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(Bk - 0.63)2 where Bk 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

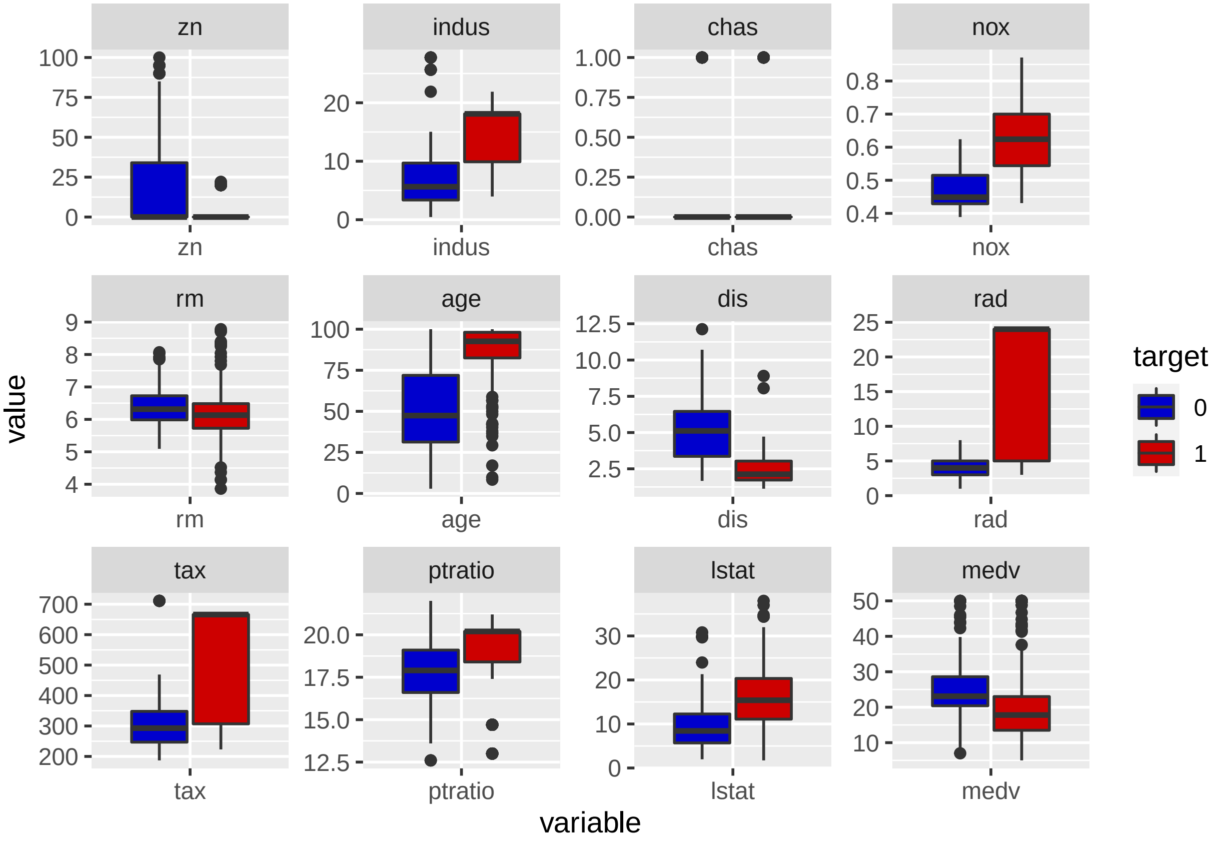
Data Exploration

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

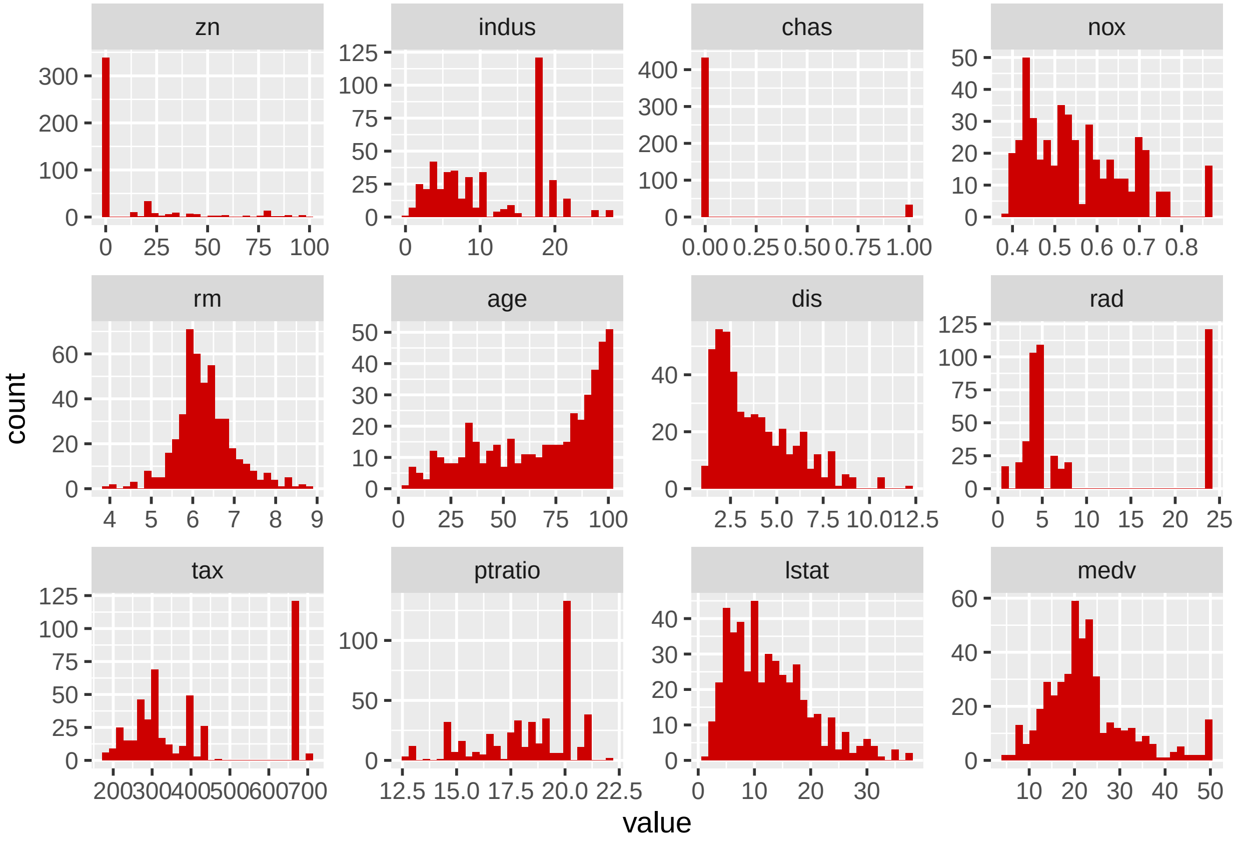
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
zn	1	466	11.577	23.365	0.000	5.354	0.000	0.000	100.000	100.000	2.177	3.814	1.082
indus	2	466	11.105	6.846	9.690	10.908	9.340	0.460	27.740	27.280	0.289	-1.243	0.317
chas	3	466	0.071	0.257	0.000	0.000	0.000	0.000	1.000	1.000	3.335	9.145	0.012
nox	4	466	0.554	0.117	0.538	0.544	0.133	0.389	0.871	0.482	0.746	-0.036	0.005
rm	5	466	6.291	0.705	6.210	6.257	0.517	3.863	8.780	4.917	0.479	1.542	0.033
age	6	466	68.368	28.321	77.150	70.955	30.023	2.900	100.000	97.100	-0.578	-1.010	1.312
dis	7	466	3.796	2.107	3.191	3.544	1.914	1.130	12.127	10.997	0.999	0.472	0.098
rad	8	466	9.530	8.686	5.000	8.698	1.483	1.000	24.000	23.000	1.010	-0.862	0.402
tax	9	466	409.502	167.900	334.500	401.508	104.523	187.000	711.000	524.000	0.659	-1.148	7.778
ptratio	10	466	18.398	2.197	18.900	18.597	1.927	12.600	22.000	9.400	-0.754	-0.400	0.102
lstat	11	466	12.631	7.102	11.350	11.881	7.072	1.730	37.970	36.240	0.906	0.503	0.329
medv	12	466	22.589	9.240	21.200	21.630	6.005	5.000	50.000	45.000	1.077	1.374	0.428
target	13	466	0.491	0.500	0.000	0.489	0.000	0.000	1.000	1.000	0.034	-2.003	0.023

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.

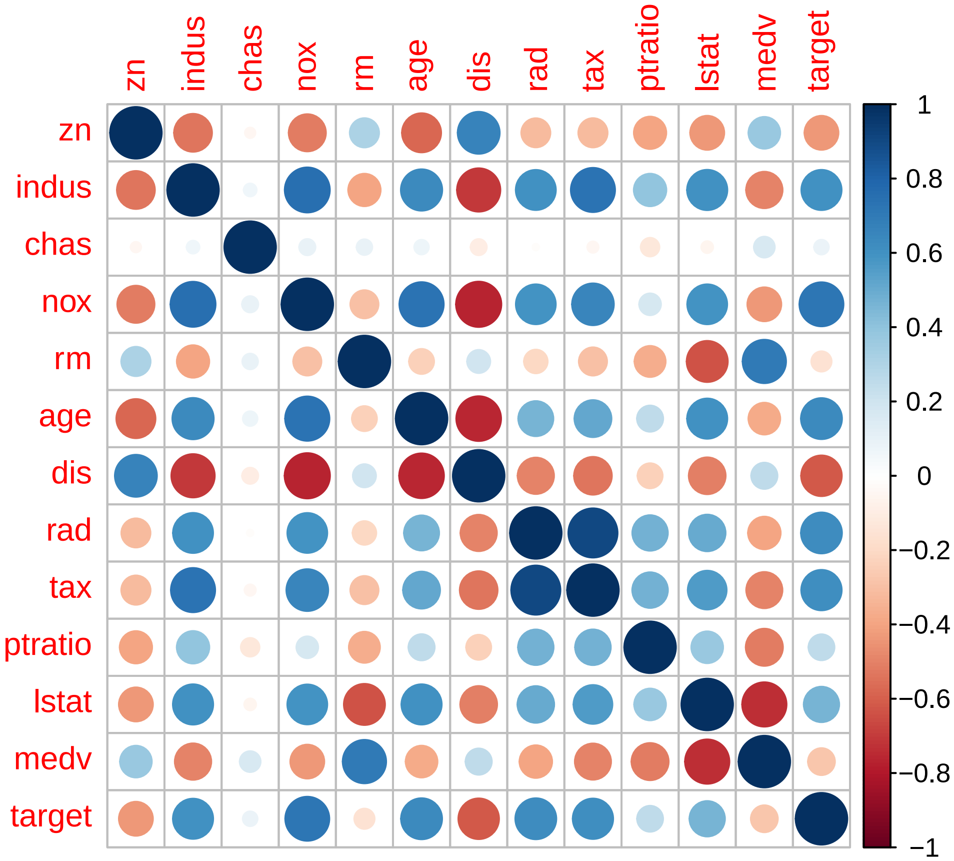


In particular, zn, nox, age, dis, ptratio, and lstat seem likely candidates for transformations.

Let's check for covariance.

	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
zn	1.000	-0.538	-0.040	-0.517	0.320	-0.573	0.660	-0.315	-0.319	-0.391	-0.433	0.377	-0.432
indus	-0.538	1.000	0.061	0.760	-0.393	0.640	-0.704	0.601	0.732	0.395	0.607	-0.496	0.605
chas	-0.040	0.061	1.000	0.097	0.091	0.079	-0.097	-0.016	-0.047	-0.129	-0.051	0.162	0.080
nox	-0.517	0.760	0.097	1.000	-0.295	0.735	-0.769	0.596	0.654	0.176	0.596	-0.430	0.726
rm	0.320	-0.393	0.091	-0.295	1.000	-0.233	0.199	-0.208	-0.297	-0.360	-0.632	0.705	-0.153
age	-0.573	0.640	0.079	0.735	-0.233	1.000	-0.751	0.460	0.512	0.255	0.606	-0.378	0.630
dis	0.660	-0.704	-0.097	-0.769	0.199	-0.751	1.000	-0.495	-0.534	-0.233	-0.508	0.257	-0.619
rad	-0.315	0.601	-0.016	0.596	-0.208	0.460	-0.495	1.000	0.906	0.471	0.503	-0.398	0.628
tax	-0.319	0.732	-0.047	0.654	-0.297	0.512	-0.534	0.906	1.000	0.474	0.564	-0.490	0.611
ptratio	-0.391	0.395	-0.129	0.176	-0.360	0.255	-0.233	0.471	0.474	1.000	0.377	-0.516	0.251
lstat	-0.433	0.607	-0.051	0.596	-0.632	0.606	-0.508	0.503	0.564	0.377	1.000	-0.736	0.469
medv	0.377	-0.496	0.162	-0.430	0.705	-0.378	0.257	-0.398	-0.490	-0.516	-0.736	1.000	-0.271
target	-0.432	0.605	0.080	0.726	-0.153	0.630	-0.619	0.628	0.611	0.251	0.469	-0.271	1.000

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.

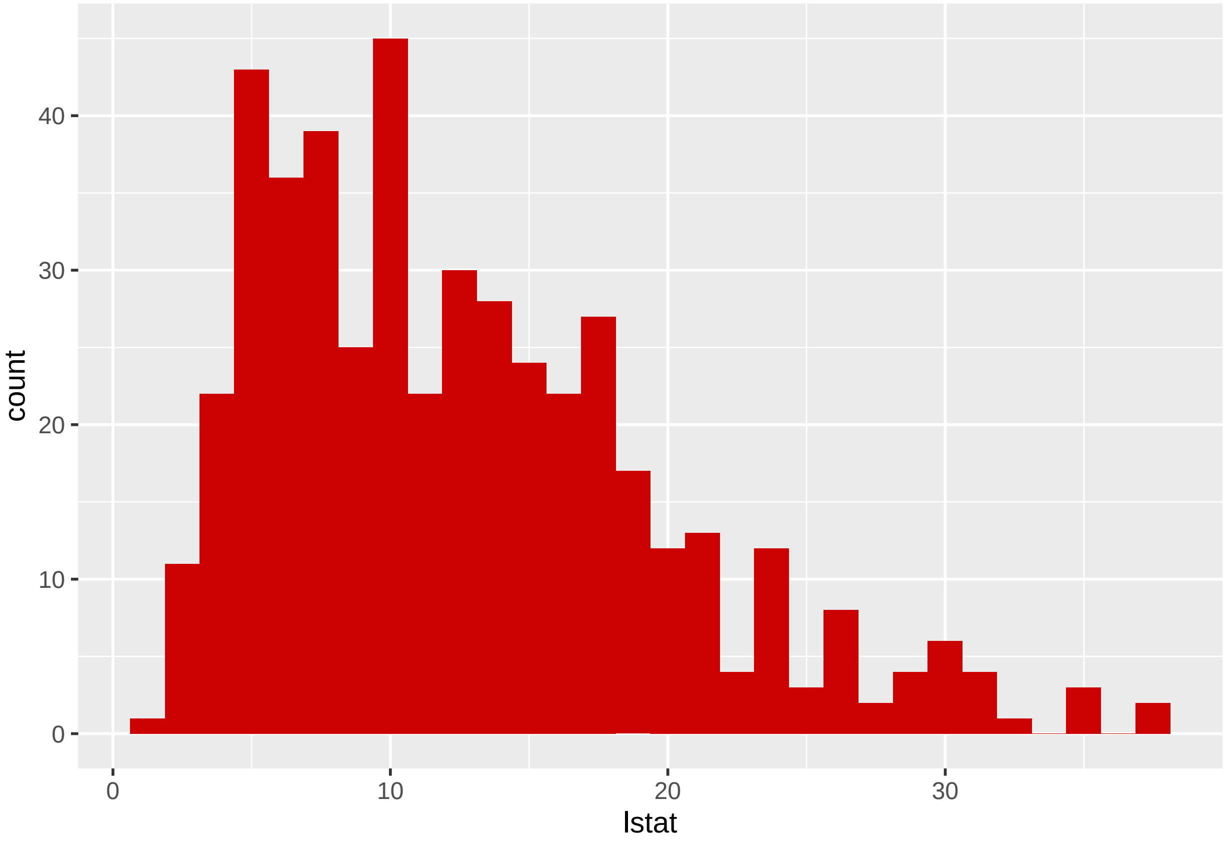
As a final step, let's look just a correlation between the independent variables and the target variables.

	target
zn	-0.432
indus	0.605
chas	0.080
nox	0.726
rm	-0.153
age	0.630
dis	-0.619
rad	0.628
tax	0.611
ptratio	0.251
lstat	0.469
medv	-0.271
target	1.000

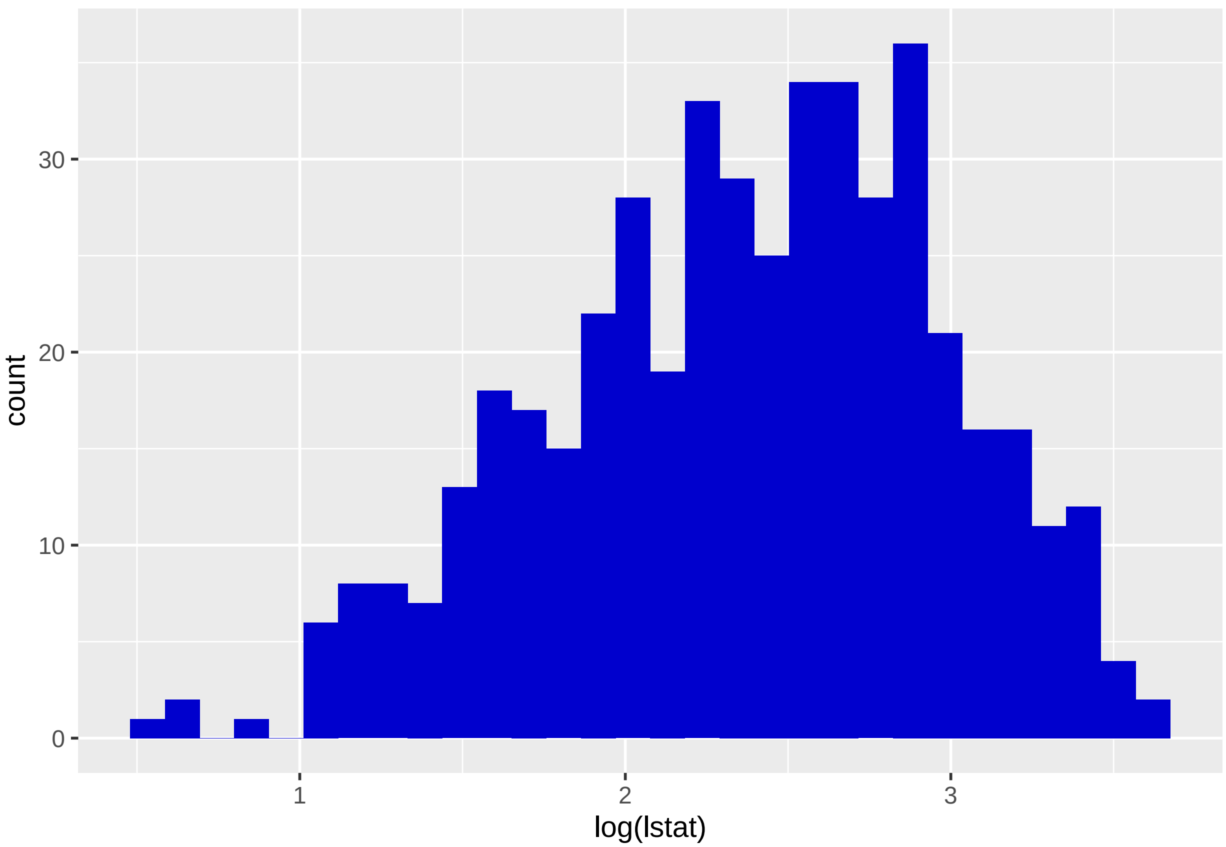
We see that Nox(nitrogen oxides concentration), shows the closest correlation with the target variable at .73. Next, age, rad, tax, and indus all correlate with the target value just above .6.

Data Preparation

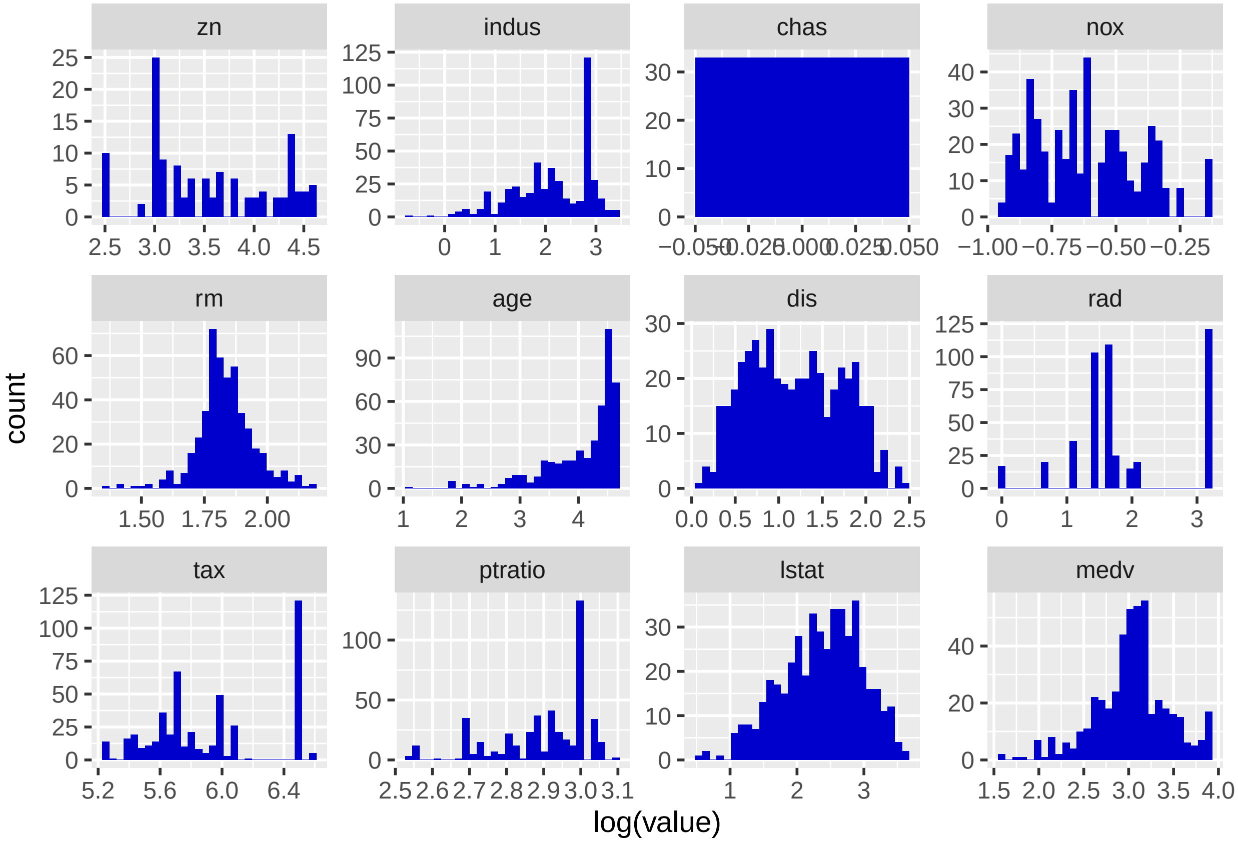
Let's look at how transformations might solve distribution issues with some of our variables. Earlier, we saw a strong right skew in the distribution of the variable lstat, which tracks the "lower status" of a neighborhood's population. Probably not the best phrasing.



What would a log transformation do to this distribution?

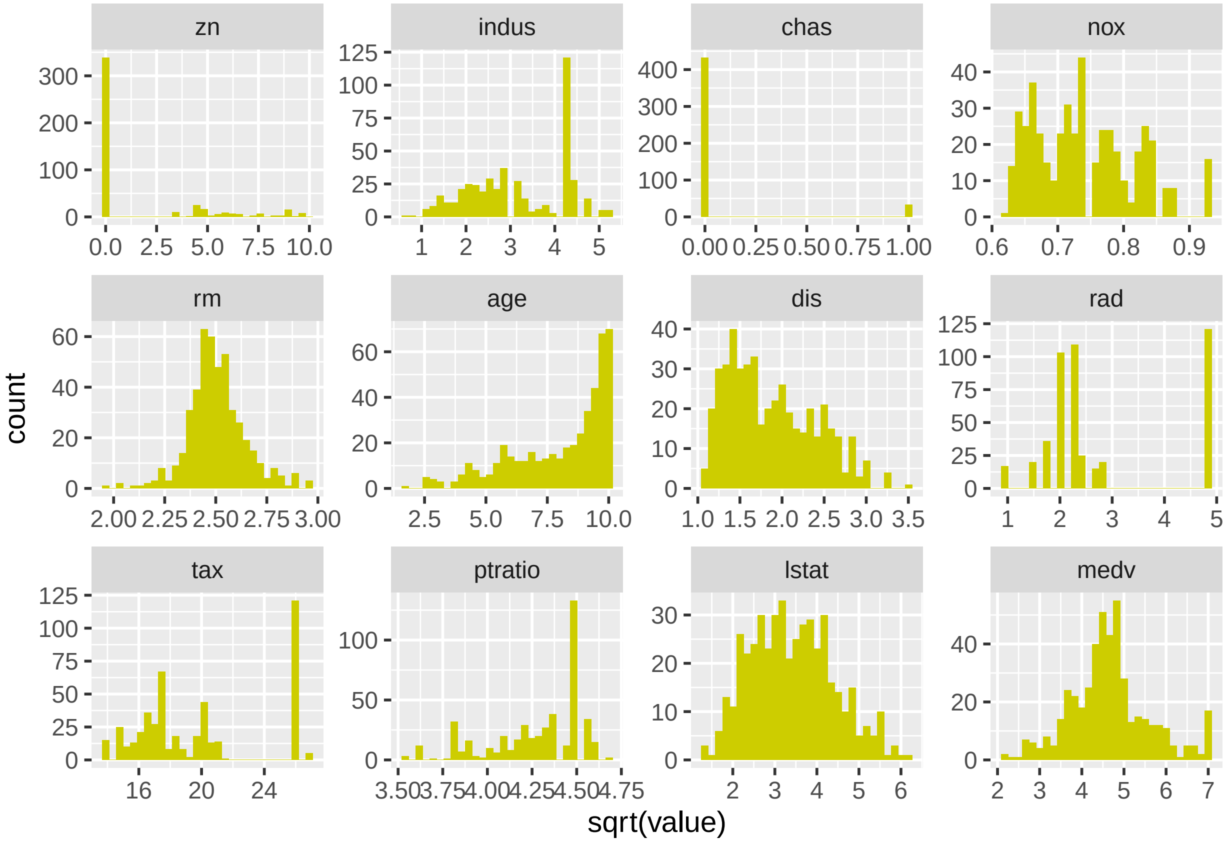


Looks slightly better. Let’s generate log transformations for all variables in the dataset.



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.

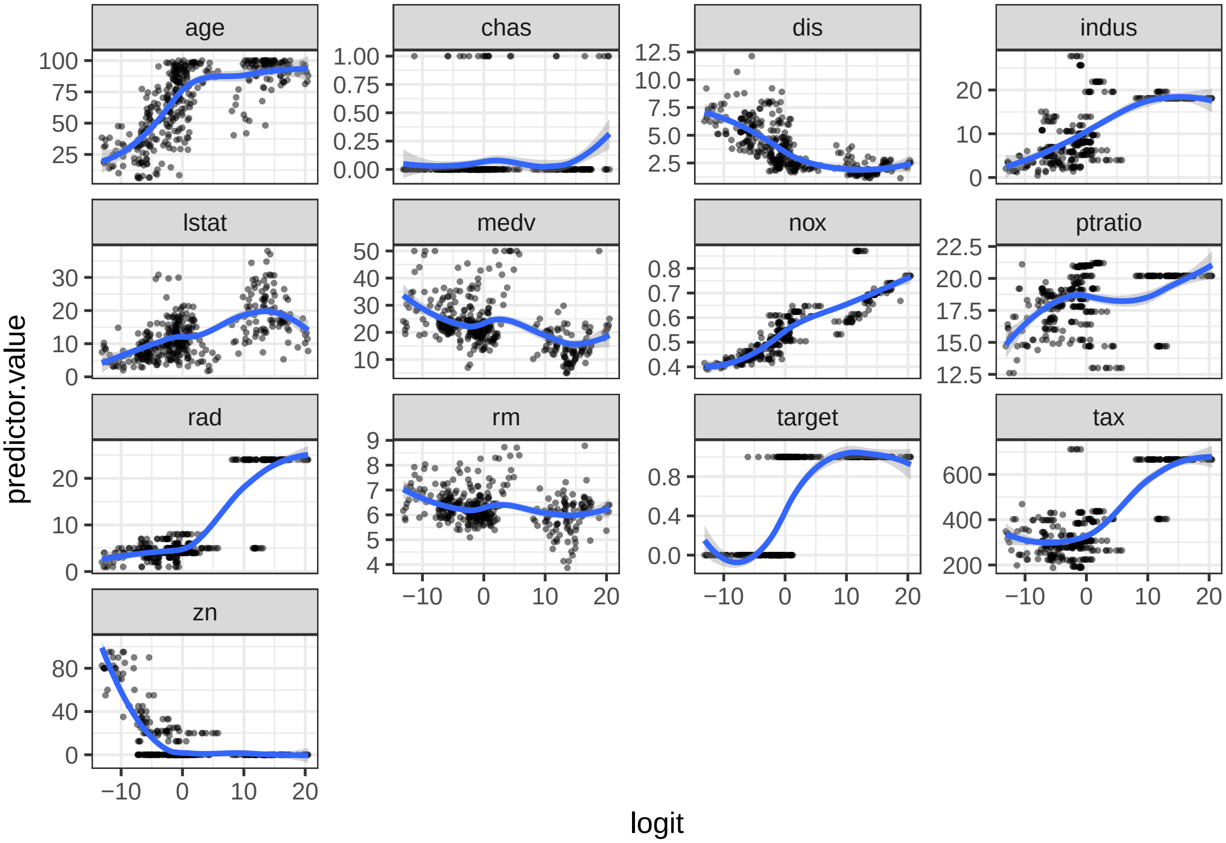
Model 1 - All Variables Untransformed

```
##
## 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
```

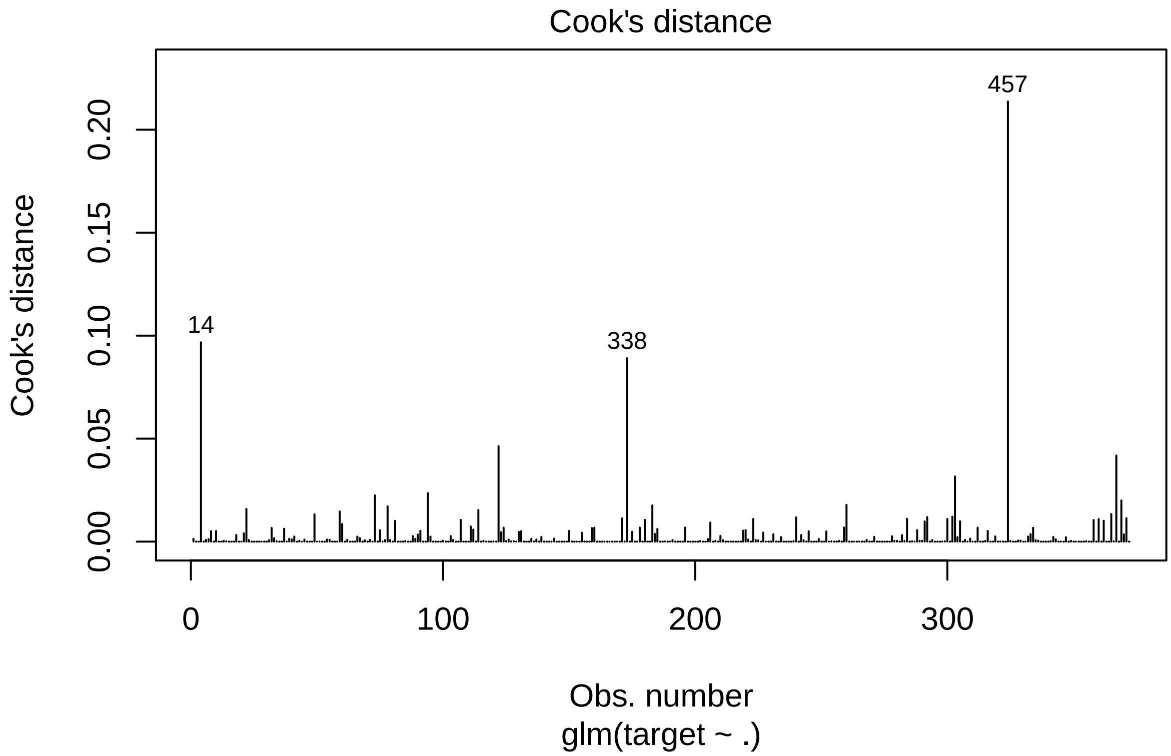
Our most significant variables generally tie to the variables we saw have the highest correlations with the target value earlier. We have an AIC of 190.56 and a residual deviance of 164.56.

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	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid	index
14	1	22	5.9	0	0.4	8.3	8.4	8.9	7	330	19.1	3.5	42.8	-0.5	1.2	1.4	0.3	0.7	0.1	1.7	4
338	1	20	7.0	0	0.5	5.9	42.1	4.4	3	223	18.6	13.0	21.1	-6.1	1.1	3.5	0.0	0.7	0.1	3.5	173
457	1	0	10.6	0	0.5	5.4	9.8	3.6	4	277	18.6	29.6	23.7	-4.4	1.6	3.0	0.0	0.7	0.2	3.0	324

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	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid	index
338	1	20	7.0	0	0.5	5.9	42.1	4.4	3	223	18.6	13.0	21.1	-6.1	1.1	3.5	0	0.7	0.1	3.5	173
457	1	0	10.6	0	0.5	5.4	9.8	3.6	4	277	18.6	29.6	23.7	-4.4	1.6	3.0	0	0.7	0.2	3.0	324

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 appling 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 or 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 0.12316893      0     0
## 463 0.99999695      1     1
## 179 0.99999674      1     1
## 14  0.41826530      1     0
## 195 0.99998970      1     1
## 426 0.08268266      0     0
```

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) valuesusing our training dataset.

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

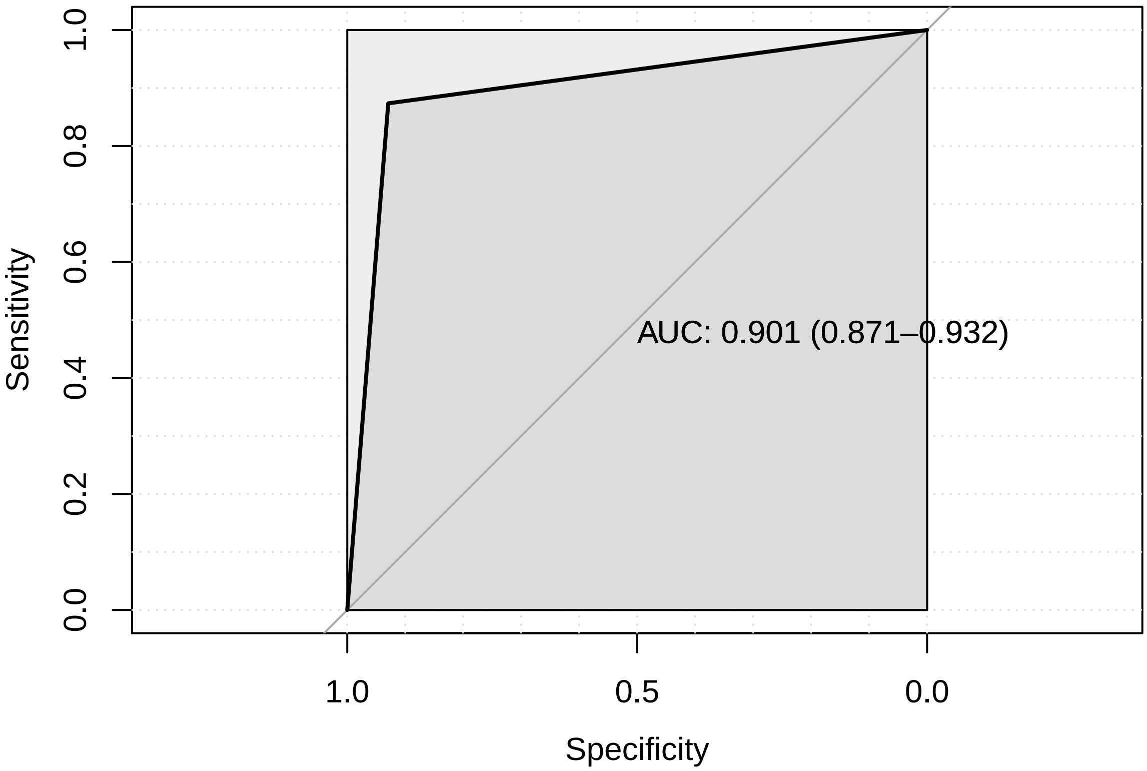
$$ClassificationErrorRate = \frac{FP + FN}{TP + TN + FP + FN}$$

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

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

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

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



Below is our Confidence matrix, which gives us an idea of how many predictions were correct and how many we got wrong - in ther words how confident we can be in our model. Under that we can see a breakdown of each of the calculated metrics from the equations above.

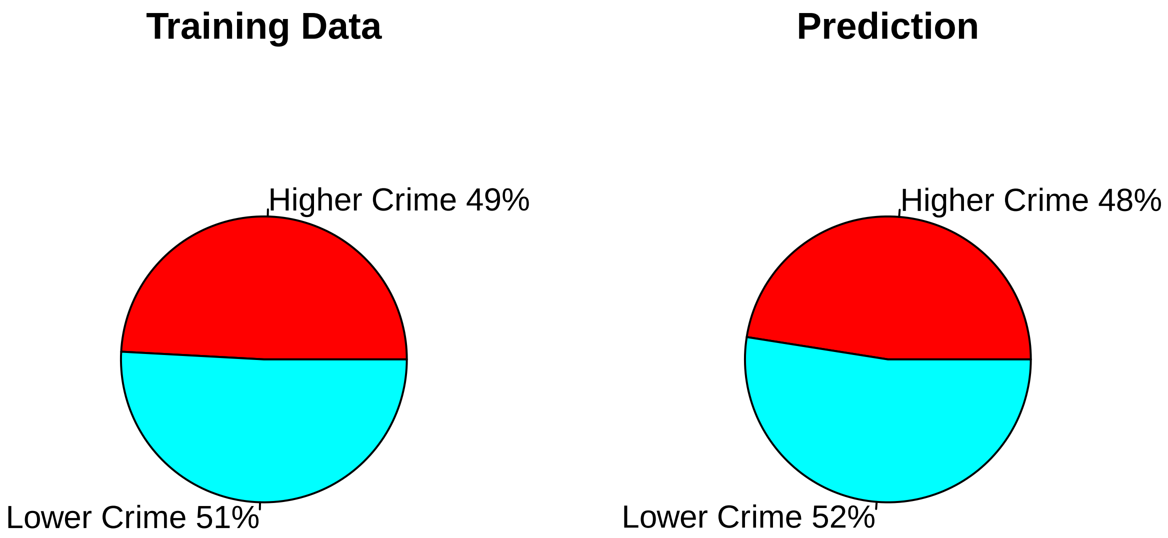
```
##                 Actually_Positive Actually_Negative
## Guess Positive                152                14
## Guess Negative                 22               184
```

```
##   round.Accuracy..4. round.CER..4. round.Precision..4.
## 1             0.9032        0.0968              0.9157
##   round.Sensitivity..4. round.Specificity..4. round.F1..4.
## 1                0.8736                0.9293       0.8941
```

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.



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 valididy 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.