

DATA621 HW3

Matthew Baker, Misha Kollontai, Erinda Budo, Don Padmaperuma, Subhalaxmi Rout

10/21/2020

Overview

In this homework assignment, we will build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels. We will provide classifications and probabilities for the evaluation data set using our binary logistic regression model. 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**)

Libraries

Data Import

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.70	50.0	1
0	19.58	1	0.871	5.403	100.0	1.3216	5	403	14.7	26.82	13.4	1
0	18.10	0	0.740	6.485	100.0	1.9784	24	666	20.2	18.85	15.4	1
30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7	0
0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9	0
0	8.56	0	0.520	6.781	71.3	2.8561	5	384	20.9	7.67	26.5	0
0	18.10	0	0.693	5.453	100.0	1.4896	24	666	20.2	30.59	5.0	1
0	18.10	0	0.693	4.519	100.0	1.6582	24	666	20.2	36.98	7.0	1
0	5.19	0	0.515	6.316	38.1	6.4584	5	224	20.2	5.68	22.2	0
80	3.64	0	0.392	5.876	19.1	9.2203	1	315	16.4	9.25	20.9	0
22	5.86	0	0.431	6.438	8.9	7.3967	7	330	19.1	3.59	24.8	0
0	12.83	0	0.437	6.286	45.0	4.5026	5	398	18.7	8.94	21.4	0
0	18.10	0	0.532	7.061	77.0	3.4106	24	666	20.2	7.01	25.0	1
22	5.86	0	0.431	8.259	8.4	8.9067	7	330	19.1	3.54	42.8	1
0	2.46	0	0.488	6.153	68.8	3.2797	3	193	17.8	13.15	29.6	0

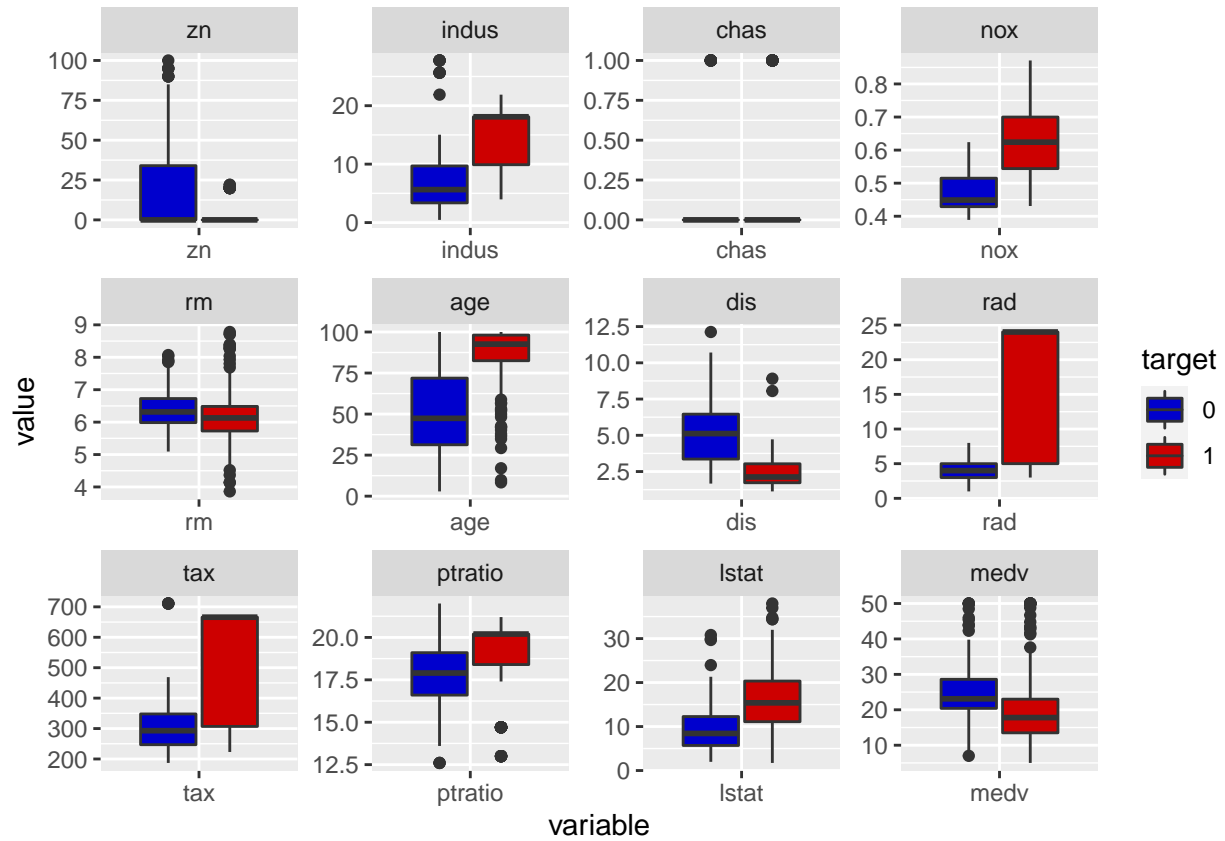
Data Exploration

Let's calculate summary statistics and generate a box plot for further review.

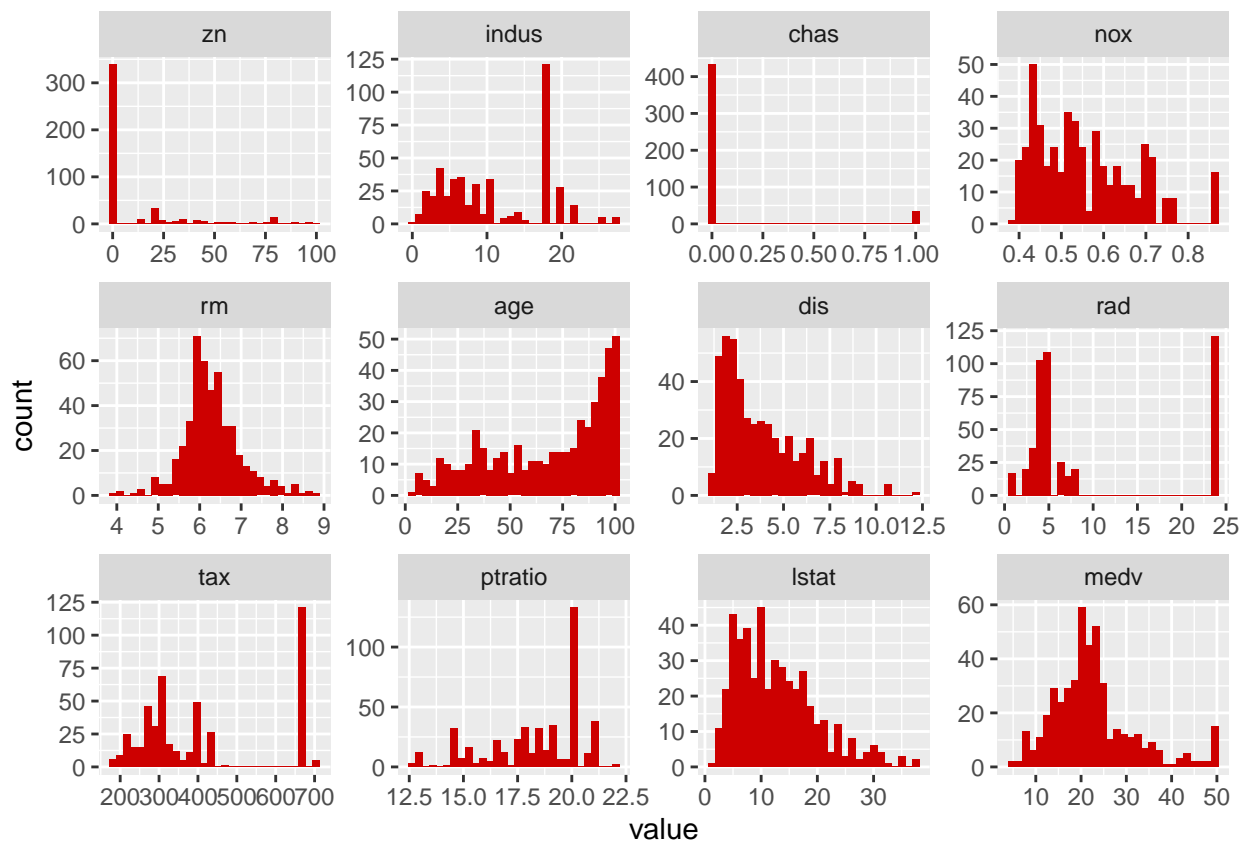
	vars	n	mean	sd	median	trimmed	mad	min	max	ra
zn	1	466	11.5772532	23.3646511	0.00000	5.3542781	0.0000000	0.0000	100.0000	100.0
indus	2	466	11.1050215	6.8458549	9.69000	10.9082353	9.3403800	0.4600	27.7400	27.2
chas	3	466	0.0708155	0.2567920	0.00000	0.0000000	0.0000000	0.0000	1.0000	1.0
nox	4	466	0.5543105	0.1166667	0.53800	0.5442684	0.1334340	0.3890	0.8710	0.4
rm	5	466	6.2906738	0.7048513	6.21000	6.2570615	0.5166861	3.8630	8.7800	4.9
age	6	466	68.3675966	28.3213784	77.15000	70.9553476	30.0226500	2.9000	100.0000	97.1
dis	7	466	3.7956929	2.1069496	3.19095	3.5443647	1.9144814	1.1296	12.1265	10.9
rad	8	466	9.5300429	8.6859272	5.00000	8.6978610	1.4826000	1.0000	24.0000	23.0
tax	9	466	409.5021459	167.9000887	334.50000	401.5080214	104.5233000	187.0000	711.0000	524.0
ptratio	10	466	18.3984979	2.1968447	18.90000	18.5970588	1.9273800	12.6000	22.0000	9.4
lstat	11	466	12.6314592	7.1018907	11.35000	11.8809626	7.0720020	1.7300	37.9700	36.2
medv	12	466	22.5892704	9.2396814	21.20000	21.6304813	6.0045300	5.0000	50.0000	45.0
target	13	466	0.4914163	0.5004636	0.00000	0.4893048	0.0000000	0.0000	1.0000	1.0

There are 466 records in our training set and no missing values for any variable. We see no missing values that would require imputation using medians or other methods.

Now let's visualize using box plots. We are going to separate the box plots by the target value, which will tell if the neighborhood is high crime or not.



In order to check for skewness, we will examine the distribution of each variable independent of target variable value.



In particular, zn, nox, age, dis, ptratio, and lstat seem likely candidates for transformations.

Let's check for covariance.

```
zn
indus
chas
nox
rm
age
dis
rad
tax
ptratio
lstat
medv
target
zn
1.0000000
-0.5382664
```

-0.0401620
-0.5170452
0.3198141
-0.5725805
0.6601243
-0.3154812
-0.3192841
-0.3910357
-0.4329925
0.3767171
-0.4316818
indus
-0.5382664
1.0000000
0.0611832
0.7596301
-0.3927118
0.6395818
-0.7036189
0.6006284
0.7322292
0.3946898
0.6071102
-0.4961743
0.6048507
chas
-0.0401620
0.0611832
1.0000000
0.0974558
0.0905098
0.0788837
-0.0965771
-0.0159004
-0.0467648
-0.1286606

-0.0514232

0.1615653

0.0800419

nox

-0.5170452

0.7596301

0.0974558

1.0000000

-0.2954897

0.7351278

-0.7688840

0.5958298

0.6538780

0.1762687

0.5962426

-0.4301227

0.7261062

rm

0.3198141

-0.3927118

0.0905098

-0.2954897

1.0000000

-0.2328125

0.1990158

-0.2084457

-0.2969343

-0.3603471

-0.6320245

0.7053368

-0.1525533

age

-0.5725805

0.6395818

0.0788837

0.7351278

-0.2328125
1.0000000
-0.7508976
0.4603143
0.5121245
0.2554479
0.6056200
-0.3781560
0.6301062
dis
0.6601243
-0.7036189
-0.0965771
-0.7688840
0.1990158
-0.7508976
1.0000000
-0.4949919
-0.5342546
-0.2333394
-0.5075280
0.2566948
-0.6186731
rad
-0.3154812
0.6006284
-0.0159004
0.5958298
-0.2084457
0.4603143
-0.4949919
1.0000000
0.9064632
0.4714516
0.5031013
-0.3976683

0.6281049

tax

-0.3192841

0.7322292

-0.0467648

0.6538780

-0.2969343

0.5121245

-0.5342546

0.9064632

1.0000000

0.4744223

0.5641886

-0.4900329

0.6111133

ptratio

-0.3910357

0.3946898

-0.1286606

0.1762687

-0.3603471

0.2554479

-0.2333394

0.4714516

0.4744223

1.0000000

0.3773560

-0.5159153

0.2508489

lstat

-0.4329925

0.6071102

-0.0514232

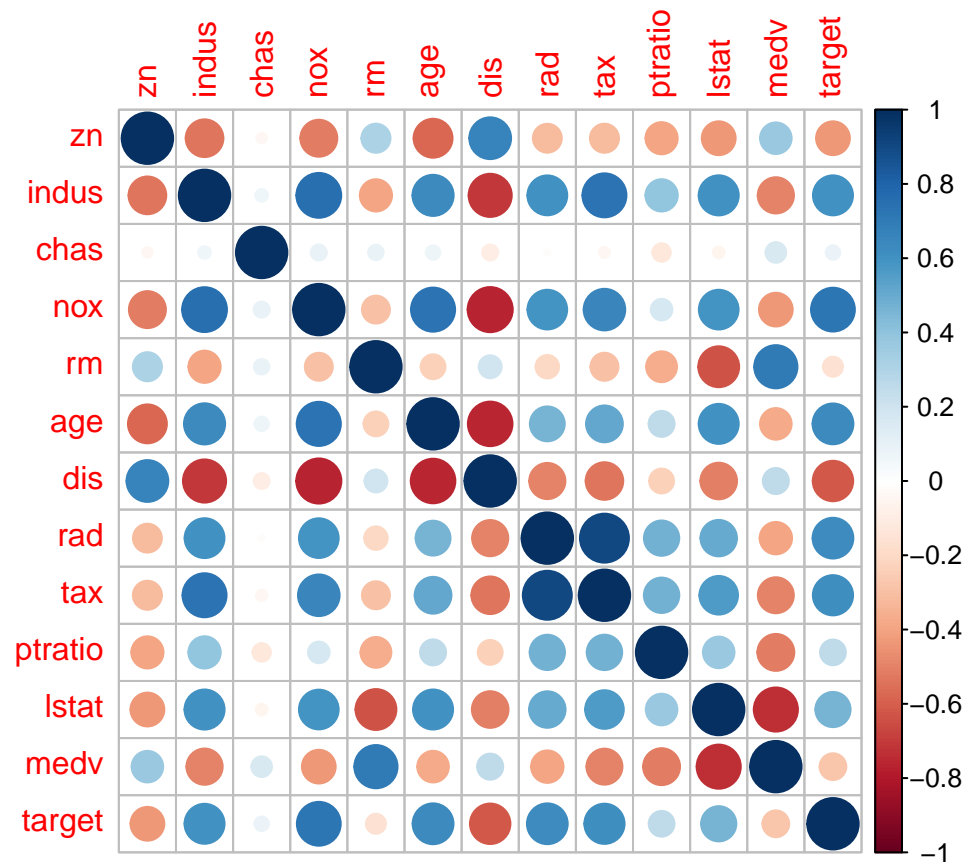
0.5962426

-0.6320245

0.6056200

-0.5075280
0.5031013
0.5641886
0.3773560
1.0000000
-0.7358008
0.4691270
medv
0.3767171
-0.4961743
0.1615653
-0.4301227
0.7053368
-0.3781560
0.2566948
-0.3976683
-0.4900329
-0.5159153
-0.7358008
1.0000000
-0.2705507
target
-0.4316818
0.6048507
0.0800419
0.7261062
-0.1525533
0.6301062
-0.6186731
0.6281049
0.6111133
0.2508489
0.4691270
-0.2705507
1.0000000

We see some very high positive and negative correlations between variables. Let's construct a more effective visualization.



We see candidates for combination due to covariance.

As a final step, let's look just a correlation between the independent variables and the target variables.

target

zn

-0.4316818

indus

0.6048507

chas

0.0800419

nox

0.7261062

rm

-0.1525533

age

0.6301062

dis

-0.6186731

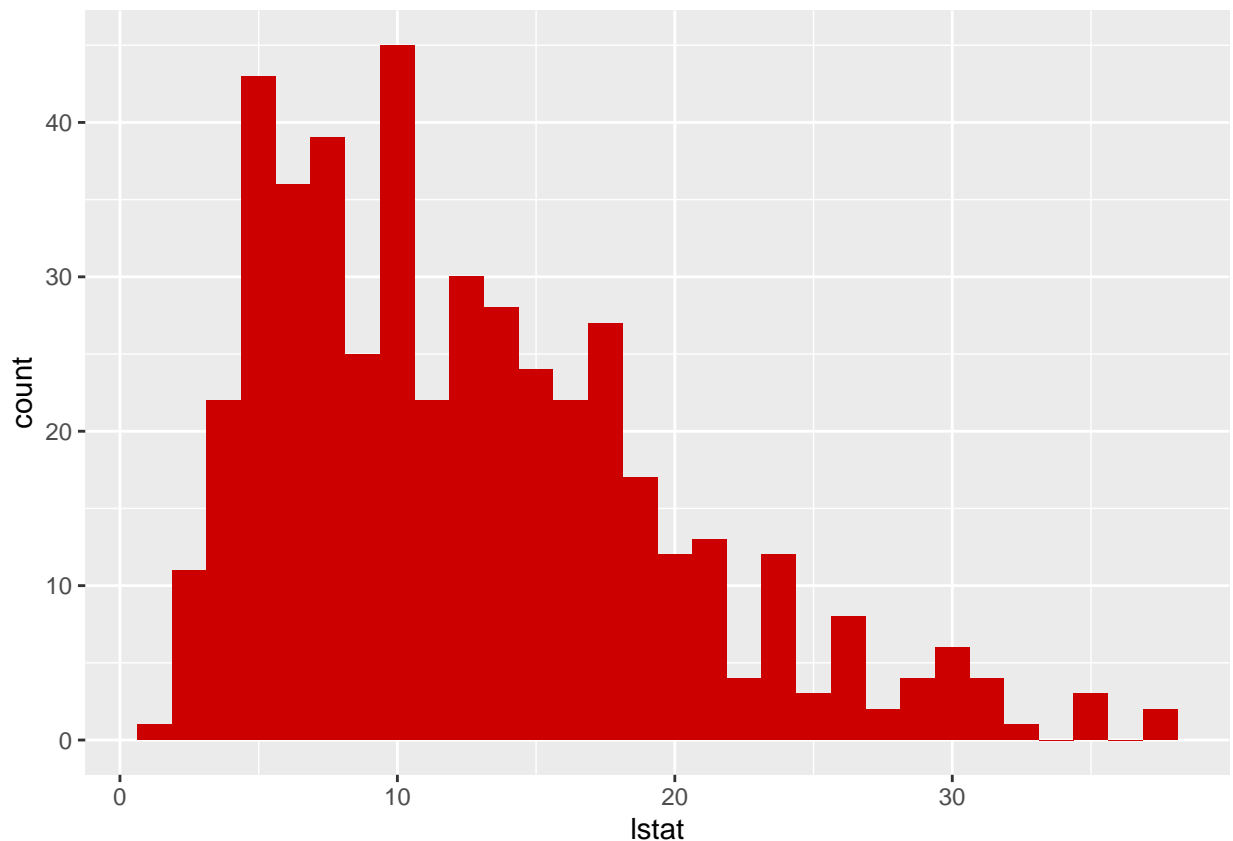
rad

```
0.6281049
tax
0.6111133
ptratio
0.2508489
lstat
0.4691270
medv
-0.2705507
target
1.0000000
```

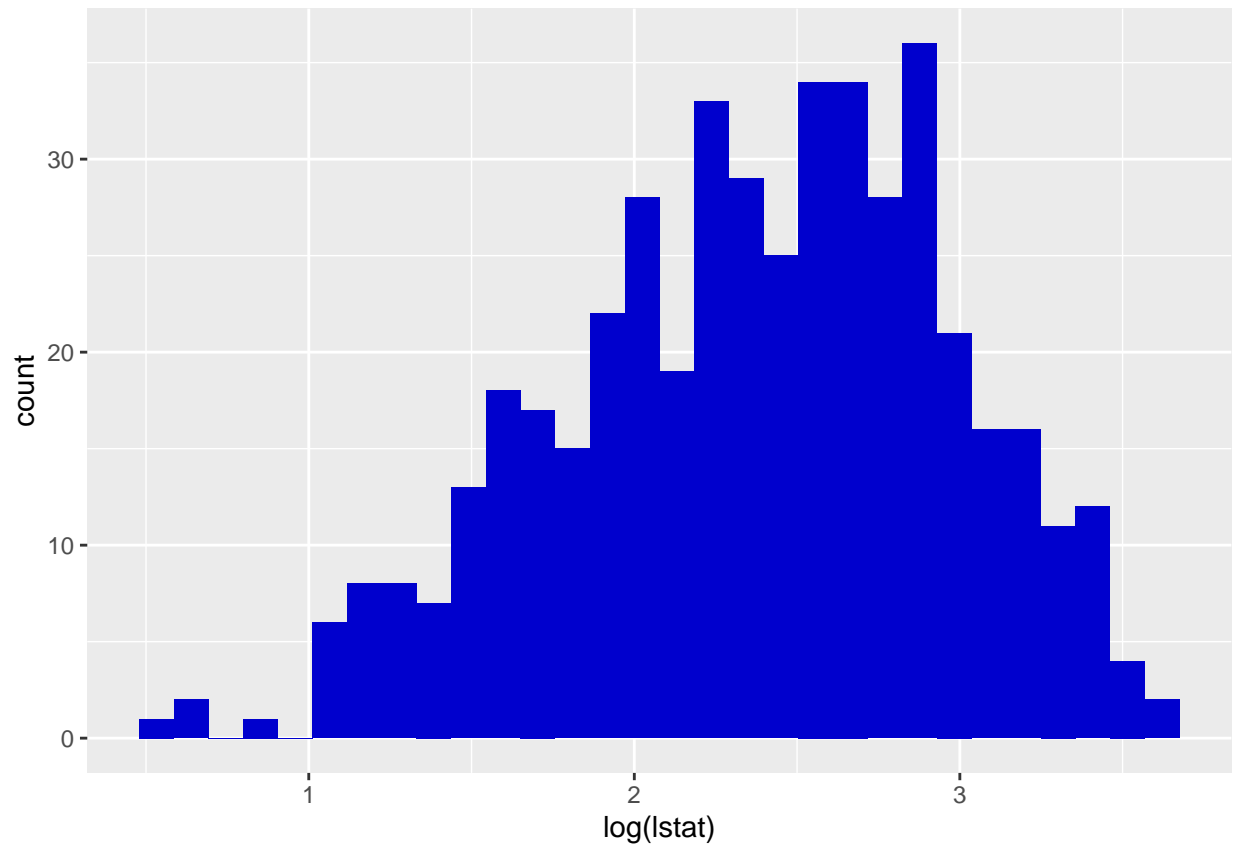
We see that Nox(nitrogen oxides concentration), shows the closest correlation with the target variable at .73. Next, age, rad, tax, and indus all correlate with the target value just above .6.

Data Preparation

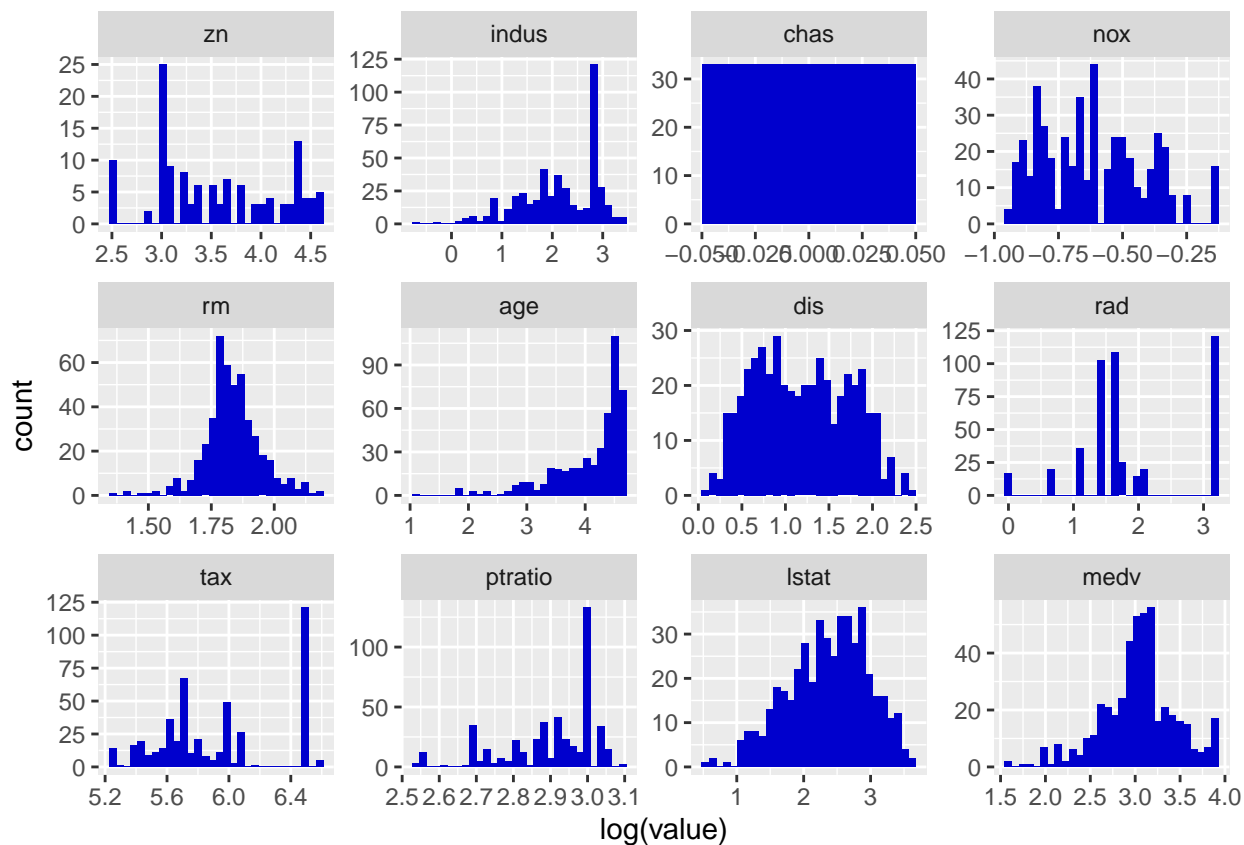
Let's look at how transformations might solve distribution issues with some of our variables. Earlier, we saw a strong right skew in the distribution of the variable lstat, which tracks the “lower status” of a neighborhood's population. Probably not the best phrasing.



What would a log transformation do to this distribution?

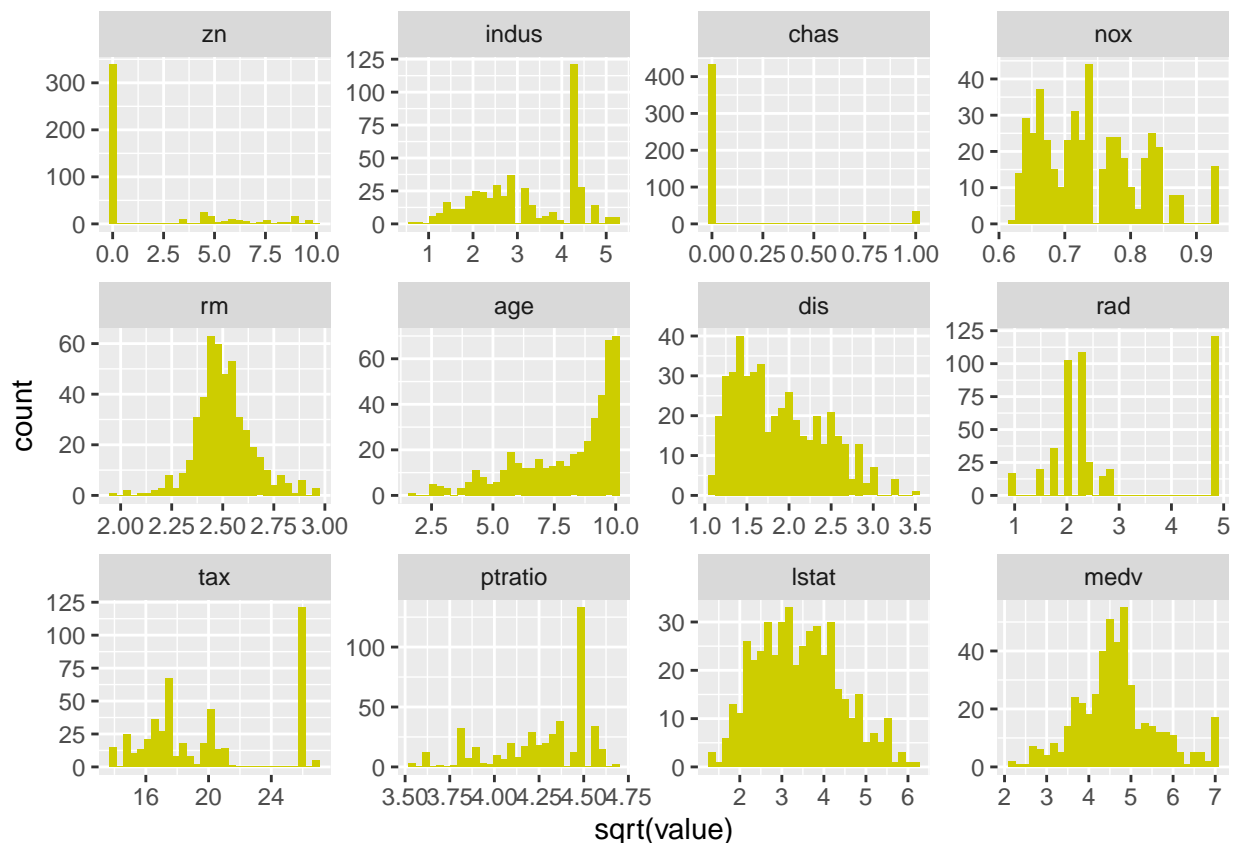


Looks slightly better. Let's generate log transformations for all variables in the dataset.



Medv looks slightly better. However, age remains strongly left skewed. Dis is now bimodal.

What about other transformations such as quadratic ones?



Not a lot of improvement.

Let's split our training data into a true training set and a validation set. We'll go 80/20 training to validation.

Build Models

We will start with a model containing all untransformed variables.

Model 1 - All Variables Untransformed

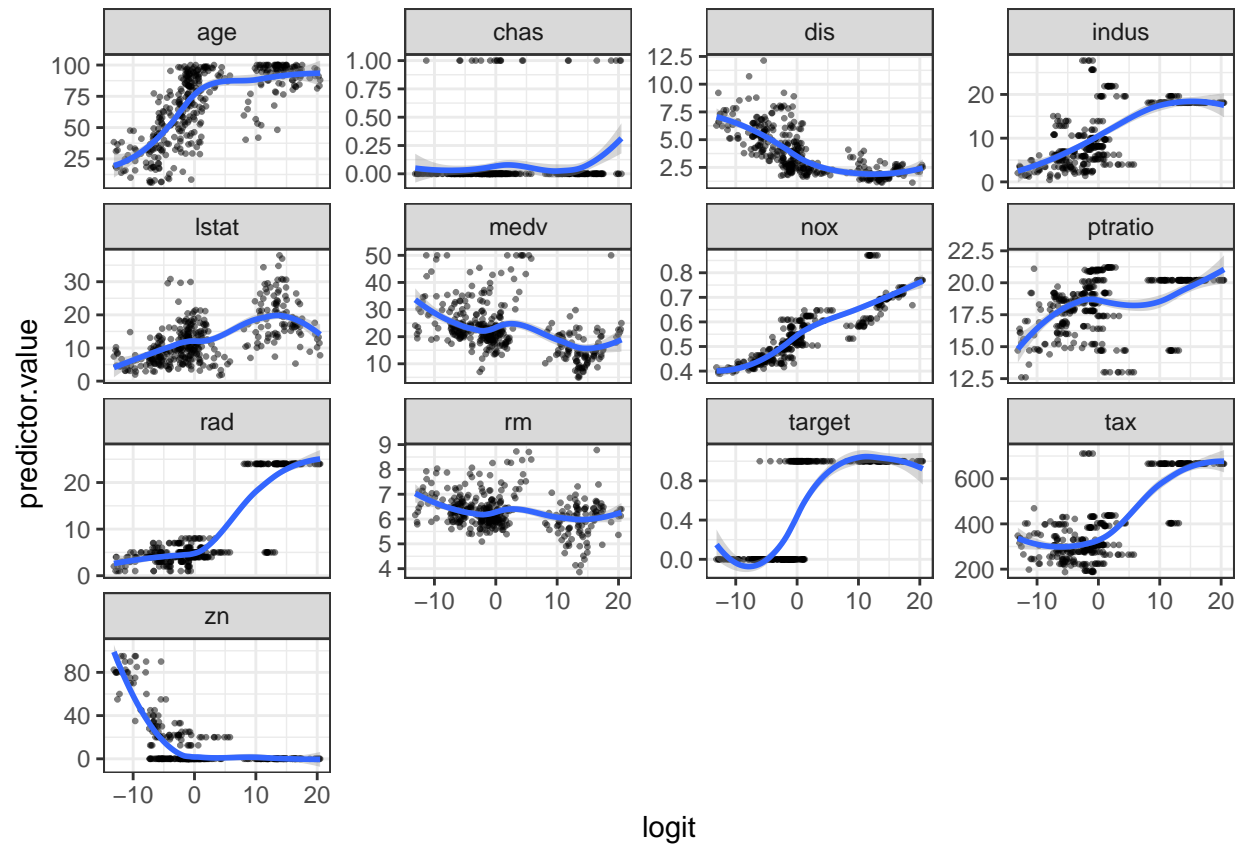
```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = df_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7382  -0.2855  -0.0054   0.0040   3.4853
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.553696   7.376105  -5.498 3.84e-08 ***
## zn          -0.066245   0.037617  -1.761 0.07824 .
## indus       -0.061615   0.051686  -1.192 0.23321
## chas         0.146507   0.881580   0.166 0.86801
## nox         48.484863   8.483921   5.715 1.10e-08 ***
```

```
## rm          -0.210853   0.814385  -0.259  0.79570
## age          0.030010   0.014946   2.008  0.04465 *
## dis          0.751881   0.240194   3.130  0.00175 **
## rad          0.583748   0.179135   3.259  0.00112 **
## tax         -0.004762   0.003069  -1.551  0.12078
## ptratio      0.332626   0.133255   2.496  0.01255 *
## lstat        0.010455   0.062146   0.168  0.86639
## medv         0.159637   0.077352   2.064  0.03904 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 514.15  on 371  degrees of freedom
## Residual deviance: 164.56  on 359  degrees of freedom
## AIC: 190.56
##
## Number of Fisher Scoring iterations: 9
```

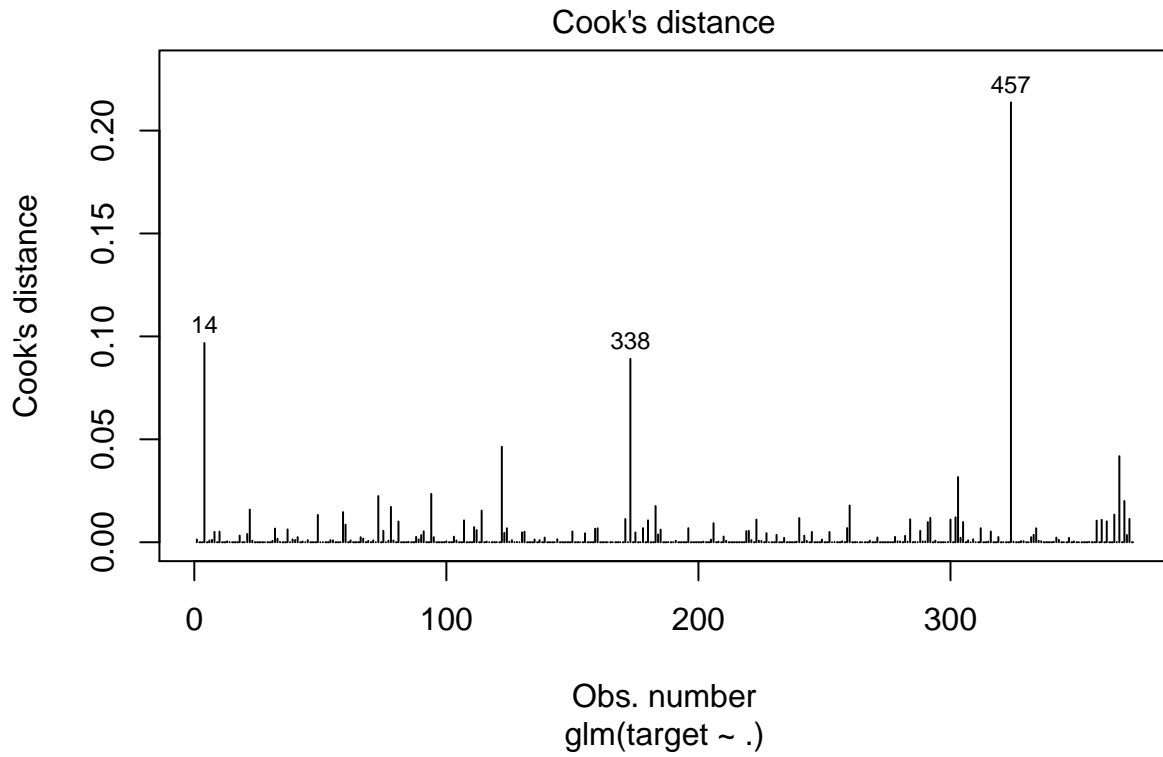
Our most significant variables generally tie to the variables we saw have the highest correlations with the target value earlier. We have an AIC of 190.56 and a residual deviance of 164.56.

Let's run further diagnostics on the model. We will set a probability of .5 as being the cutoff for determining if a neighborhood will be high crime. Here, we check the relationship between the logit of the outcome and each predictive variable. (Target and the binary dummy variable chas should be ignored.) Again, these steps also could be labelled as data preparation.

```
##      415      463      179      14      195      426
## "neg" "pos" "pos" "neg" "pos" "neg"
```



Let's use Cook's Distance to check for outliers.



.rownames	target	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	.fitted
14	1	22	5.86	0	0.431	8.259	8.4	8.9067	7	330	19.1	3.54	42.8	-0.530255
338	1	20	6.96	0	0.464	5.856	42.1	4.4290	3	223	18.6	13.00	21.1	-6.071232
457	1	0	10.59	0	0.489	5.412	9.8	3.5875	4	277	18.6	29.55	23.7	-4.351593

In particular, let's look at the points that are more than 3 standardized residuals from 0.

.rownames	target	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	.fitted
338	1	20	6.96	0	0.464	5.856	42.1	4.4290	3	223	18.6	13.00	21.1	-6.071232
457	1	0	10.59	0	0.489	5.412	9.8	3.5875	4	277	18.6	29.55	23.7	-4.351594

Observation 338 is an influential outlier.

Next, we check multicollinearity.

```
##          zn      indus      chas      nox      rm      age      dis      rad
## 1.962172 2.847931 1.279593 4.266650 4.984121 2.674408 4.013695 1.988754
##      tax ptratio  lstat  medv
## 2.150815 2.100852 2.457363 6.781834
```

Medv has vif greater than 5 which means high amount of multicollinearity.

So we have:

1. Multiple predictors that do not have linear relationships with the logic of the outcome variable.

2. One influential outlier - index 338.
3. One with potentially problematically high multicollinearity.

Model 2 - Remove least impactful variables

The second model will only use the variables that have statistically significant p-values. We will also use some transformations for the variables where they improved the distribution.

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + log(dis) + rad +
##      ptratio + medv, family = "binomial", data = df_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7035  -0.3090  -0.0076   0.0054   3.5399
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -47.50578     7.75642  -6.125 9.09e-10 ***
## zn          -0.06107     0.03326  -1.836 0.066347 .
## indus       -0.05845     0.04902  -1.192 0.233159
## nox         51.96792     8.60641   6.038 1.56e-09 ***
## age         0.03296     0.01240   2.659 0.007841 **
## log(dis)     4.07456     0.97460   4.181 2.91e-05 ***
## rad         0.50675     0.15139   3.347 0.000816 ***
## ptratio     0.32852     0.11905   2.760 0.005788 **
## medv        0.17417     0.04471   3.895 9.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 514.15  on 371  degrees of freedom
## Residual deviance: 160.35  on 363  degrees of freedom
## AIC: 178.35
##
## Number of Fisher Scoring iterations: 9
```

Removing the 4 least impactful variables as well as applying a log transformation to *dis* has moved our AIC value from 190.6 to 178.4. Not a significant improvement, but a step in the right direction. The number of variables is lower, with a negligible drop in value of the prediction. This change is something we'd look to keep as the added simplicity adds more than the missing variables provided.

Model 3

Let's see what kind of model we would get if we selected only the somewhat normally distributed variables - *indus*, *rm*, *lstat*, *medv*

```
##
## Call:
## glm(formula = target ~ indus + rm + lstat + medv, family = "binomial",
```

```

##      data = df_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8806  -0.6681  -0.4265   0.6505   2.2355
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.93880    1.86354  -5.333 9.64e-08 ***
## indus        0.19053    0.02630   7.243 4.38e-13 ***
## rm           0.86049    0.30375   2.833 0.00461 **
## lstat        0.14244    0.03293   4.325 1.52e-05 ***
## medv         0.02178    0.02699   0.807 0.41976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 514.15  on 371  degrees of freedom
## Residual deviance: 347.06  on 367  degrees of freedom
## AIC: 357.06
##
## Number of Fisher Scoring iterations: 4

```

We see a dramatic increase in the AIC and a doubling of the residual deviance. Clearly looking at these variables alone does not paint a good picture of what we are trying to do.

Selecting the Model

Now that we have created a few models, let us select one and evaluate it against our known predictor variable. Model 2 was the most promising, so let's dive into that one. We will then calculate various metrics for this model:

1. Accuracy
2. Classification Error Rate
3. Precision
4. Sensitivity
5. Specificity
6. F1 score
7. AUC
8. Confusion Matrix

We will then finally make predictions using the data set.

To start, we must calculate the predictions of our model:

```

##      model actual guess
## 415 1.231689e-01      0      0
## 463 9.999970e-01      1      1
## 179 9.999967e-01      1      1

```

## 14	4.182653e-01	1	0
## 195	9.999897e-01	1	1
## 426	8.268266e-02	0	0
## 306	9.847468e-02	0	0
## 118	2.563587e-01	1	0
## 299	4.044724e-03	0	0
## 229	6.336271e-01	1	1
## 244	9.999999e-01	1	1
## 465	7.869284e-03	0	0
## 374	5.045953e-02	0	0
## 153	2.603806e-02	0	0
## 90	2.470638e-03	0	0
## 91	2.337386e-03	0	0
## 256	9.999959e-01	1	1
## 197	2.419171e-01	0	0
## 451	9.999877e-01	1	1
## 441	6.393918e-03	0	0
## 348	3.242919e-01	0	0
## 137	3.906406e-01	0	0
## 355	2.190204e-01	0	0
## 328	7.940905e-06	0	0
## 26	8.008334e-06	0	0
## 7	9.999941e-01	1	1
## 461	9.999999e-01	1	1
## 445	8.865142e-03	0	0
## 254	2.838663e-05	0	0
## 211	9.999931e-01	1	1
## 420	9.279938e-02	0	0
## 78	2.168909e-01	0	0
## 81	2.784046e-01	0	0
## 43	4.738145e-02	0	0
## 359	9.999417e-01	1	1
## 373	9.999885e-01	1	1
## 332	3.273390e-01	0	0
## 143	9.939325e-01	1	1
## 32	2.122978e-01	0	0
## 109	7.961454e-01	1	1
## 263	7.077589e-01	1	1
## 393	9.999999e-01	1	1
## 330	4.807555e-02	0	0
## 23	9.187993e-03	0	0
## 411	3.170748e-01	0	0
## 309	3.187285e-03	0	0
## 135	2.723785e-05	0	0
## 421	2.340407e-04	0	0
## 422	5.743592e-01	1	1
## 224	9.956742e-01	1	1
## 166	3.082900e-02	0	0
## 217	9.999801e-01	1	1
## 290	9.999989e-01	1	1
## 69	8.461204e-01	1	1
## 72	1.805114e-01	0	0
## 76	6.922416e-04	0	0
## 63	9.999999e-01	1	1

## 141	6.001034e-03	0	0
## 210	3.420661e-01	0	0
## 353	1.308910e-01	0	0
## 347	9.999929e-01	1	1
## 453	6.115149e-02	0	0
## 294	4.635788e-03	0	0
## 277	9.999999e-01	1	1
## 41	5.117245e-03	0	0
## 452	7.146616e-01	1	1
## 316	3.562882e-01	0	0
## 223	2.753458e-03	0	0
## 16	1.310081e-01	0	0
## 116	4.507820e-06	0	0
## 94	2.397886e-01	0	0
## 262	1.567392e-02	0	0
## 235	5.448714e-01	0	1
## 86	1.199734e-03	0	0
## 342	2.432238e-01	0	0
## 39	1.990401e-02	0	0
## 159	1.885854e-01	0	0
## 240	5.680438e-01	0	1
## 209	9.199569e-02	0	0
## 454	9.999962e-01	1	1
## 460	4.340517e-01	1	0
## 34	9.999979e-01	1	1
## 4	1.420393e-03	0	0
## 13	9.999430e-01	1	1
## 413	6.282705e-02	0	0
## 243	9.999916e-01	1	1
## 308	1.000000e+00	1	1
## 278	7.428488e-01	1	1
## 89	3.539272e-01	0	0
## 25	4.105794e-01	1	0
## 291	6.764123e-01	1	1
## 368	9.999964e-01	1	1
## 286	9.998116e-01	1	1
## 390	3.964700e-01	1	0
## 121	6.334937e-01	1	1
## 110	6.835852e-03	0	0
## 158	5.077755e-05	0	0
## 64	3.849335e-04	0	0
## 199	9.999065e-01	1	1
## 67	1.094581e-01	0	0
## 151	1.000000e+00	1	1
## 335	9.999978e-01	1	1
## 85	9.044435e-01	1	1
## 165	8.135995e-02	0	0
## 136	3.430643e-03	0	0
## 51	4.736323e-06	0	0
## 74	7.313640e-01	0	1
## 178	2.483233e-03	0	0
## 236	1.913345e-05	0	0
## 98	9.999228e-01	1	1
## 424	4.856801e-01	1	0

## 214	9.499709e-02	0	0
## 127	9.999859e-01	1	1
## 212	6.878370e-01	0	1
## 174	6.260515e-04	0	0
## 273	9.351037e-01	1	1
## 302	3.803253e-02	0	0
## 310	3.534005e-03	0	0
## 232	1.000000e+00	1	1
## 381	4.637447e-02	0	0
## 365	1.000000e+00	1	1
## 280	4.361056e-02	1	0
## 113	4.473325e-01	0	0
## 107	3.326489e-01	0	0
## 366	9.407574e-02	0	0
## 154	1.466003e-01	0	0
## 102	6.915393e-02	0	0
## 255	9.568255e-01	1	1
## 160	9.933416e-01	1	1
## 155	6.199642e-01	1	1
## 5	2.256300e-01	0	0
## 326	1.346678e-03	0	0
## 272	9.999730e-01	1	1
## 345	2.763313e-03	0	0
## 288	1.880504e-01	0	0
## 403	9.999807e-01	1	1
## 55	7.288395e-01	1	1
## 238	4.912760e-03	0	0
## 252	1.362837e-01	0	0
## 391	8.062089e-04	0	0
## 349	5.010343e-03	0	0
## 439	1.895357e-04	0	0
## 226	4.649900e-03	0	0
## 48	4.379653e-01	0	0
## 77	9.999900e-01	1	1
## 83	9.999992e-01	1	1
## 184	3.303039e-04	0	0
## 327	1.219027e-04	0	0
## 196	3.668028e-03	0	0
## 257	1.183891e-01	0	0
## 168	9.999977e-01	1	1
## 387	9.999854e-01	1	1
## 396	8.291856e-04	0	0
## 334	8.953251e-05	0	0
## 346	5.587465e-01	1	1
## 20	9.999937e-01	1	1
## 462	9.999998e-01	1	1
## 164	3.769699e-06	0	0
## 52	4.592706e-01	0	0
## 22	6.835917e-01	1	1
## 177	9.999999e-01	1	1
## 42	9.999926e-01	1	1
## 84	2.889852e-05	0	0
## 11	1.469514e-02	0	0
## 372	2.113396e-03	0	0

## 364	1.030888e-02	0	0
## 194	9.999999e-01	1	1
## 343	9.949088e-04	0	0
## 322	9.934503e-01	1	1
## 198	1.000000e+00	1	1
## 249	1.991575e-01	1	0
## 200	9.713768e-01	1	1
## 338	1.900596e-03	1	0
## 250	7.471437e-04	0	0
## 292	4.500874e-01	1	0
## 398	9.999988e-01	1	1
## 33	2.464487e-02	0	0
## 40	6.229674e-01	1	1
## 10	2.222471e-06	0	0
## 295	6.099286e-01	0	1
## 125	9.982693e-01	1	1
## 265	9.999849e-01	1	1
## 440	2.816242e-01	1	0
## 186	5.008080e-01	0	1
## 61	6.271847e-01	1	1
## 443	9.998422e-01	1	1
## 152	1.570374e-03	0	0
## 54	6.617275e-02	0	0
## 442	1.000000e+00	1	1
## 394	1.160161e-02	0	0
## 185	5.199023e-02	0	0
## 253	1.000000e+00	1	1
## 115	2.209269e-03	0	0
## 430	9.999999e-01	1	1
## 279	9.999964e-01	1	1
## 205	2.656012e-01	0	0
## 267	9.999975e-01	1	1
## 377	9.999997e-01	1	1
## 380	1.507094e-03	0	0
## 215	9.999987e-01	1	1
## 404	1.391056e-03	0	0
## 57	1.050833e-01	0	0
## 105	9.999070e-01	1	1
## 134	9.999535e-01	1	1
## 129	1.273640e-01	0	0
## 218	7.918109e-01	1	1
## 106	1.000000e+00	1	1
## 283	9.315404e-01	1	1
## 379	9.999983e-01	1	1
## 27	3.388781e-01	0	0
## 213	8.180534e-01	1	1
## 447	9.998777e-01	1	1
## 188	9.999999e-01	1	1
## 245	6.301129e-04	0	0
## 341	9.999988e-01	1	1
## 402	1.877317e-03	0	0
## 407	4.510847e-03	0	0
## 284	1.000000e+00	1	1
## 242	4.372585e-01	1	0

## 222	2.425071e-01	1	0
## 431	6.665649e-02	0	0
## 93	9.999953e-01	1	1
## 145	6.714748e-01	0	1
## 192	1.555496e-01	0	0
## 363	1.038464e-01	0	0
## 285	6.573430e-04	0	0
## 148	7.292191e-01	1	1
## 163	1.296879e-02	0	0
## 466	9.999990e-01	1	1
## 161	9.999117e-01	1	1
## 66	6.911281e-01	1	1
## 384	9.999991e-01	1	1
## 360	1.353884e-03	0	0
## 225	8.690828e-01	1	1
## 133	1.283883e-03	0	0
## 117	9.999996e-01	1	1
## 269	1.000000e+00	1	1
## 362	1.157361e-04	0	0
## 356	2.708268e-02	0	0
## 329	4.612409e-01	1	0
## 301	1.678647e-03	0	0
## 45	3.788297e-01	0	0
## 146	9.431104e-01	1	1
## 170	1.000000e+00	1	1
## 246	1.114487e-01	0	0
## 296	1.000000e+00	1	1
## 375	9.999990e-01	1	1
## 189	9.999999e-01	1	1
## 264	2.672225e-01	0	0
## 176	9.992137e-01	1	1
## 456	3.856264e-03	0	0
## 173	5.887179e-01	1	1
## 191	1.119640e-02	0	0
## 53	6.897978e-02	0	0
## 104	6.846447e-05	0	0
## 450	9.999984e-01	1	1
## 410	1.361679e-01	0	0
## 220	9.384917e-04	0	0
## 275	1.118016e-01	0	0
## 216	6.454739e-01	0	1
## 2	9.999784e-01	1	1
## 386	1.000000e+00	1	1
## 130	9.458762e-03	0	0
## 409	9.996295e-01	1	1
## 24	3.993683e-02	0	0
## 271	9.999999e-01	1	1
## 312	9.999992e-01	1	1
## 204	7.403909e-01	1	1
## 248	9.999866e-01	1	1
## 444	1.776073e-05	0	0
## 418	3.626946e-01	0	0
## 350	2.037638e-03	0	0
## 383	5.334861e-03	0	0

## 399	9.999738e-01	1	1
## 219	9.999903e-01	1	1
## 260	9.999715e-01	1	1
## 416	7.771999e-05	0	0
## 112	1.339007e-01	0	0
## 369	9.086491e-01	1	1
## 80	2.032456e-01	0	0
## 36	2.244270e-05	0	0
## 35	3.415006e-01	0	0
## 87	9.578509e-01	1	1
## 367	7.232966e-01	0	1
## 323	9.999816e-01	1	1
## 111	9.795415e-01	1	1
## 423	9.902382e-01	1	1
## 325	3.785153e-01	1	0
## 282	1.196185e-01	0	0
## 319	2.124643e-01	0	0
## 31	4.676370e-01	1	0
## 73	4.782819e-01	1	0
## 259	6.998994e-04	0	0
## 307	8.098927e-01	1	1
## 122	4.083833e-03	0	0
## 251	3.865454e-03	0	0
## 92	4.708197e-05	0	0
## 208	2.803011e-02	0	0
## 315	9.647474e-01	1	1
## 433	7.511211e-01	0	1
## 201	4.966445e-06	0	0
## 297	1.391450e-01	1	0
## 458	7.656512e-01	0	1
## 414	1.776843e-01	0	0
## 321	6.060630e-01	1	1
## 140	2.440942e-03	0	0
## 357	2.441302e-01	0	0
## 21	1.112830e-02	0	0
## 318	6.440007e-02	0	0
## 314	8.854152e-03	0	0
## 276	1.000000e+00	1	1
## 30	2.486884e-01	0	0
## 459	1.000000e+00	1	1
## 183	9.999999e-01	1	1
## 274	8.410852e-02	0	0
## 287	3.935952e-01	1	0
## 82	9.998809e-01	1	1
## 96	9.999898e-01	1	1
## 340	6.230720e-01	1	1
## 149	9.999964e-01	1	1
## 258	3.634421e-02	0	0
## 203	9.999971e-01	1	1
## 157	9.999991e-01	1	1
## 457	7.253530e-03	1	0
## 434	9.167377e-02	0	0
## 150	1.135520e-03	0	0
## 449	1.000000e+00	1	1

## 119	8.878904e-01	1	1
## 303	7.642608e-02	0	0
## 344	9.999940e-01	1	1
## 108	2.528915e-02	0	0
## 388	3.369902e-01	0	0
## 227	3.310587e-01	0	0
## 333	5.493574e-01	1	1
## 270	7.295393e-02	0	0
## 144	8.877099e-01	1	1
## 376	9.999999e-01	1	1
## 12	3.961198e-03	0	0
## 437	3.782146e-04	0	0
## 28	9.999993e-01	1	1
## 358	9.454929e-01	1	1
## 50	7.897293e-01	1	1
## 370	2.465733e-01	0	0
## 47	9.999906e-01	1	1
## 361	1.253585e-02	0	0
## 231	9.999991e-01	1	1
## 6	4.155308e-01	0	0
## 187	6.200295e-05	0	0
## 15	6.546783e-02	0	0
## 19	9.999971e-01	1	1
## 59	2.206955e-02	0	0
## 3	1.000000e+00	1	1
## 371	9.999609e-01	1	1
## 97	1.595275e-03	0	0
## 239	7.675915e-02	0	0
## 182	4.820345e-03	0	0
## 75	9.999715e-01	1	1
## 100	5.565203e-01	0	1
## 429	9.999942e-01	1	1
## 233	2.060231e-01	1	0
## 351	1.523541e-04	0	0
## 305	4.587527e-01	1	0
## 202	9.999932e-01	1	1
## 193	1.658804e-03	0	0
## 18	5.760076e-01	1	1
## 142	6.825972e-02	0	0
## 304	6.991085e-01	0	1
## 397	2.016938e-03	0	0
## 9	6.999953e-01	0	1
## 128	4.283102e-01	0	0
## 138	5.048634e-01	1	1
## 8	9.999973e-01	1	1

To calculate some of the metrics we need, let's first find the 'True Positive' (TP), 'True Negative' (TN), 'False Positive' (FP) and 'False Negative' (FN) values using our training dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

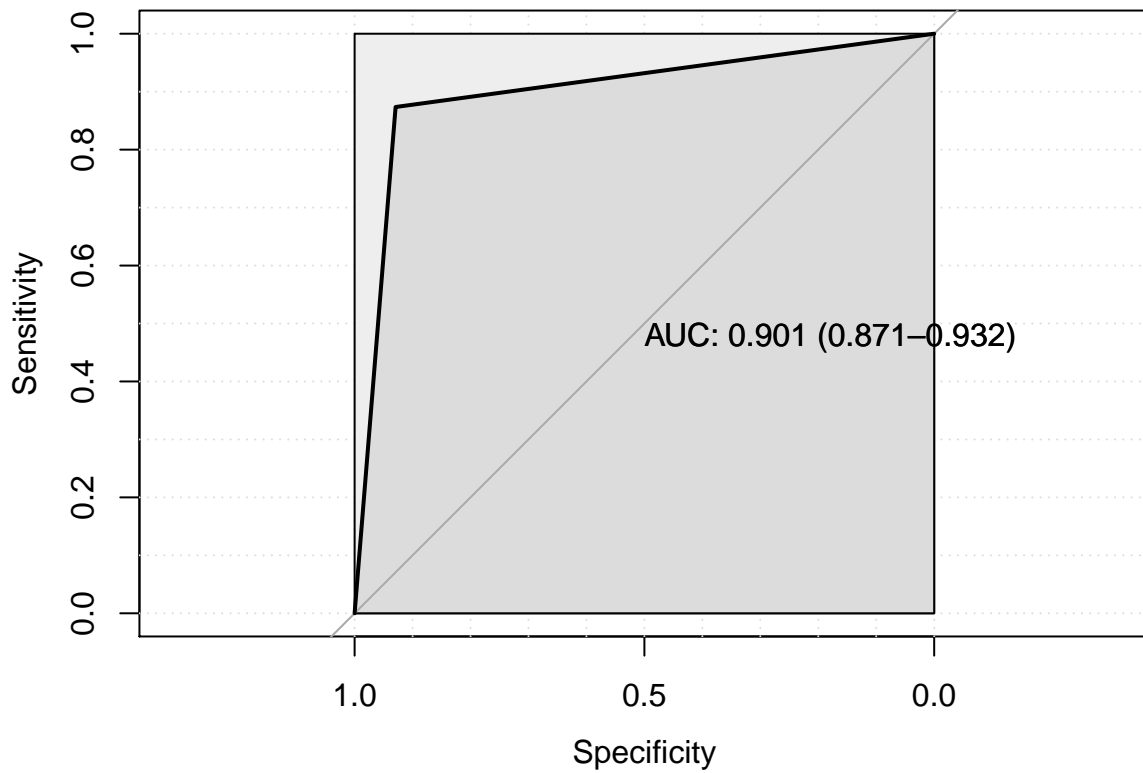
$$ClassificationErrorRate = \frac{FP + FN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$F1 = 2 * \frac{Precision * Sensitivity}{Precision + Sensitivity}$$



```
##           Actually_Positive Actually_Negative
## Guess Positive             152             14
## Guess Negative             22             184
```

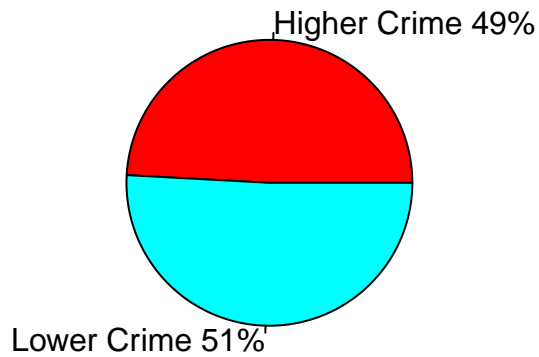
```
##   Accuracy      CER Precision Sensitivity Specificity      F1
## 1 0.9032258 0.09677419 0.9156627  0.8735632  0.9292929 0.8941176
```

When based on the training data set, our metrics paint a fairly good picture. We can see that the accuracy, precision and specificity of our model are all just over 90%, with both the sensitivity and F1-score falling slightly short of that level. Finally, below is our prediction for the Evaluation dataset.

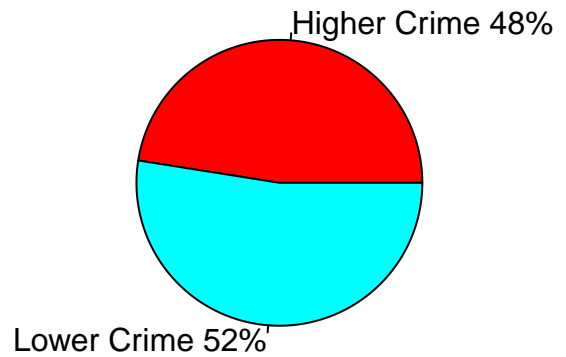
##	model	guess
## 1	1.094556e-01	0
## 2	7.215436e-01	1
## 3	7.869617e-01	1
## 4	3.877930e-01	0
## 5	7.767790e-02	0
## 6	1.457893e-01	0
## 7	1.710946e-01	0
## 8	1.361376e-02	0
## 9	3.285544e-03	0
## 10	1.498454e-03	0
## 11	6.891437e-02	0
## 12	3.419831e-02	0
## 13	7.368536e-01	1
## 14	7.185692e-01	1
## 15	5.585294e-01	1
## 16	9.636747e-02	0
## 17	3.191036e-01	0
## 18	9.860797e-01	1
## 19	6.406181e-02	0
## 20	3.889144e-06	0
## 21	3.846253e-06	0
## 22	2.375987e-02	0
## 23	1.000607e-01	0
## 24	2.068125e-01	0
## 25	1.820772e-01	0
## 26	5.845129e-01	1
## 27	2.247867e-04	0
## 28	1.000000e+00	1
## 29	9.999998e-01	1
## 30	9.998605e-01	1
## 31	9.999998e-01	1
## 32	9.999999e-01	1
## 33	9.999999e-01	1
## 34	1.000000e+00	1
## 35	9.999999e-01	1
## 36	9.999999e-01	1
## 37	1.000000e+00	1
## 38	9.999995e-01	1
## 39	8.204779e-01	1
## 40	3.305822e-01	0

Let's take a look at the difference between the values in our prediction and those of the training dataset. One comparison we can make is to see what proportion of the given neighborhoods was classified as 'Higher Crime' and compare that to the proportion of our predicted values.

Training Data



Prediction



Though the number of datapoints for the prediction is fairly low (40) and we don't know that the distribution of evaluated neighborhoods is at all similar to that of the training set, the fact that the breakdown is similar in the two sets is a promising sign for the validity of our model.

We have no way of knowing how well our model performs with real data as we were not provided the target values for the evaluation dataset.