Crime Logistical Model, Homework #3, Group #5

Anthony A, Alice A. Friedman, Seung min song

04/02/2023

Abstract

We will evaluate data related to neighborhood crime to create a model to predict if a neighborhood is "high crime," which we are defining as above the median crime rate. To do this, we will create a logistical model that rates each neighborhood's crime level.

Data Exploration

The data has 12 features and the training data set additionally has 1 target variable, which marks neighborhoods as "high crime" (target==1) or not high crime (target==0).

- 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)

Assumptions for Logistical Regression

When we do logistical regression, we have to consider our assumptions and then verify our data holds up to them. The assumptions necessary for logistical regression are:

1. Binary outcome. The model will predict only "yes" or "no". This assumption is met as we are predicting only "high crime" or "not high crime" rather than any particular level of crime.

2. No multicollinearity between features: The features should be independent of each other.

3. Independent observations: The values in any one observation should not affect the values in any other. In practice, this is not likely to be strictly true as being *near* a high crime neighborhood should certainly affect the crime levels of other neighborhoods. However, this is an assumption we will have to say is "true enough" for the purpose of this model.

4. Features are linearly related to the log-odds of the target variable. Note that this does not mean the the features are linearly related to the target itself, as with linear regression.
5. Large sample size: Our minimum sample size at n% probability would be 10*13/n so our sample size is adequate (466 > 260) where 50% of samples are expected to be over the median by definition.
6. No outliers: Outliers can have a significant distorting effect in logistic regression, and so should typically be removed before building the model.

Going forward, for our research question we ask "is the crime present in this neighborhood above the median rate?"

Summary Statistics

Upon analyzing the target variable, we observe that 237 out of the total observations have a crime rate below the median, whereas 229 have a crime rate above the median. Consequently, our training data set comprises an almost equal number of neighborhoods categorized as at-risk and not-at-risk.

```
##        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
```

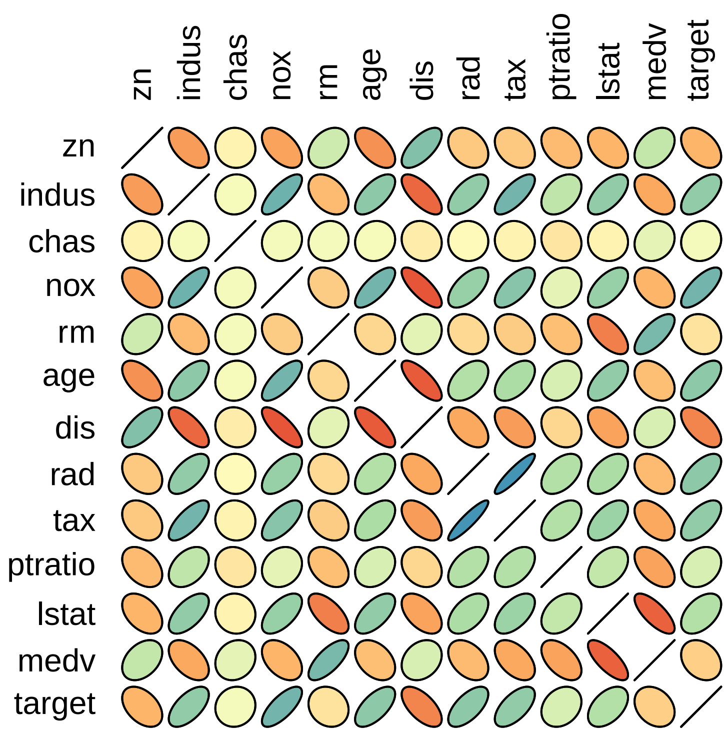
Correlation

Our first curiosity is if there is any strong correlation to the target variable. Accordingly, the nox, age, rad, tax, and indus variables show moderate positive correlation with the target (>0.6). Additionally, the dis variable shows moderate negative correlation with the target(< -0.6).

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

However, to verify our assumptions to run logistical regression, we also need to verify that there is no multicollinearity. According to our visualization of the correlation matrix below, there are several variables that appear to be collinear (|correlation| > 0.7):

This will need to be dealt with before proceeding with model development.



	zn	indus	chas	nox	rm	age	dis	rad	
zn	1.0000000	-0.5382664	-0.0401620	-0.5170452	0.3198141	-0.5725805	0.6601243	-0.3154812	-0.319
indus	-0.5382664	1.0000000	0.0611832	0.7596301	-0.3927118	0.6395818	-0.7036189	0.6006284	0.732
chas	-0.0401620	0.0611832	1.0000000	0.0974558	0.0905098	0.0788837	-0.0965771	-0.0159004	-0.046
nox	-0.5170452	0.7596301	0.0974558	1.0000000	-0.2954897	0.7351278	-0.7688840	0.5958298	0.653
rm	0.3198141	-0.3927118	0.0905098	-0.2954897	1.0000000	-0.2328125	0.1990158	-0.2084457	-0.296
age	-0.5725805	0.6395818	0.0788837	0.7351278	-0.2328125	1.0000000	-0.7508976	0.4603143	0.512
dis	0.6601243	-0.7036189	-0.0965771	-0.7688840	0.1990158	-0.7508976	1.0000000	-0.4949919	-0.534
rad	-0.3154812	0.6006284	-0.0159004	0.5958298	-0.2084457	0.4603143	-0.4949919	1.0000000	0.906
tax	-0.3192841	0.7322292	-0.0467648	0.6538780	-0.2969343	0.5121245	-0.5342546	0.9064632	1.000
ptratio	-0.3910357	0.3946898	-0.1286606	0.1762687	-0.3603471	0.2554479	-0.2333394	0.4714516	0.474
lstat	-0.4329925	0.6071102	-0.0514232	0.5962426	-0.6320245	0.6056200	-0.5075280	0.5031013	0.564
medv	0.3767171	-0.4961743	0.1615653	-0.4301227	0.7053368	-0.3781560	0.2566948	-0.3976683	-0.490
target	-0.4316818	0.6048507	0.0800419	0.7261062	-0.1525533	0.6301062	-0.6186731	0.6281049	0.611

Data structure

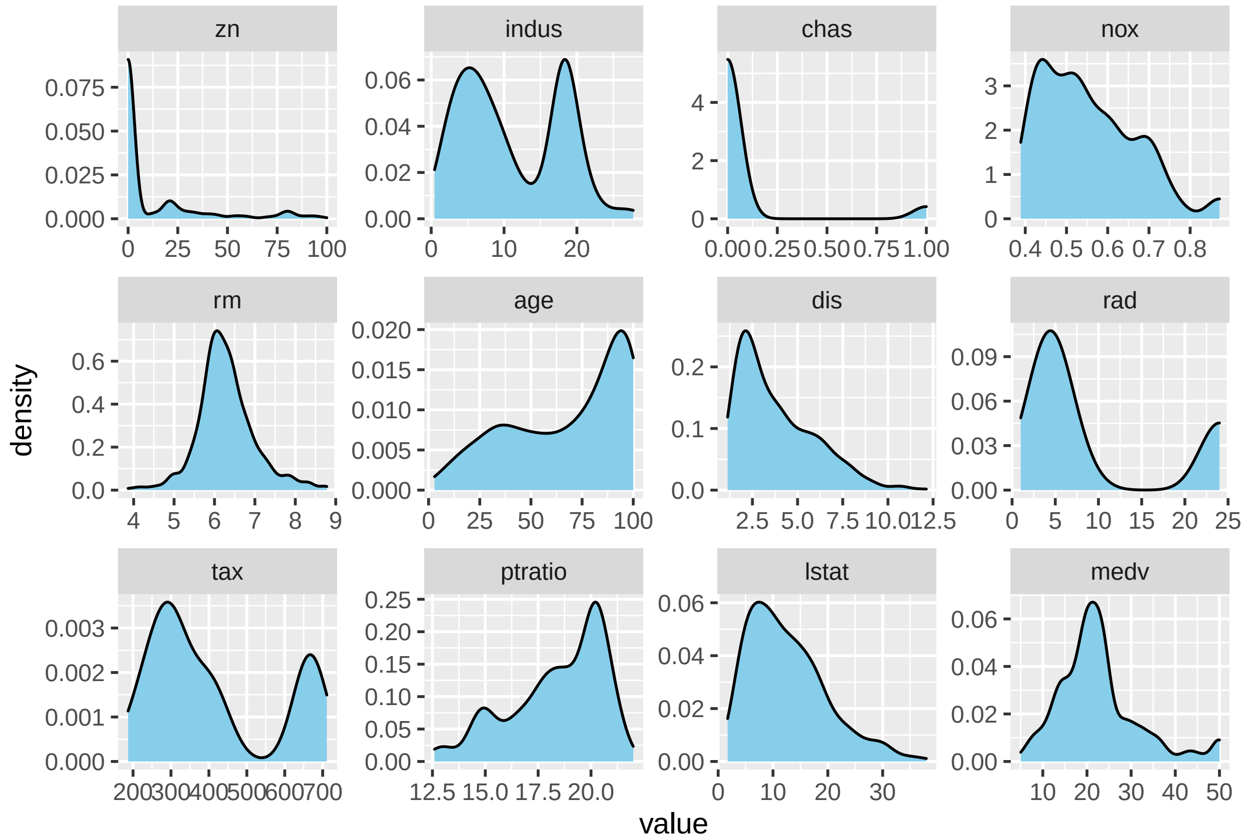
There are 466 observations and 13 variables in the training dataset.

Missing values

The dataset is complete with no missing values, so imputation is not necessary on this dataset.

Visulization of the data set

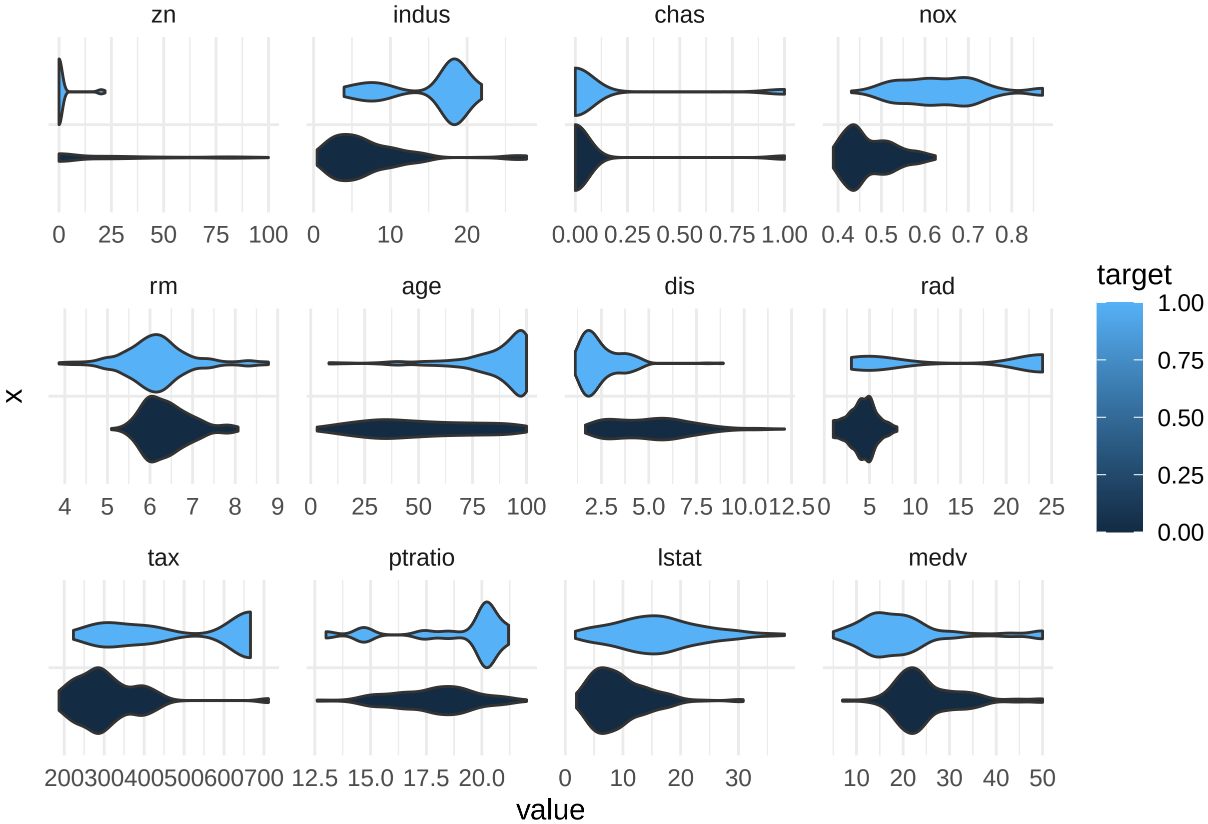
Let's examine what the data looks like without separating the target variables. At first, we notice that indus and rad appear binomial. Additionally, age and nox seem to be skewed in one direction. We do not have to normalize the data until necessary because it is not a necessary condition for logistical regression.



Next, let's examine the distribution of the features with respect to the target variable. Shown below, in light blue are the distributions of the features where crime is high (target is 1), and in dark blue are the distributions where crime is low.

Noticeably, some features strongly show a difference between the two plots, such as indus whereas others are more similar. We should expect all of the models to weigh indus heavily.

Additionally, we can expect other features with notable differences to get weighed with some significance such as nox and lstat and rad. Features like chas, where the distributions are nearly identical, are not likely targets for the model. This makes sense, as there is no obvious causal reason why bordering the Charles River should affect crime.



Data Preparation

As both chas and target are categorical variables, we have changed their class from integer to factor.

Cross Validation

To verify our results, we will split the available labeled data into training and validation sets. This will help us score our model on test data and to be able to verify the results later.

Outliers

Based on the analysis above, there were no outliers that needed to be removed as all the values appeared to be reasonable.

Buckets

The code is using the summary information provided above to transform the data by putting it into "buckets" or categories. The variable cols_to_bin contains the names of the columns in the data set that we want to bin, and breaks contains the breaks for the buckets we want to use for each variable.

```
## binned_col
##    [0,20]  (20,40]  (40,60]  (60,80] (80,100]
```

```
##         372            46            16            19            13
## binned_col
##    [0,5]    (5,10]  (10,15]  (15,20]  (20,30]
##      113       135        42       152        24
## binned_col
##    [0,0.4]  (0.4,0.5]  (0.5,0.6]  (0.6,0.7]  (0.7,0.9]
##          9        168        137         99         53
## binned_col
## [0,4]  (4,5]  (5,6]  (6,7]  (7,8]  (8,9]
##     1     14    142    249     48     12
## binned_col
##    [0,20]    (20,40]    (40,60]    (60,80]  (80,100]
##        33         70         69         75       219
## binned_col
##    [0,2]     (2,4]     (4,6]     (6,8]    (8,10]  (10,12]
##      103       186        96        62        14         4
## binned_col
##    [0,5]    (5,10]  (10,15]  (15,20]  (20,25]  (25,30]
##      285        60         0         0      121         0
## binned_col
##       [0,250]     (250,500]     (500,750]  (750,1e+03]
##            61           279           126            0
## binned_col
##   [0,14]  (14,16]  (16,18]  (18,20]  (20,22]
##       16       66      100      100      184
## binned_col
##    [0,5]    (5,10]  (10,15]  (15,20]  (20,25]  (25,30]  (30,40]
##       57       145      114       83       37       19       11
## binned_col
##   [0,10]  (10,20]  (20,30]  (30,40]  (40,50]
##       23      173      191       50       29
```

New Variables

tax_per_room (tpr) variable would represent the insights into the relationship between the cost of living in a particular area and the size of the living space. A higher value of **tax_per_room (tpr)** means that the tax rate is higher relative to the number of rooms in the dwelling.

age_dis_ratio(adr) can help identify areas that have an older housing inventory and are farther from job centers, which may affect preference for those locations.

```
##   zn indus chas   nox    rm   age    dis rad tax ptratio lstat medv target
## 1  0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7  3.70 50.0      1
## 2  0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 26.82 13.4      1
## 3  0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 18.85 15.4      1
## 4 30  4.93    0 0.428 6.393   7.8 7.0355   6 300    16.6  5.19 23.7      0
## 5  0  2.46    0 0.488 7.155  92.2 2.7006   3 193    17.8  4.82 37.9      0
## 6  0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9  7.67 26.5      0
##         tpr       adr
## 1  50.82608 47.020871
## 2  74.58819 75.665860
## 3 102.69854 50.545896
## 4  46.92633  1.108663
## 5  26.97414 34.140561
```

```
## 6   56.62882  24.964112
```

Logistical Model Building

To begin, we will create a null model that does not make any prediction. This will help us verify our first model works better than random guessing. Therefore, the first model that beats the residual deviance of 646 will be our first model candidate.

```
##
## Call:
## stats::glm(formula = target ~ NULL, family = "binomial", data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.163   -1.163   -1.163    1.192    1.192
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.03434    0.09266  -0.371    0.711
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 645.88  on 465  degrees of freedom
## AIC: 647.88
##
## Number of Fisher Scoring iterations: 3
```

Model 1 - Full Model

We will start with a top-down approach and begin by including all of the variables.

According to this, our most statistically consistent variables are those with extremely low p-values. Notably, the variables nox and rad both have very high significance below 0.001 so we will definitely use these in the future. This is expected based on their distributions with respect to the target variable, above.

Thankfully, the AIC and residual deviance are better than the null model already, so we are off to a good start.

Multicollinearity will impact our ability to evaluate the impact of any one variable, and because there is known multicollinearity among the feature set, we will want a model with fewer or combined features as our final model in order to understand the relationship between the features and crime levels better.

```
##
## Call:
## stats::glm(formula = target ~ ., family = "binomial", data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90563  -0.16864  -0.00079   0.00118   2.70123
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -51.250712  11.063253  -4.633 3.61e-06 ***
## zn           -0.034007   0.038238  -0.889 0.373809
## indus        -0.038911   0.063663  -0.611 0.541067
## chas1         0.992021   0.895195   1.108 0.267792
## nox          59.617163   9.666229   6.168 6.93e-10 ***
## rm           -1.013799   1.609692  -0.630 0.528819
## age           0.077636   0.023288   3.334 0.000857 ***
## dis           0.584154   0.329319   1.774 0.076093 .
## rad           0.833147   0.210486   3.958 7.55e-05 ***
## tax          -0.006486   0.028887  -0.225 0.822345
## ptratio       0.650007   0.171031   3.801 0.000144 ***
## lstat         0.004030   0.069243   0.058 0.953585
## medv          0.244723   0.088042   2.780 0.005443 **
## tpr           0.003375   0.170373   0.020 0.984195
## adr          -0.086695   0.049251  -1.760 0.078365 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 518.38  on 373  degrees of freedom
## Residual deviance: 144.82  on 359  degrees of freedom
## AIC: 174.82
##
## Number of Fisher Scoring iterations: 9
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45  5
##          1  2 40
##
##                Accuracy : 0.9239
##                  95% CI : (0.8495, 0.9689)
##     No Information Rate : 0.5109
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8475
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9574
##             Specificity : 0.8889
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.9524
##              Prevalence : 0.5109
##          Detection Rate : 0.4891
##    Detection Prevalence : 0.5435
##       Balanced Accuracy : 0.9232
##
##        'Positive' Class : 0
##
```

Model 2 - Selection by P-Values and Multicollinearity

As described above, and as expected given known real-world relationships between things like proximity to highways and air pollution, there are several variable combinations with high collinearity:

- tax and rad (corr > 0.90 – very high!)

- nox and dis (corr > 0.76)
- age and dis (corr > 0.75)
- indus and nox (corr > 0.75)
- indus and tax (corr > 0.73)
- nox and age (corr > 0.73)
- indus and dis (cor > 0.70)
- medv and room (cor > 0.70)

One approach could be to simply drop tax as a feature as it is highly correlated with more than one other feature in the data set. Another option is linearly combine highly correlated variables into a single variable.

Before we begin stepwise addition and subtraction, let's also run a model with only variables with significance below 0.1 value. Conveniently, this approach also eliminates several of the suspect variable combinations; however we are still left with

- nox and age (corr > 0.73)

Age of homes should not be directly causally related to air quality for any conceivable reason, and so this may be a spurious association, like Nick Cage movies and swimming pool accidents. We will therefore leave that one in.

We expect that the previously strong features will persist but the weaker features may undergo interesting changes.

This model has a slightly lower accuracy than the full model, but it's very close – indicating that we didn't lose a lot of predictive value in reducing the feature set quite significantly.

```
##
## Call:
## stats::glm(formula = target ~ nox + age + rad + tax + ptratio +
##     medv + adr, family = "binomial", data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90885  -0.21094  -0.00231   0.00196   2.89843
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -46.895523   7.338000  -6.391 1.65e-10 ***
## nox          53.043824   8.441991   6.283 3.31e-10 ***
## age           0.080635   0.018880   4.271 1.95e-05 ***
## rad           0.818382   0.183438   4.461 8.14e-06 ***
## tax          -0.006563   0.003244  -2.023 0.043067 *
## ptratio       0.565174   0.139292   4.057 4.96e-05 ***
## medv          0.134581   0.039240   3.430 0.000604 ***
## adr          -0.137061   0.035426  -3.869 0.000109 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 518.38  on 373  degrees of freedom
## Residual deviance: 151.18  on 366  degrees of freedom
## AIC: 167.18
##
## Number of Fisher Scoring iterations: 9
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45  6
##          1  2 39
##
##                Accuracy : 0.913
##                  95% CI : (0.8358, 0.9617)
##     No Information Rate : 0.5109
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8257
##
##  Mcnemar's Test P-Value : 0.2888
##
##             Sensitivity : 0.9574
##             Specificity : 0.8667
##          Pos Pred Value : 0.8824
##          Neg Pred Value : 0.9512
##              Prevalence : 0.5109
##          Detection Rate : 0.4891
##    Detection Prevalence : 0.5543
##       Balanced Accuracy : 0.9121
##
##        'Positive' Class : 0
##
```

Model 3 - Backward Selection from Full Model

Next, let's use the AIC scoring mechanism to do stepwise subtraction. This will give us a subset of variables at a local maximum. Generally speaking, it's difficult to find an absolute maximum but random search helps. Therefore, we will keep this local maximum that we obtain from stepwise selection on the full model.

Interestingly, this provides us with the same model as Model 2!

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax + ptratio + medv +
##     adr, family = "binomial", data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90885  -0.21094  -0.00231   0.00196   2.89843
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -46.895523   7.338000  -6.391 1.65e-10 ***
## nox          53.043824   8.441991   6.283 3.31e-10 ***
## age           0.080635   0.018880   4.271 1.95e-05 ***
## rad           0.818382   0.183438   4.461 8.14e-06 ***
## tax          -0.006563   0.003244  -2.023 0.043067 *
## ptratio       0.565174   0.139292   4.057 4.96e-05 ***
## medv          0.134581   0.039240   3.430 0.000604 ***
## adr          -0.137061   0.035426  -3.869 0.000109 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 518.38  on 373  degrees of freedom
## Residual deviance: 151.18  on 366  degrees of freedom
## AIC: 167.18
##
## Number of Fisher Scoring iterations: 9
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45  6
##          1  2 39
##
##                Accuracy : 0.913
##                  95% CI : (0.8358, 0.9617)
##     No Information Rate : 0.5109
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8257
##
##  Mcnemar's Test P-Value : 0.2888
##
##             Sensitivity : 0.9574
##             Specificity : 0.8667
##          Pos Pred Value : 0.8824
##          Neg Pred Value : 0.9512
##              Prevalence : 0.5109
##          Detection Rate : 0.4891
##    Detection Prevalence : 0.5543
##       Balanced Accuracy : 0.9121
##
##        'Positive' Class : 0
##
```

Model 4 - Forward & Backward Stepwise from No Features

Finally, let's use that same AIC scoring mechanism to do forward stepwise addition. This will reduce some of the extra variables we added while still constraining ourselves to the most optimal fit. Notably, when we run forward selection on our previous model, it reaches the same step previously and stops. Therefore, let's instead run it from a no feature model. As shown below, this is the same resulting model with one feature, nox.

```
##
## Call:
## stats::glm(formula = target ~ nox + age + rad + tax + ptratio +
##     medv + adr, family = "binomial", data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90885  -0.21094  -0.00231   0.00196   2.89843
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -46.895523   7.338000  -6.391 1.65e-10 ***
## nox          53.043824   8.441991   6.283 3.31e-10 ***
## age           0.080635   0.018880   4.271 1.95e-05 ***
## rad           0.818382   0.183438   4.461 8.14e-06 ***
## tax          -0.006563   0.003244  -2.023 0.043067 *
## ptratio       0.565174   0.139292   4.057 4.96e-05 ***
## medv          0.134581   0.039240   3.430 0.000604 ***
## adr          -0.137061   0.035426  -3.869 0.000109 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 518.38  on 373  degrees of freedom
## Residual deviance: 151.18  on 366  degrees of freedom
## AIC: 167.18
##
## Number of Fisher Scoring iterations: 9
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45  6
##          1  2 39
##
##                Accuracy : 0.913
##                  95% CI : (0.8358, 0.9617)
##     No Information Rate : 0.5109
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8257
##
##  Mcnemar's Test P-Value : 0.2888
##
##             Sensitivity : 0.9574
##             Specificity : 0.8667
##          Pos Pred Value : 0.8824
##          Neg Pred Value : 0.9512
##              Prevalence : 0.5109
```

```
##            Detection Rate : 0.4891
##      Detection Prevalence : 0.5543
##         Balanced Accuracy : 0.9121
##
##          'Positive' Class : 0
##
```

Model Selection

We previously realized that our models beat the null hypothesis so now we have some potentially successful models. To compare logistical models, we can assess accuracy, precision, deviance, AIC, and so on. We will also consider practicalities – all else being equal, a model wiht fewer features is cheaper to run and easier to explian and maintain.

It's not clear whether we should prefer precision or specificity here so we will use the derived F1 metric.

First, we will do chi square testing to prove the statistical validity of the models and then look at F1 to replace old models.

Accordingly, the residual deviance for the full model is 172. The next two models have 0.34 and 0.066 significance value. Since our alpha value is 0.05, these two models are not significant enough to replace the old model. The last model's p value is low enough to be relevant but the residual deviance is too high to warrant replacing it. Therefore, model 1 is our final model.

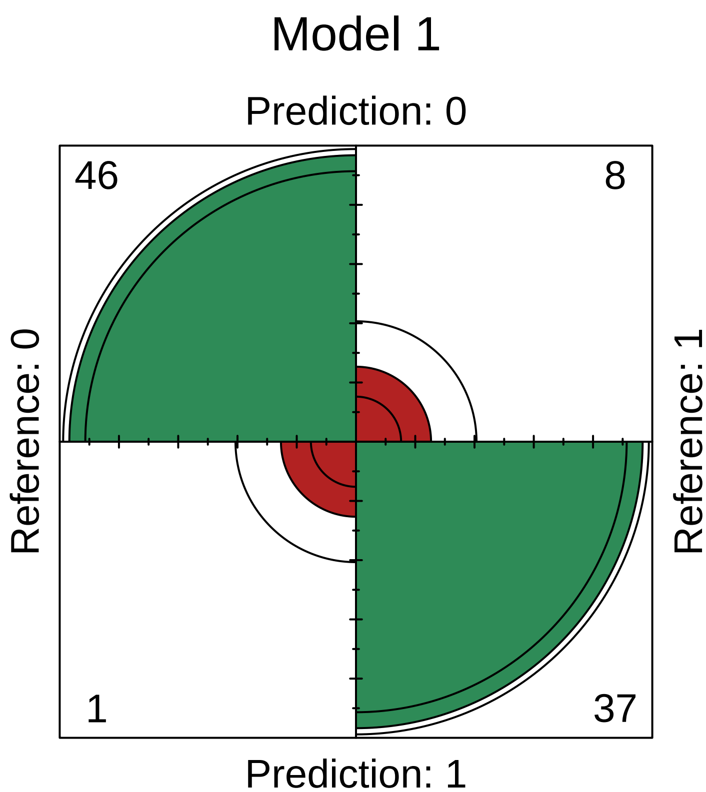
```
## Analysis of Deviance Table
##
## Model 1: target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv + tpr + adr
## Model 2: target ~ nox + age + rad + tax + ptratio + medv + adr
## Model 3: target ~ nox + age + rad + tax + ptratio + medv + adr
## Model 4: target ~ nox + age + rad + tax + ptratio + medv + adr
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       359     144.81
## 2       366     151.18 -7  -6.3637    0.498
## 3       366     151.18  0   0.0000
## 4       366     151.18  0   0.0000
```

These are the statistics for all of the models in case we were curious. We would use this table to verify the F1 value if a new model were to pass the previous tests.

```
##             F1  Accuracy TP FP FN TN
## [1,] 0.9108911 0.9021739 46  8  1 37
## [2,] 0.9108911 0.9021739 46  8  1 37
## [3,] 0.9108911 0.9021739 46  8  1 37
## [4,] 0.9108911 0.9021739 46  8  1 37
```

Because the predicted values are ultimately the same across models, the selected model is the one with fewer features, which is Model 2.

Shown below is a tabulated form and visualization of our final model.




```
## target ~ nox + age + rad + tax + ptratio + medv + adr
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 46  8
##          1  1 37
##
##                Accuracy : 0.9022
##                  95% CI : (0.8224, 0.9543)
##     No Information Rate : 0.5109
##     P-Value [Acc > NIR] : 9.642e-16
##
##                   Kappa : 0.8036
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.9787
##             Specificity : 0.8222
##          Pos Pred Value : 0.8519
##          Neg Pred Value : 0.9737
##               Precision : 0.8519
##                  Recall : 0.9787
##                      F1 : 0.9109
##              Prevalence : 0.5109
##          Detection Rate : 0.5000
```

```
##      Detection Prevalence : 0.5870
##         Balanced Accuracy : 0.9005
##
##          'Positive' Class : 0
##
## NULL
```

Conclusions and Final Thoughts

In the real world, accuracy, specificity, and sensitivity are all factors worth considering, but so, too, is the cost to gather and host input data and the processing time to run complex models. Because these factors are important, we have selected a simpler model than one which provides slightly superior results.

All of our models highly select for nitrogen oxide concentration (nox) so we assume that it is a byproduct of population. Additional indicators of poverty are a low number of rooms per dwelling (rm) and apparently distance to an accessible highway (rad) is also another. Part of our responsibility to our community is giving the proper equity so that logistical means for garbage cleaning and public transportation are accounted for.

References

Datasets are provided by CUNY School of Professional Studies for academic purposes. It is reflective of public data gathered online.

Appendix

Shown here is a copy of all relevant R code.

```
MakePredictions
```

```
## function(model, testData, excludeCol = "target", threshold = 0.5) {
##   probability = stats::predict(model, testData[, ! colnames(testData) %in% excludeCol])
##   class = probability %>%
##     { .[. > threshold] = 1 ; . } %>%
##       { .[. <= threshold] = 0 ; . } %>%
##         as.factor(.)
##
##   data.frame(class, probability)
## }
## <bytecode: 0x000000002eab7e30>
```

```
VerifyKfold
```

```
## function(model, testData, excludeCol = "target", threshold = 0.5) {
##   MakePredictions(model, testData, excludeCol, threshold) %>%
##     { caret::confusionMatrix(.$class, testData[, colnames(testData) %in% excludeCol], m
## }
## <bytecode: 0x000000002f5095e0>
```

```
ValidationPipeline
```

```
## function(model, testData, excludeCol, plotname = "", threshold = 0.5) {
##   p = MakePredictions(model, testData, excludeCol)
##   d = VerifyKfold(model, testData, excludeCol)
##   g = graphics::fourfoldplot(d$table, color = c("#B22222", "#2E8B57"), main = plotname)
##
##   print(formula(model))
##   print(d)
##   print(g)
##
##   invisible(list(p, d, g))
## }
```

PlotCorrEllipse

```
## function(corData, pal = RColorBrewer::brewer.pal(5, "Spectral"),
##                             highlight = c("both", "positive", "negative")[1], hiMod = 1)
##   colorRange = 100
##   coloredVals = corData*50 + 50
##   skewRange = max(min(hiMod[[1]], 2), 0)
##
##   if (is.numeric(pal)) {
##     warning("Passed number as argument for palette. It works but was this intentional?")
##   }
##   if (skewRange != hiMod) {
##     warning("Color range skew only allowed between 0 and 2 where 0.5*n% of value range a
cor(X) C [-1, 1]")
##   }
##   if (highlight[[1]] == "positive") {
##     colorRange = 50 * skewRange
##     coloredVals = 1 - corData*50 + 50
##   }
##   if (highlight[[1]] == "negative") {
##     colorRange = 50 * skewRange
##     coloredVals = corData*50 + 50
##   }
##
##   ellipse::plotcorr(corData, mar = c(1,1,1,1),
##                     col = grDevices::colorRampPalette(pal)(colorRange)[coloredVals])
##
## }
## <bytecode: 0x0000000030e50080>
```