus    0.29    -1.24 0.32        0
## chas     3.34     9.15 0.01        0
## nox      0.75    -0.04 0.01        0
## rm       0.48     1.54 0.03        0
## age     -0.58    -1.01 1.31        0
## dis      1.00     0.47 0.10        0
## rad      1.01    -0.86 0.40        0
## tax      0.66    -1.15 7.78        0
## ptratio -0.75    -0.40 0.10        0
## black   -2.92     7.34 4.23        0
## lstat    0.91     0.50 0.33        0
## medv     1.08     1.37 0.43        0
## target   0.03    -2.00 0.02        0
```

Visual Exploration

Boxplots

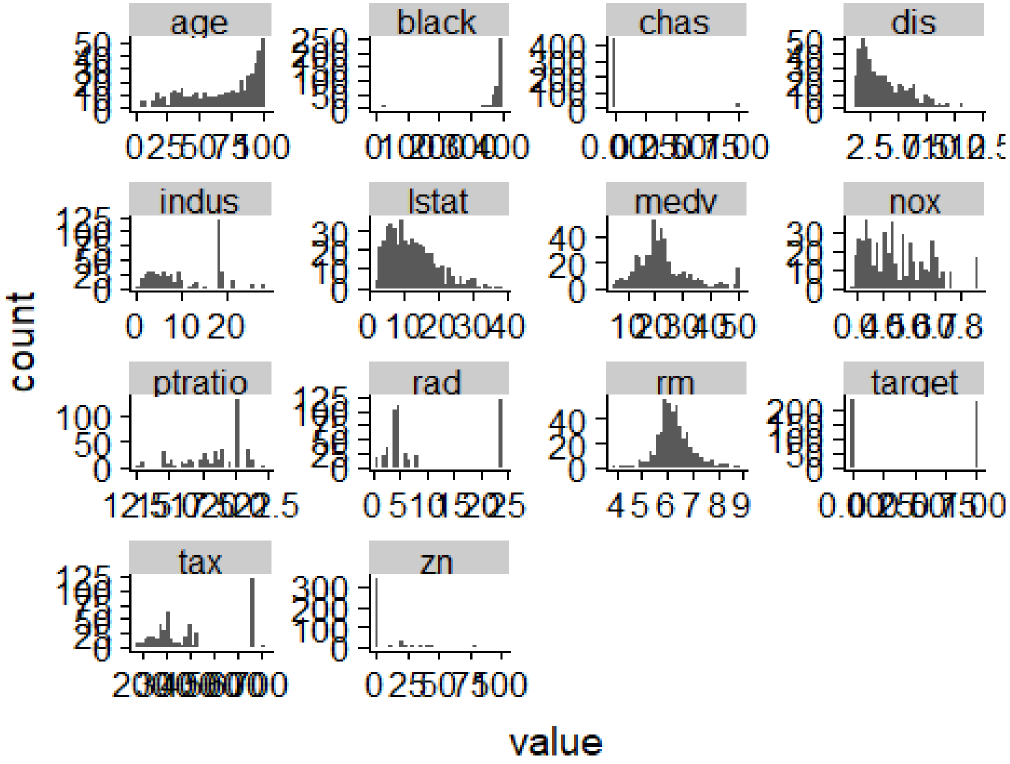
The below boxplots show all of the variables listed in the dataset. This visualization will assist in showing how the data is spread for each variable.

The boxplots show that some variables have a large amount of variance between each other, for example, **rad**, **zn**, and **tax**.



Histograms

Looking at the histograms below, it seems that the variable **zn**, **dis**, **age**, and **lstat** are skewed. Since they're skewed, it might suggest that we apply a transformation to help normalize those variables.



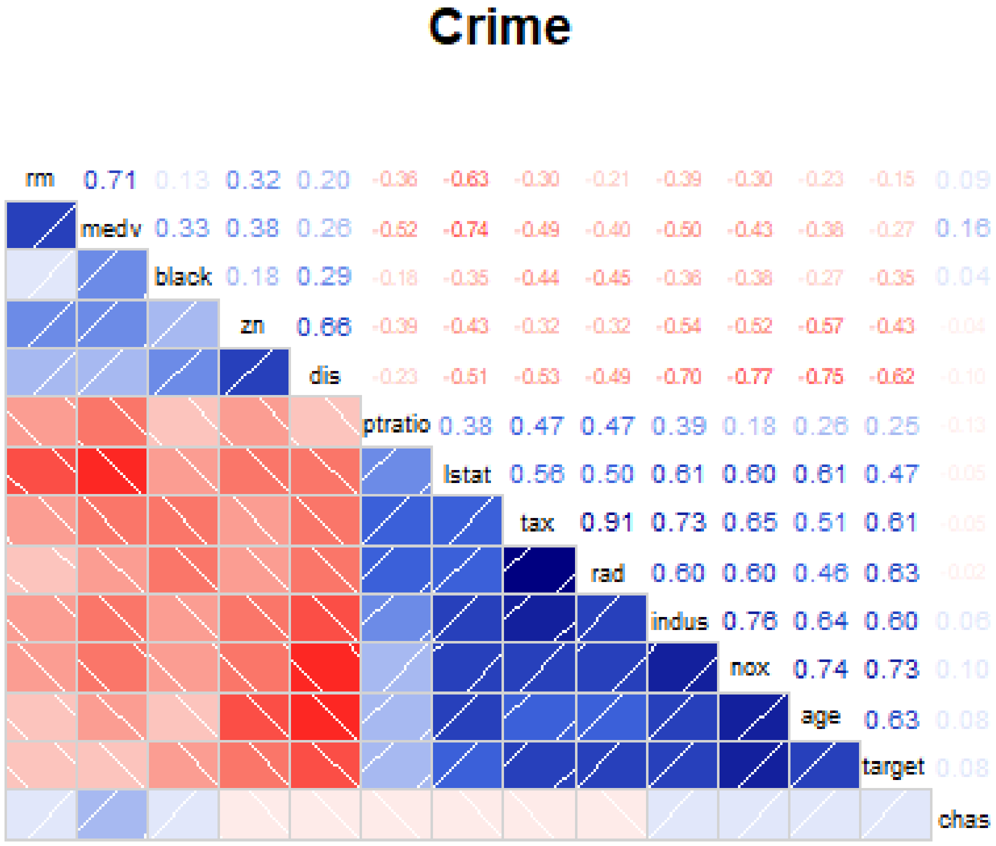
Correlation

The correlation plot below shows how variables in the dataset are related to each other. Looking at the plot, we can see that certain variables are more related than others.

For this project, it makes sense to break down the correlation by target - since that's what we're trying to predict.

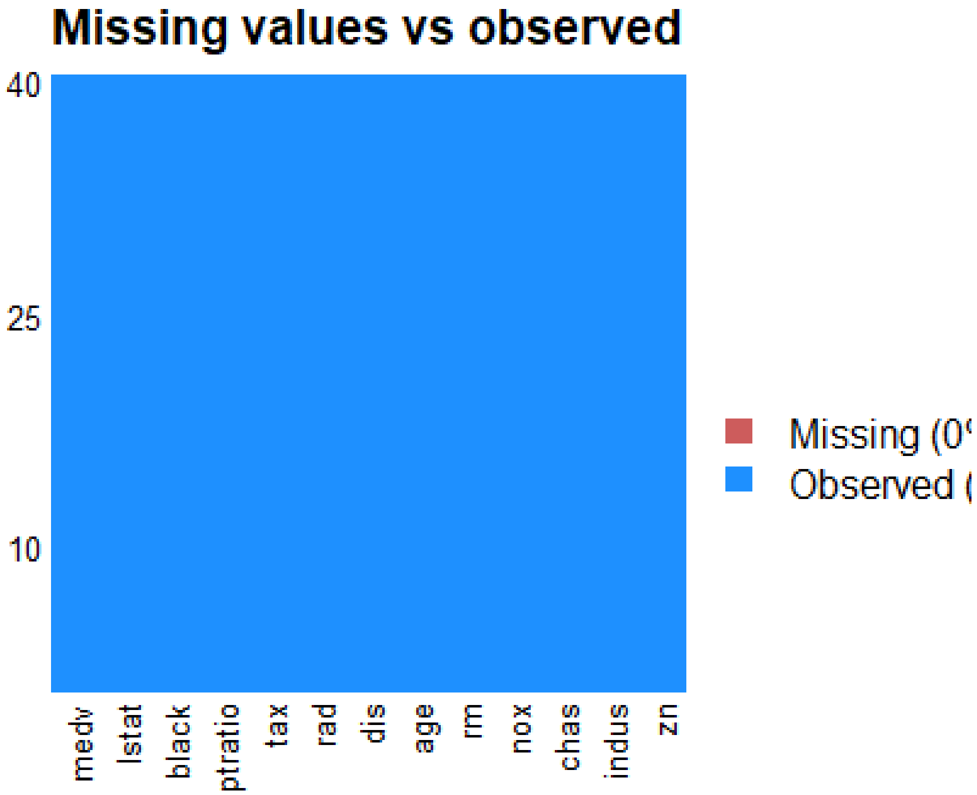
```
##          zn       indus        chas         nox          rm         age
## -0.43168176  0.60485074  0.08004187  0.72610622 -0.15255334  0.63010625
##         dis         rad         tax     ptratio       black       lstat
## -0.61867312  0.62810492  0.61111331  0.25084892 -0.35295680  0.46912702
##        medv      target
## -0.27055071  1.00000000
```

Below is a visual representation of the correlation plot. **rad** and **tax** are the top two variables with the highest correlation (0.91) with a positive correlation.



Missing Values

According to the graph, the data shows no missing variables. In this case, we will not have to account for NA values.



Data Preparation

The data in the crime dataset presents some factors that would lead one to have to perform some “Transformations” on the data. These transformations include adding the **log** of variables, and adding **quadratic** terms.

age and **lstat** are both skewed, so we will add log terms to the model to help build a better fitting model. **zn** and **rad** both have high variance, so applying a quadratic will help normalize the variables.

Build Models

Throughout this section, various models will be created to try to determine which will allow for the best “fit” to predict weather crime appears in a major city as given by the dataset. Different methods of model creation will be used, as discussed below.

Model 1 - Base Model: All variables

All of the variables will be tested to determine the base model they provided. This will allow us to see which variables are significant in our dataset, and allow us to make other models based on that. This model will be based off of the original data - before transformed (log/quad) variables have been added to account for potential issues in the data.

Looking at the model, 9 of the variables are statistically significant via their p-values. The variable nox has the largest cofficient, which also has the highest correlation with the response variable.

```
## 
## Call:
## glm(formula = target ~ ., family = "binomial", data = crime_train)
## 
## 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    
## chas          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
```

The models p-value is low - meaning the null must go - or we reject the null hypothesis that the coefficients are not related to the response variable.

The AIC is 214.15, with a residual deviance of 186.

Model 2 - Backwards elimination - original data

Variables will be removed one by one to determine best fit model. After each variable is removed, the model will be ‘ran’ again - until the most optimal output (r2, f-stat) are produced. Only the final output will be shown. This model is similar to the ‘forward selection’ variant - however I find it easier to work our way backwards and to eliminate variables rather than add them.

```
##
## Call:
## glm(formula = target ~ . - lstat - indus, family = binomial,
##     data = crime_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1802  -0.1317  -0.0019   0.0020   3.4471
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.695936   6.719182  -5.015 5.31e-07 ***
## zn           -0.058474   0.032118  -1.821  0.06867 .
## chas          0.883472   0.742410   1.190  0.23404
## nox          45.852121   7.184459   6.382 1.75e-10 ***
## rm           -0.906620   0.684211  -1.325  0.18515
## age           0.038058   0.012558   3.031  0.00244 **
## dis           0.764512   0.233408   3.275  0.00106 **
## rad           0.730418   0.158162   4.618 3.87e-06 ***
## tax          -0.007777   0.002716  -2.864  0.00419 **
## ptratio       0.435162   0.132545   3.283  0.00103 **
## black        -0.012373   0.006363  -1.945  0.05183 .
## medv          0.200483   0.071726   2.795  0.00519 **
## ---
## 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: 189.14  on 454  degrees of freedom
## AIC: 213.14
##
## Number of Fisher Scoring iterations: 9
```

The AIC is 213.14, with a residual deviance of 189.14. This model has a lower AIC when compared to model 1, however a slightly higher residual deviance.

Model 3 - All data - including transformations.

Model 3 includes all original variables, plus the created variables from the transformations (log and quad). The log and quadratic variables should help negate the large amount of skew in the data - or help them to become more normalized.

```
## 
## Call:
## glm(formula = target ~ ., family = "binomial", data = transform_crime)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1264  -0.1182  -0.0008   0.0006   3.6775  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -2.349e+01  7.674e+00  -3.061 0.002203 ** 
## zn          -5.076e-02  8.178e-02  -0.621 0.534815    
## indus       -1.025e-01  5.699e-02  -1.798 0.072209 .  
## chas         1.129e+00  8.046e-01   1.403 0.160687    
## nox          5.438e+01  9.013e+00   6.034  1.6e-09 ***
## rm          -1.369e+00  8.175e-01  -1.675 0.093911 .  
## age          1.009e-01  2.921e-02   3.454 0.000552 ***
## dis          6.962e-01  2.647e-01   2.630 0.008536 ** 
## rad          7.024e-01  7.040e-01   0.998 0.318466    
## tax         -8.260e-03  3.727e-03  -2.216 0.026687 *  
## ptratio      4.816e-01  1.414e-01   3.406 0.000660 ***
## black       -1.225e-02  6.335e-03  -1.933 0.053184 .  
## lstat        1.454e-01  1.266e-01   1.148 0.250784    
## medv         1.980e-01  7.674e-02   2.581 0.009860 ** 
## logage      -3.092e+00  1.102e+00  -2.806 0.005023 ** 
## loglstat    -1.957e+00  1.637e+00  -1.195 0.231961    
## quadzn      -1.865e-04  2.588e-03  -0.072 0.942554    
## quadrad      4.647e-03  6.345e-02   0.073 0.941613    
## ---
## 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: 177.33  on 448  degrees of freedom
## AIC: 213.33
## 
## Number of Fisher Scoring iterations: 13
```

The AIC is 213.33, with a residual deviance of 177.33 This model has a very slightly higher AIC when compared to model 2, however a slightly lower residual deviance. When compared to model 1, this performs slightly better with a lower AIC and residual deviance.

Model 4 - Only Significant Variables

Model 4 uses only significant variables - including those from the transformations.

```
## 
## Call:
## glm(formula = target ~ indus + nox + rm + age + dis + tax + ptratio + 
##     black + medv + logage, family = "binomial", data = transform_crime)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.47882  -0.37078  -0.06167   0.16655   2.97476  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -24.926302   5.805363  -4.294 1.76e-05 ***
## indus        -0.136749   0.047429  -2.883 0.003936 ** 
## nox          43.493967   6.654248   6.536 6.31e-11 ***
## rm           -0.701425   0.534087  -1.313 0.189077    
## age           0.068533   0.022810   3.005 0.002660 ** 
## dis           0.510113   0.159032   3.208 0.001338 ** 
## tax           0.004350   0.001724   2.524 0.011616 *  
## ptratio       0.367778   0.101720   3.616 0.000300 ***
## black        -0.011914   0.005596  -2.129 0.033267 *  
## medv          0.195032   0.055063   3.542 0.000397 ***
## logage       -1.884538   0.970968  -1.941 0.052272 .  
## ---
## 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: 239.91  on 455  degrees of freedom
## AIC: 261.91
## 
## Number of Fisher Scoring iterations: 7
```

The AIC is 261.91 - which is significantly higher than previous models, with a residual deviance of 239.91 - which is also much higher than previous models. This model performs slightly worse than model 1.

Model 5 - Backwards Elimination - Only Significant Variables

Model 5 uses only significant variables - including those from the transformations. Variables will be removed one by one to determine best fit model. After each variable is removed, the model will be 'ran' again - until the most optimal output (r2, f-stat) are produced. Only the final output will be shown.

```
## 
## Call:
## glm(formula = target ~ indus + nox + rm + age + dis + tax + ptratio + 
##     black + medv + logage, family = "binomial", data = transform_crime)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.47882  -0.37078  -0.06167   0.16655   2.97476  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -24.926302   5.805363  -4.294 1.76e-05 ***
## indus        -0.136749   0.047429  -2.883 0.003936 ** 
## nox          43.493967   6.654248   6.536 6.31e-11 ***
## rm           -0.701425   0.534087  -1.313 0.189077    
## age           0.068533   0.022810   3.005 0.002660 ** 
## dis           0.510113   0.159032   3.208 0.001338 ** 
## tax           0.004350   0.001724   2.524 0.011616 *  
## ptratio       0.367778   0.101720   3.616 0.000300 ***
## black        -0.011914   0.005596  -2.129 0.033267 *  
## medv          0.195032   0.055063   3.542 0.000397 ***
## logage       -1.884538   0.970968  -1.941 0.052272 .  
## ---
## 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: 239.91  on 455  degrees of freedom
## AIC: 261.91
## 
## Number of Fisher Scoring iterations: 7
```

The AIC is 261.91 - which is significantly higher than previous models, with a residual deviance of 239.91 - which is also much higher than previous models.

Model 6 - Leaps Package

The Leaps package is an “regression subset selection” tool. The package automatically generates all possible models. The tool is basically used to find the “best” model. This tool will be compared to another (further down below). According to the documentation, the algorithm within leaps returns a best “model of each size”.

After running the package, I inputed the “best” model mannualy in R, as to not have to rerun each time. The below output shows the results of that model.

```
## 
## Call:
## glm(formula = transform_crime$target ~ nox + age + rad + ptratio + 
##     medv + logage + quadrad, family = "binomial", data = transform_crime)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.05357  -0.30004  -0.04184   0.01074   2.91188  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -20.108923   4.384475  -4.586 4.51e-06 ***
## nox          25.411290   4.193786   6.059 1.37e-09 ***
## age           0.055513   0.020528   2.704  0.00685 ** 
## rad           0.539888   0.421872   1.280  0.20064    
## ptratio       0.273311   0.101230   2.700  0.00694 ** 
## medv          0.087290   0.029080   3.002  0.00268 ** 
## logage       -1.811286   0.862736  -2.099  0.03578 *  
## quadrad      -0.001642   0.039667  -0.041  0.96698    
## ---
## 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: 221.27  on 458  degrees of freedom
## AIC: 237.27
## 
## Number of Fisher Scoring iterations: 12
```

The AIC is 237.27- which is significantly lower than previous models, with a residual deviance of 221.2- which is also lower than previous models.

Model 7- glmulti Package

The glmulti package is an “automated model selection and model averaging” tool. The package automatically generates all possible models “with the specified response and explanatory variables”. The tool is basically used to find the “best” model. The library/function itself took over 10 minutes for my PC to run - since it was set to use an exhaustive approach to selecting the best model.

After running the package, I inputed the “best” model mannualy in R, as to not have to rerun (10+ mins) each time.

The glmulti package is designed to handle binary logistic regression.

glmulti - all data including transformations

```
## 
## Call:
## glm(formula = transform_crime$target ~ nox + age + rad + ptratio + 
##     medv + logage + quadrad, data = transform_crime)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -0.62263  -0.18943  -0.04644   0.16223   0.97175  
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.0819950  0.3189065  -3.393 0.000752 ***
## nox          1.8557632  0.2169094   8.555  < 2e-16 ***
## age          0.0075058  0.0018394   4.081 5.30e-05 ***
## rad          0.0659960  0.0154537   4.271 2.37e-05 ***
## ptratio      0.0146221  0.0085612   1.708 0.088321 .  
## medv         0.0077351  0.0019676   3.931 9.75e-05 ***
## logage      -0.1829056  0.0744502  -2.457 0.014390 *  
## quadrad     -0.0017688  0.0005524  -3.202 0.001459 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.09394292)
## 
##     Null deviance: 116.466  on 465  degrees of freedom
## Residual deviance:  43.026  on 458  degrees of freedom
## AIC: 230.26
## 
## Number of Fisher Scoring iterations: 2
```

glmulti - all original data only

```
## 
## Call:
## glm(formula = crime_train$target ~ 1 + nox + age + rad + ptratio + 
##     medv, data = transform_crime)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -0.5997  -0.2030  -0.0439   0.1252   0.9264  
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.4128361  0.2249301  -6.281 7.79e-10 ***
## nox          1.9566942  0.2157623   9.069  < 2e-16 ***
## age          0.0035317  0.0007664   4.608 5.27e-06 ***
## rad          0.0171066  0.0023402   7.310 1.19e-12 ***
## ptratio      0.0127163  0.0086324   1.473    0.141    
## medv         0.0080212  0.0019934   4.024 6.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.09682873)
## 
##     Null deviance: 116.466  on 465  degrees of freedom
## Residual deviance:  44.541  on 460  degrees of freedom
## AIC: 242.39
## 
## Number of Fisher Scoring iterations: 2
```

Looking at the two models for the glmulti package, the model with the transformed variables performs slightly better. The transformed model has both a lower residual deviance and AIC value. The model actually has some of the best values from all of the other models. The base model comes close to this model – with it’s AIC being slightly lower and residual deviance being higher.

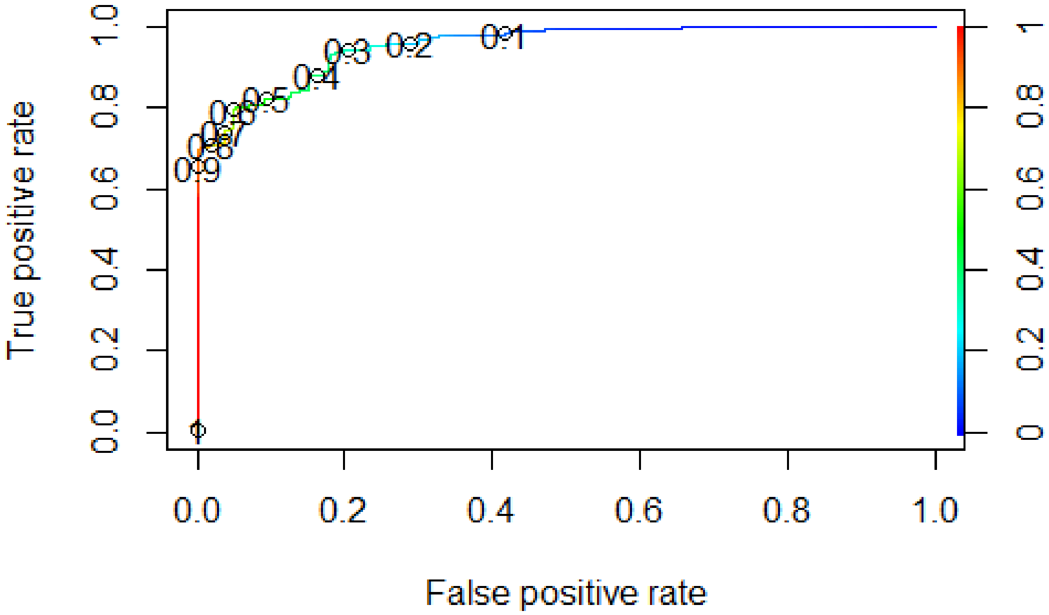
Model Selection

Based on the above models, I’ve decided to use model 7 - as determined by the glmulti package. The AIC and residual deviance for this model seemed to give the best values that would be suited for the prediction. Model 7 also had the lowest residual deviance – which means it had the best “fit” (since higher numbers indicate bat fit), and significant p-values.

Evaulating the model

Before I ran the evaulation data through the model, I decided to split the trianing data into an 80/20 split. My thoughts are that this will allow me to better check the accuracy of my model, given the fact that this way I can actually check if the ‘target’ variable as predicted by the model is correct or not.

After splitting the data and creating two new variables (training and testing), I created an ROC graph to help determine what threshold I should use in my model



Looking at the graph, the 0.3 threshold seems to be the most ideal soultion for my testing. 0.3 gives about a 0.9 TP rate, while giving only ~0.2 FP rate. 0.2 and 0.2 give slightly higher TP rates, but also give a high FP rate - which, in my opinion, isn’t worth the slight increase in TP. 0.4 gives a slightly lower FP rate, but a siginifcant different in TP rate.

Confusion Matrix

```
##          PredictedValue
## ActualValue FALSE TRUE
##          0   138   37
##          1    12  176

## [1] 0.873
```

After testing model 7, on the split training data, the accuracy is around .873, which seems like a good fit. The **sensitivity** is around 0.82, and **specificity** is around 0.945. The sensitivity shows us how correctly the model can identify the true positive rate. The specificity is the ability to test the true negative rate. In this case, both factors are high enough that our model shows a good fit.

The **misclassification** rate is about 0.135.

The **precision** is about 0.827.

The **F-Score** is 0.823. The F-Score is a measure of the models accuracy and considers both precision and recall to calculate this score. In this case, the higher the score the better the model.

```
##          PredictedValue
## ActualValue FALSE TRUE
##          0    48   14
##          1     1   40

## [1] 0.894
```

After testing model 7, on the split testing data, the accuracy is around .894, which seems like a good fit. I believe this model will be sufficient to test the evaluation data with.

Testing With Model 7

Using the glmulti model (7) on the evaulation dataset, with a threshold of 0.3, the model predicts that there are 12 observations below the median crime rate, and about 28 above the median crime rate.

```
## predict12
##  0  1
## 12 28

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2763  0.2616  0.4140  0.5128  0.8257  1.2303
```

For a point of refrence, I decided to run the first model against the evaulation dataset as well (baseline model). This model seems to predict an even 20/20 split for the predictions.

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 0.0000009 0.0832492 0.4714571 0.5201166 0.9999995 1.0000000

## predict11
##  0  1 
## 20 20
```

References

All subset regression with leaps, bestglm, glmulti, and meifly. (n.d.). Retrieved from https://rstudio-pubs-static.s3.amazonaws.com/2897_9220b21cfc0c43a396ff9abf122bb351.html

All subset regression with leaps, bestglm, glmulti, and meifly. (n.d.). Retrieved from https://rstudio-pubs-static.s3.amazonaws.com/2897_9220b21cfc0c43a396ff9abf122bb351.html

All subset regression with leaps, bestglm, glmulti, and meifly. (n.d.). Retrieved from https://rstudio-pubs-static.s3.amazonaws.com/2897_9220b21cfc0c43a396ff9abf122bb351.html

Model selection and multimodel inference made easy. (n.d.). Retrieved from https://cran.r-project.org/web/packages/glmulti/glmulti.pdf

Markham, K. (2016, June 08). Simple guide to confusion matrix terminology. Retrieved from https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Appendix

library(tidyverse) library(knitr) library(psych) library(readr) library(kableExtra) library(ggiraph) library(cowplot) library(reshape2) library(corrgram) library(gridExtra) library(usdm) library(mice) library(pROC) library(reshape2) library(caTools) library(caret) library(ROCR)

crime_train <- read_csv(“https://raw.githubusercontent.com/nschettini/CUNY-MSDS-DATA-621/master/crime-training-data.csv”) crime_eval <- read_csv(“https://raw.githubusercontent.com/nschettini/CUNY-MSDS-DATA-621/master/crime-evaluation-data.csv”)

train <- describe(crime_train) train$na_count <- sapply(crime_train, function(y) sum(length(which(is.na(y)))))

kable(train, “html”, escape = F) %>% kable_styling(“striped”, full_width = T) %>% column_spec(1, bold = T) %>% scroll_box(width = “100%”, height = “700px”)

long <- melt(crime_train, id.vars= “target”)%>% dplyr::filter(variable != “chas”) %>% mutate(target = as.factor(target))

ggplot(data = long, aes(x = variable, y = value)) + geom_boxplot(aes(fill = target)) + facet_wrap(~ variable, scales = “free”)

crime_hist <- crime_train

crime_hist %>% keep(is.numeric) %>%
gather() %>%
ggplot(aes(value)) +
facet_wrap(~ key, scales = “free”) +
geom_histogram(bins = 35)

ggplot(crime_train, aes(crime_train$medv ,target)) + geom_point() + geom_smooth(method = “glm”, method.args = list(family = “binomial”), se = FALSE)

kable(cor(drop_na(crime_train))[,14], “html”, escape = F) %>% kable_styling(“striped”, full_width = F) %>% column_spec(1, bold = T) %>% scroll_box(height = “500px”)

corrgram(drop_na(crime_train), order=TRUE, upper.panel=panel.cor, main=“Moneyball”)

library(Amelia) missmap(crime_eval, main = “Missing values vs observed”)

transform_crime <- crime_train transform_crime$logage < -log(transform_{c}rime$age) transform_crime$loglstat < -log(transform_{c}rime$lstat) transform_crime$quadzn < -transform_{c}rime$zn^{2} $^{\text{transform_crime}}$$quadrad < -transform_{c}rime$$^{\text{rad}}$2

crime_eval1 <- crime_eval crime_eval1$logage < -log(crime_{e}val$age) crime_eval1$loglstat < -log(crime_{e}val$lstat) crime_eval1$quadzn < -crime_{e}val$zn^{2} $^{\text{crime_eval1}}$$quadrad < -crime_{e}val$$^{\text{rad}}$2

model1 <- glm(target ~., family = “binomial”, data=crime_train) summary(model1) model2 <- glm(target ~. -lstat -indus, family = binomial, data=crime_train) summary(model2)

model3 <- glm(target ~., family = “binomial”, data=transform_crime) summary(model3)

model4 <- glm(target ~ indus + nox + rm + age + dis + tax + ptratio + black +medv + logage, family = “binomial”, data=transform_crime) summary(model4)

model5 <- glm(target ~ indus + nox + rm + age + dis + tax + ptratio + black +medv + logage, family = “binomial”, data=transform_crime) summary(model5)

library(leaps)

x <- model.matrix(transform_crime$target\ . -1, data = transform_{c}rime)y < -transform_{c}rime$target bestmods <- leaps(x, y, nbest=1) bestmods min(bestmods$Cp)

leapsmodel <- glm(transform_crime$target ~ nox + age + rad + ptratio + medv + logage + quadrad, family = “binomial”, transform_crime)

summary(leapsmodel)

library(rJava) library(glmulti)

glmulti.lm.out <- glmulti(crime_train$target ~., data = crime_train, level = 1, # No interaction considered method = “h”, # Exhaustive approach crit = “aic”, # AIC as criteria confsetsize = 5, # Keep 5 best models plotty = F, report = F, # No plot or interim reports fitfunction = “lm”) # lm function

glmulti.lm.out@formulas

modelglmulti <- glm(transform_crime$target ~ nox + age + rad + ptratio + medv + logage + quadrad, data = transform_crime)

summary(modelglmulti)

modelglmulti2 <- glm(crime_train$target ~ 1 + nox + age + rad + ptratio + medv, data = transform_crime)

summary(modelglmulti2)

splitdata <- transform_crime

split <- sample.split(splitdata, SplitRatio = 0.8) split training <- subset(splitdata, split == “TRUE”) testing <- subset(splitdata, split == “FALSE”)

modelglmulti3 <- glm(training$target ~ 1 + nox + age + rad + ptratio + medv, family=“binomial”, data = training) res <- predict(modelglmulti3, newdata=training, type=“response”)

ROCRPred = prediction(res, training$target) ROCRPref <- performance(ROCRPred, “tpr”,“fpr”)

plot(ROCRPref, colorize=TRUE, print.cutoffs.at=seq(0.1,by=0.1))

```
(table(ActualValue=training$target, PredictedValue=res>0.3))
round((149+167)/(149+167+9+37),3)

res <- predict(modelglmulti3, newdata=testing, type=“response”)
(table(ActualValue=testing$target, PredictedValue=res>0.3)) round((42+51)/(42+51+9+2),3)

predict1 <- predict(modelglmulti, newdata=crime_eval1, type=“response”) predict12 <-
ifelse(predict1 > 0.3, 1, 0) table(predict12) summary(predict1)

predict2 <- predict(model1, newdata=crime_eval1, type=“response”) summary(predict2)
predict11 <- ifelse(predict2 > 0.5, 1, 0) table(predict11)
```