DATA 621 Homework 3

Ilya Kats

Summary

This report covers an attempt to build a binary logistic regression model to predict whether the crime rate is above the median crime rate. The model is based on a data set containing information on crime for various Boston neighborhoods.

Data Exploration

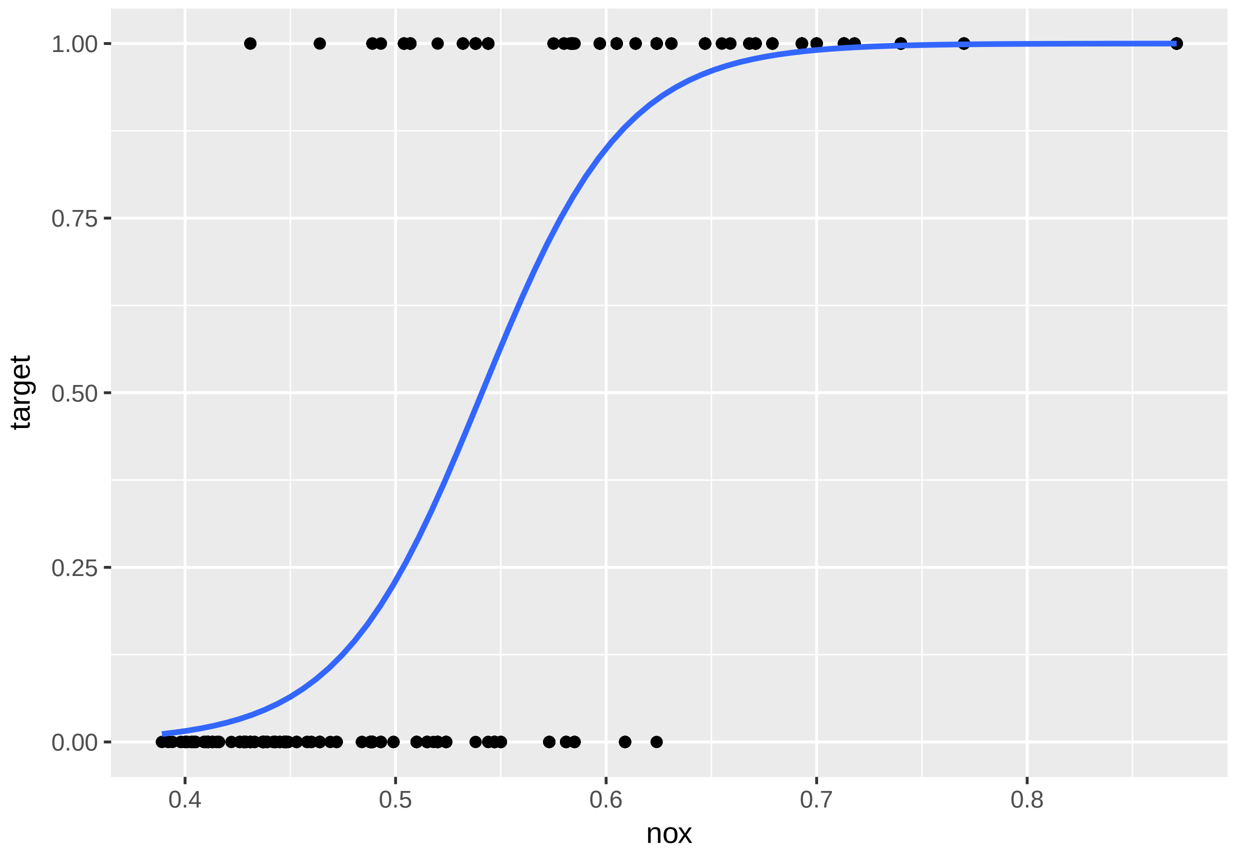
The data set includes 466 observations with 12 variables (excluding the target variable).

Summary of Variables

Variable	Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
zn	0.0000	0.00000	11.5772532	23.3646511	100.0000	0	339
indus	0.4600	9.69000	11.1050215	6.8458549	27.7400	0	0
chas	0.0000	0.00000	0.0708155	0.2567920	1.0000	0	433
nox	0.3890	0.53800	0.5543105	0.1166667	0.8710	0	0
rm	3.8630	6.21000	6.2906738	0.7048513	8.7800	0	0
age	2.9000	77.15000	68.3675966	28.3213784	100.0000	0	0
dis	1.1296	3.19095	3.7956929	2.1069496	12.1265	0	0
rad	1.0000	5.00000	9.5300429	8.6859272	24.0000	0	0
tax	187.0000	334.50000	409.5021459	167.9000887	711.0000	0	0
ptratio	12.6000	18.90000	18.3984979	2.1968447	22.0000	0	0
lstat	1.7300	11.35000	12.6314592	7.1018907	37.9700	0	0
medv	5.0000	21.20000	22.5892704	9.2396814	50.0000	0	0
target	0.0000	0.00000	0.4914163	0.5004636	1.0000	0	237

Independent Variables

- `zn` - *proportion of residential land zoned for large lots (over 25,000 square feet)* - 339 out of 466 (or about 76%) of observations have a value of 0. It is possible that majority of neighborhoods will not have any residential land zoned for large lots. Therefore, it is likely that 0 represents a valid value rather than a missing one.
- `indus` - *proportion of non-retail business acres per suburb*
- `chas` - *a dummy variable for whether the suburb borders the Charles River (1) or not (0)* - 433 out of 466 (or about 92.9%) of observations have a value of 0. Even though the Charles River is a promimnent feature of the Boston area, it is quite reasonable to assume that most neighborhoods do not border the river.
- `nox` - *nitrogen oxides concentration (parts per 10 million)* - Looking at the scatterplot there seems to be some correlation between the nitrogen oxides concentration and the target variable.



- `rm` - *average number of rooms per dwelling* - Because this is an average of number of rooms, this is a continous variable.
- `age` - *proportion of owner-occupied units built prior to 1940* - There is nothing unusual about this variable; however, it is interesting to note that the mean of 68.37 and median of 77.15 shows that there is a large number of pre-war buildings. Not surprising for an old city like Boston.
- `dis` - *weighted mean of distances to five Boston employment centers* - Majority of observations above the median crime rate are within 5 miles of an employment center (there are only 2 observations over 5 miles away). And there may be some correlation between distance and the target variable.
- `rad` - *index of accessibility to radial highways* - This is a discrete variable with 9 different values in the observations (1 through 8 and 24). The smallest bucket is `rad` value of 7 with 15 observations.
- `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/Dependent Variable

- `target` - *whether the crime rate is above the median crime rate (1) or not (0)* - There are 237 observation with `target` value of 0 and 229 observations with `target` value of 1 making it about 50/50 split, or more precisely there are **50.86% of 0s and 49.14% of 1s**.

Correlation Matrix

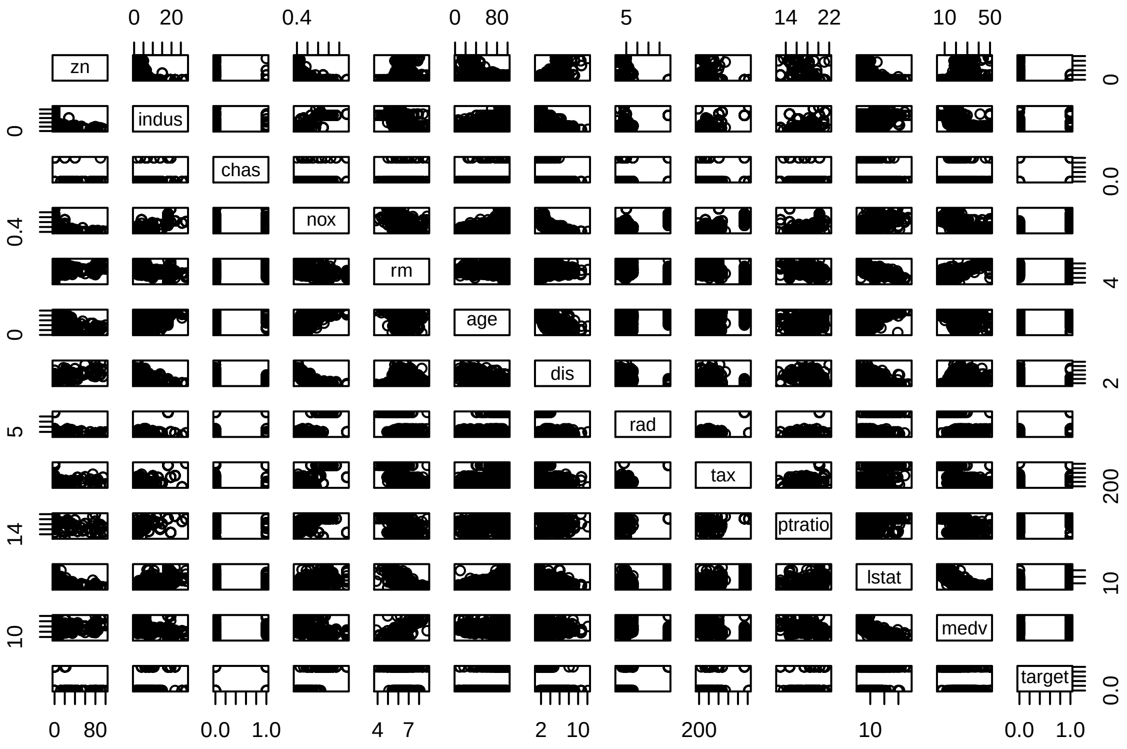
Below is the correlation matrix for the data set.

- There is a very high correlation (0.91) between `tax` and `rad`. Meaning behind `rad` values/categories is not explicitly specified. It may be that the higher the highway accessibility is, the higher property taxes are. Alternatively, radial highways and higher property taxes may signify suburbs while lack of radial highways may imply inner city (with possibly poorer, lower taxed properties).

- `nox` has the highest correlation with the target variable, but `age`, `dis`, `rad` and `tax` are also fairly highly correlated to `target` (above 0.6).
- The following pairs have correlation at or above 0.7 (or below -0.7 in case of negative correlation): `nox`/`indus`, `dis`/`indus`, `tax`/`indus`, `age`/`nox`, `dis`/`nox`, `medv`/`rm`, `dis`/`age` and `medv`/`lstat`.

	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
zn	1	-0.54	-0.04	-0.52	0.32	-0.57	0.66	-0.32	-0.32	-0.39	-0.43	0.38	-0.43
indus	-0.54	1	0.06	0.76	-0.39	0.64	-0.7	0.6	0.73	0.39	0.61	-0.5	0.6
chas	-0.04	0.06	1	0.1	0.09	0.08	-0.1	-0.02	-0.05	-0.13	-0.05	0.16	0.08
nox	-0.52	0.76	0.1	1	-0.3	0.74	-0.77	0.6	0.65	0.18	0.6	-0.43	0.73
rm	0.32	-0.39	0.09	-0.3	1	-0.23	0.2	-0.21	-0.3	-0.36	-0.63	0.71	-0.15
age	-0.57	0.64	0.08	0.74	-0.23	1	-0.75	0.46	0.51	0.26	0.61	-0.38	0.63
dis	0.66	-0.7	-0.1	-0.77	0.2	-0.75	1	-0.49	-0.53	-0.23	-0.51	0.26	-0.62
rad	-0.32	0.6	-0.02	0.6	-0.21	0.46	-0.49	1	0.91	0.47	0.5	-0.4	0.63
tax	-0.32	0.73	-0.05	0.65	-0.3	0.51	-0.53	0.91	1	0.47	0.56	-0.49	0.61
ptratio	-0.39	0.39	-0.13	0.18	-0.36	0.26	-0.23	0.47	0.47	1	0.38	-0.52	0.25
lstat	-0.43	0.61	-0.05	0.6	-0.63	0.61	-0.51	0.5	0.56	0.38	1	-0.74	0.47
medv	0.38	-0.5	0.16	-0.43	0.71	-0.38	0.26	-0.4	-0.49	-0.52	-0.74	1	-0.27
target	-0.43	0.6	0.08	0.73	-0.15	0.63	-0.62	0.63	0.61	0.25	0.47	-0.27	1

Scatterplot Matrix



- Reviewing the scatterplot matrix shows several pairs with possible relationships. Two most prominent are `nox/dis` and `lstat/medv`.
- `rm` and `medv` may have a linear relationship as well.
- `rad` and `tax` have a very prominent outlier. Further inspection of data shows that all observations with `rad` value of 24 have a `tax` value of 666, so the outlier is actually multiple observations mapped to the same spot. Interestingly, all of these obervations also have a `target` value of 1. This combination may warrant closer inspection.

Above data analysis treats `rad` and `chas` as numeric variables; however, treating them as categorical variables may better reflect the nature of those variables, so for modelling they will be converted to factors.

Modelling

The dependent variable, `target`, is binary. For this project it is assumed that observations are independent of each other as there is no reason to believe otherwise.

As the first step, in order to test and compare performance of various models, data was split into training (75%) and testing (25%) sets. The training set includes 350 randomly chosen observations and the testing set includes 116 remaining observations.

Model 1: Variables with High Correlation to Target Variable

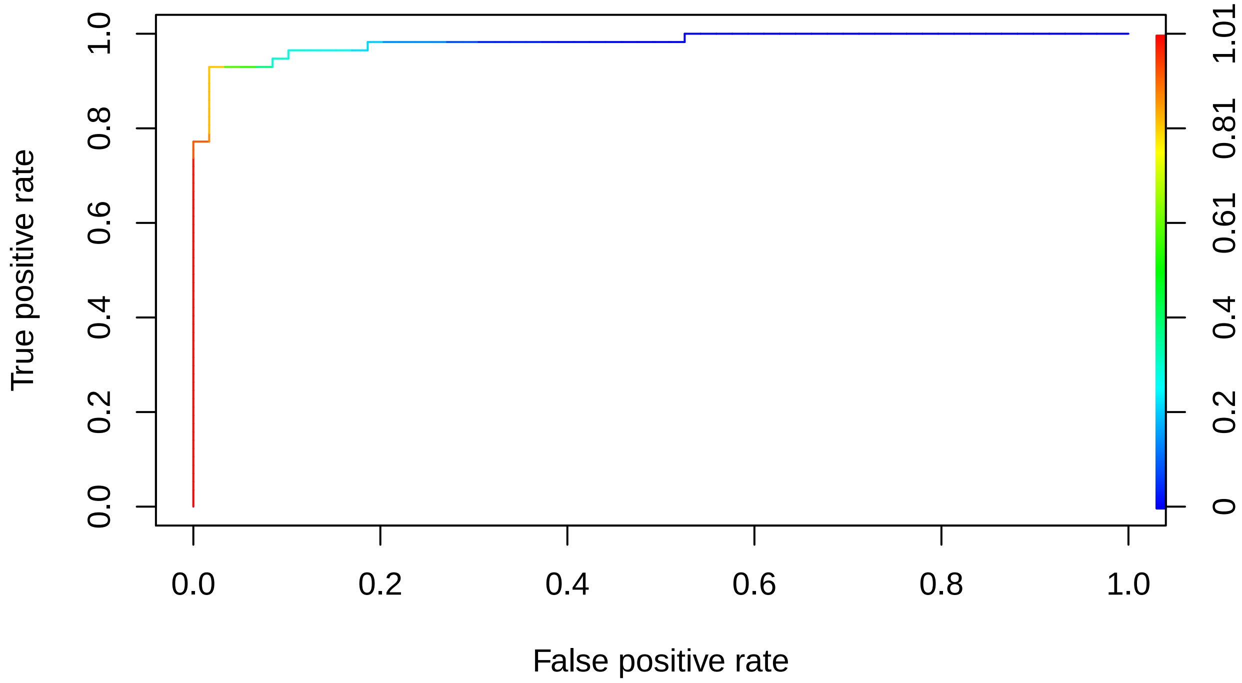
The first model includes 5 variables with the highest correlation coefficients when compared agains the target variable. This simple model will allow for testing methodology as well as corresponding R code.

```
##
## Call:
## glm(formula = target ~ nox + age + dis + rad + tax, family = binomial(link = "logit"),
##     data = crimeTRAIN)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0293  -0.1272   0.0000   0.0001   3.2135
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.870e+01  3.499e+03  -0.014    0.989
## nox          6.026e+01  1.220e+01   4.939 7.84e-07 ***
## age          2.782e-03  1.372e-02   0.203    0.839
## dis          7.224e-03  2.592e-01   0.028    0.978
## rad2        -2.036e+00  4.801e+03   0.000    1.000
## rad3         2.044e+01  3.499e+03   0.006    0.995
## rad4         2.262e+01  3.499e+03   0.006    0.995
## rad5         2.018e+01  3.499e+03   0.006    0.995
## rad6         1.846e+01  3.499e+03   0.005    0.996
## rad7         2.294e+01  3.499e+03   0.007    0.995
## rad8         2.544e+01  3.499e+03   0.007    0.994
## rad24        4.349e+01  3.773e+03   0.012    0.991
## tax         -1.656e-02  3.910e-03  -4.235 2.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 485.100  on 349  degrees of freedom
## Residual deviance:  97.662  on 337  degrees of freedom
## AIC: 123.66
##
## Number of Fisher Scoring iterations: 19
```

```
##           Reference
## Prediction  0  1
##          0 56  3
##          1  4 53
```

Model summary and confusion matrix of running this model against test data are above. The accuracy rate (0.9396552) is very good and the McFadden R^2 value (0.7986761) is also high. AIC value is 123.66. Additionally, consider the ROC curve for this model.



Area under the curve is 0.9815641.

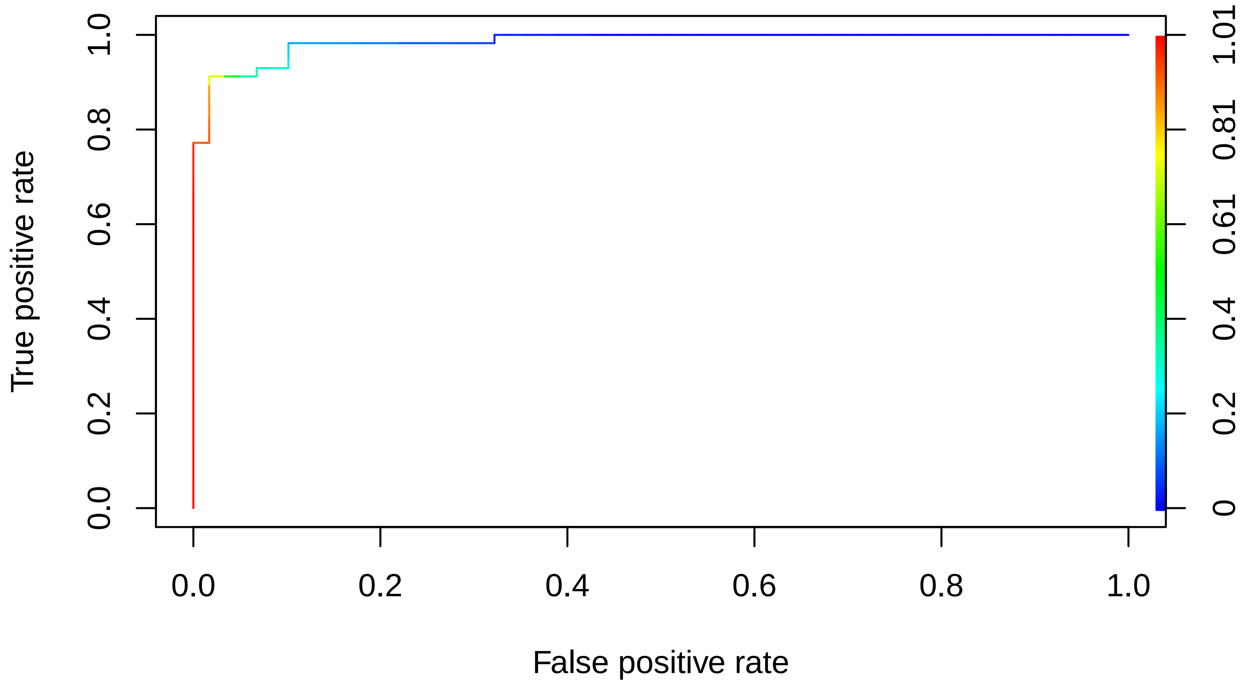
Model 2: All Variables

The second model includes all 12 available independent variables.

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = crimeTRAIN)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6329  -0.0803   0.0000   0.0001   4.1121
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.294e+01  3.400e+03  -0.016   0.9876
## zn          -1.444e-01  7.044e-02  -2.051   0.0403 *
## indus       -1.363e-01  1.221e-01  -1.116   0.2644
## chas1       -1.543e+00  1.480e+00  -1.042   0.2973
## nox          6.142e+01  1.452e+01   4.230 2.34e-05 ***
## rm          -1.867e-01  1.262e+00  -0.148   0.8824
## age          1.133e-02  1.837e-02   0.617   0.5372
## dis          3.650e-01  2.982e-01   1.224   0.2210
## rad2        -4.535e-01  4.821e+03   0.000   0.9999
## rad3         1.738e+01  3.400e+03   0.005   0.9959
## rad4         2.143e+01  3.400e+03   0.006   0.9950
## rad5         1.873e+01  3.400e+03   0.006   0.9956
## rad6         1.647e+01  3.400e+03   0.005   0.9961
## rad7         2.451e+01  3.400e+03   0.007   0.9942
## rad8         2.464e+01  3.400e+03   0.007   0.9942
## rad24        4.091e+01  3.695e+03   0.011   0.9912
## tax         -9.692e-03  6.242e-03  -1.553   0.1205
## ptratio     -1.060e-02  2.141e-01  -0.050   0.9605
## lstat        7.859e-02  7.283e-02   1.079   0.2806
## medv         1.320e-01  1.157e-01   1.141   0.2540
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 485.100  on 349  degrees of freedom
## Residual deviance:  88.097  on 330  degrees of freedom
## AIC: 128.1
##
## Number of Fisher Scoring iterations: 19
```

```
##           Reference
## Prediction  0  1
##          0 58  1
##          1  5 52
```

Model summary and confusion matrix of running this model against test data are above. The accuracy rate (0.9482759) is very good and the McFadden Rˆ2 value (0.8183936) is also high. AIC value is 128.1. Additionally, consider the ROC curve for this model.



Area under the curve is 0.9854297.

Comparing to the first model AIC has slightly increased (worse), but accuracy, McFadden R^2 and AUC all also slightly increased (better).

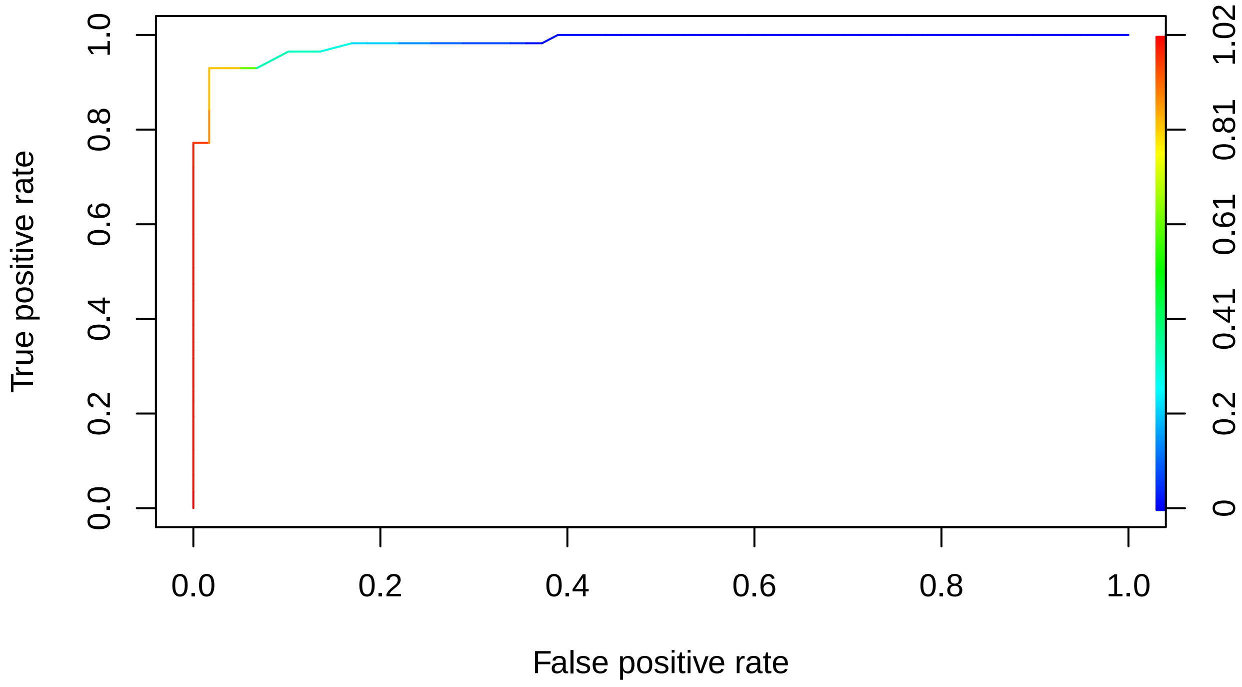
Model 3: ***StepAIC*** Method

The third model starts with all 12 available independent variables, but then drops them one by one using the stepwise algorithm.

```
##
## Call:
## glm(formula = target ~ zn + nox + rad + tax, family = binomial(link = "logit"),
##     data = crimeTRAIN)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.94915  -0.13567   0.00000   0.00011   3.15933
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.648e+01  3.400e+03  -0.014    0.989
## zn          -6.837e-02  4.509e-02  -1.516    0.129
## nox          5.669e+01  9.605e+00   5.902 3.59e-09 ***
## rad2        -2.026e+00  4.728e+03   0.000    1.000
## rad3         2.013e+01  3.400e+03   0.006    0.995
## rad4         2.244e+01  3.400e+03   0.007    0.995
## rad5         2.015e+01  3.400e+03   0.006    0.995
## rad6         1.830e+01  3.400e+03   0.005    0.996
## rad7         2.465e+01  3.400e+03   0.007    0.994
## rad8         2.573e+01  3.400e+03   0.008    0.994
## rad24        4.309e+01  3.684e+03   0.012    0.991
## tax         -1.602e-02  3.747e-03  -4.277 1.90e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 485.100  on 349  degrees of freedom
## Residual deviance:  94.991  on 338  degrees of freedom
## AIC: 118.99
##
## Number of Fisher Scoring iterations: 19
```

```
##            Reference
## Prediction  0  1
##          0 56  3
##          1  4 53
```

Model summary and confusion matrix of running this model against test data are above. The accuracy rate (0.9396552) is very good and the McFadden R^2 value (0.8041826) is also high. AIC value is 118.99. Additionally, consider the ROC curve for this model.



Area under the curve is 0.9849836.

The third model has the best (lowest) AIC value (better). Accuracy is the same as for the first model (and slightly lower than the second model). AUC is lower, but very close to the AUC value for the second model. Finally, McFadden R^2 falls between the first and second models, but the change is also very small.

Additional Models

Basic models produced very efficient results. Several other models were attempted, but they did not produce significant improvements and therefore simplier, easier to interpret basic models were preferred. Other models included variable transformations and variable interactions. Since this project does not deals with critical and sensitive data with high cost of errors, such as medical or national security projects may, the accuracy of the basic models is deemed acceptable.

Model Selection

All 3 models generated good overall results, but the third model (*StepAIC* model) is chosen for its simplicity. It is important to note that even though general parameters between models are close, one may be preferred over the other based on application. For example, the second and third models have similar number of errors (6 and 7); however, the second model has more Type II/False Negative errors and less Type I/False Positive errors than the third model. This difference in sensitivity and specificity may be important for some applications.

Additionally, one small adjustment to the model is to convert `nox` from parts per 10 million to parts per 100,000. This will help interpret the model coefficients.

```
##
## Call:
## glm(formula = target ~ zn + I(nox * 100) + rad + tax, family = binomial(link = "logit"),
##     data = crime)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9406  -0.1155   0.0000   0.0001   3.3805
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.640e+01  3.102e+03  -0.015   0.9881
## zn           -8.837e-02  4.308e-02  -2.051   0.0402 *
## I(nox * 100)  5.643e-01  8.038e-02   7.021 2.21e-12 ***
## rad2         -1.885e+00  4.225e+03   0.000   0.9996
## rad3          1.987e+01  3.102e+03   0.006   0.9949
## rad4          2.255e+01  3.102e+03   0.007   0.9942
## rad5          2.014e+01  3.102e+03   0.006   0.9948
## rad6          1.865e+01  3.102e+03   0.006   0.9952
## rad7          2.631e+01  3.102e+03   0.008   0.9932
## rad8          2.599e+01  3.102e+03   0.008   0.9933
## rad24         4.410e+01  3.322e+03   0.013   0.9894
## tax          -1.614e-02  3.094e-03  -5.218 1.81e-07 ***
## ---
## 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: 128.29  on 454  degrees of freedom
## AIC: 152.29
##
## Number of Fisher Scoring iterations: 19
```

Model Performance and Description

K-Fold Cross Validation

To assess the performance of selected model, below are results of 10-fold cross-validation. The model performs as expected.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 56  3
##          1  4 53
##
##                Accuracy : 0.9397
##                  95% CI : (0.8796, 0.9754)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8792
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9333
##             Specificity : 0.9464
##          Pos Pred Value : 0.9492
##          Neg Pred Value : 0.9298
##              Prevalence : 0.5172
##          Detection Rate : 0.4828
##    Detection Prevalence : 0.5086
##       Balanced Accuracy : 0.9399
##
##        'Positive' Class : 0
##
```

Deviance

Similarly, the deviance table below demonstrated that each variable significantly contributes to the drop in deviance difference.

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           465     645.88
## zn            1  127.411       464     518.46 < 2.2e-16 ***
## I(nox * 100)  1  230.177       463     288.29 < 2.2e-16 ***
## rad           8  127.537       455     160.75 < 2.2e-16 ***
## tax           1   32.462       454     128.29 1.216e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

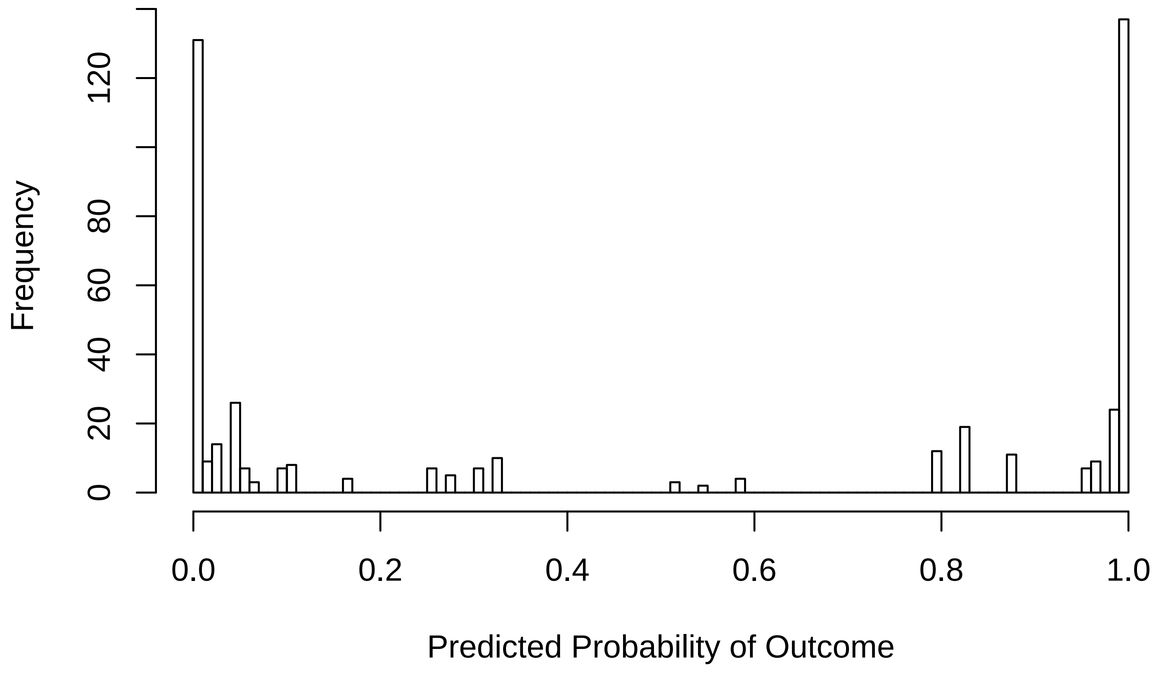
Variance Inflation Factor

```
##                  GVIF Df GVIF^(1/(2*Df))
## zn           2.866869  1        1.693183
## I(nox * 100) 2.953331  1        1.718526
## rad          5.114539  8        1.107390
## tax          2.154850  1        1.467941
```

VIFs are reasonable, so that we can assume that there is not much multicollinearity between variables.

Histogram of Predicted Probabilities

Distribution of predicted probabilities generated by running all training data through the model shows that the model is predicting 0 or 1 with high probability. There are few instances where probability shows less certainty in the selected outcome.



Coefficients/Odds/Variable Contribution

```
##              exp(model$coefficients)
## (Intercept)              7.069802e-21
## zn                       9.154199e-01
## I(nox * 100)             1.758185e+00
## rad2                     1.517949e-01
## rad3                     4.269957e+08
## rad4                     6.223348e+09
## rad5                     5.589707e+08
## rad6                     1.256121e+08
## rad7                     2.665850e+11
## rad8                     1.932043e+11
## rad24                    1.419253e+19
## tax                      9.839877e-01
```

For zn, the coefficient is negative and the odds of having an above median crime rate is 0.9154. Higher zn, meaning more large lots, is less likely to increase the crime rate.

For nox, the coefficient is positive and the odds is 1.75, so there is a 75% increase in odds with higher nox values. Higher levels of nitrogen oxide indicate more congested neighborhoods. It is possible to theorize that more urban, congested areas are more likely to have higher crime than suburban areas.

For tax, the coefficient is negative and the odds is 0.984. Decrease in crime rate is more likely with the increase of property-tax rates.

Finally, for rad all coefficients are positive except for rad value of 2. There is no explicit explanation for values of rad variable. Assuming that low values mean higher accessibility to radial highways, it is possible to theorize that living close, but not too close to highways is more likely to decrease the crime rate, but then moving away from highways is more likely to increase it. Odds are difficult to intepret possibly because of outliers (most likely rad value of 24).

APPENDIX A: Evaluation Data Set

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	prob	predict
0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	4.03	34.7	0.0000	0
0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	10.26	18.2	0.8258	1
0	8.14	0	0.538	6.495	94.4	4.4547	4	307	21.0	12.80	18.4	0.8258	1
0	8.14	0	0.538	5.950	82.0	3.9900	4	307	21.0	27.71	13.2	0.8258	1
0	5.96	0	0.499	5.850	41.5	3.9342	5	279	19.2	8.77	21.0	0.0690	0
25	5.13	0	0.453	5.741	66.2	7.2254	8	284	19.7	13.15	18.7	0.1620	0
25	5.13	0	0.453	5.966	93.4	6.8185	8	284	19.7	14.44	16.0	0.1620	0
0	4.49	0	0.449	6.630	56.1	4.4377	3	247	18.5	6.53	26.6	0.0056	0
0	4.49	0	0.449	6.121	56.8	3.7476	3	247	18.5	8.44	22.2	0.0056	0
0	2.89	0	0.445	6.163	69.6	3.4952	2	276	18.0	11.34	21.4	0.0000	0
0	25.65	0	0.581	5.856	97.0	1.9444	2	188	19.1	25.41	17.3	0.0000	0
0	25.65	0	0.581	5.613	95.6	1.7572	2	188	19.1	27.26	15.7	0.0000	0
0	21.89	0	0.624	5.637	94.7	1.9799	4	437	21.2	18.34	14.3	0.9867	1
0	19.58	0	0.605	6.101	93.0	2.2834	5	403	14.7	9.81	25.0	0.7985	1
0	19.58	0	0.605	5.880	97.3	2.3887	5	403	14.7	12.03	19.1	0.7985	1
0	10.59	1	0.489	5.960	92.1	3.8771	4	277	18.6	17.27	21.7	0.3263	0
0	6.20	0	0.504	6.552	21.4	3.3751	8	307	17.4	3.76	31.5	0.9558	1
0	6.20	0	0.507	8.247	70.4	3.6519	8	307	17.4	3.95	48.3	0.9624	1
22	5.86	0	0.431	6.957	6.8	8.9067	7	330	19.1	3.53	29.6	0.0457	0
90	2.97	0	0.400	7.088	20.8	7.3073	1	285	15.3	7.85	32.2	0.0000	0
80	1.76	0	0.385	6.230	31.5	9.0892	1	241	18.2	12.93	20.1	0.0000	0
33	2.18	0	0.472	6.616	58.1	3.3700	7	222	18.4	8.93	28.4	0.5112	1
0	9.90	0	0.544	6.122	52.8	2.6403	4	304	18.4	5.98	22.1	0.8747	1
0	7.38	0	0.493	6.415	40.1	4.7211	5	287	19.6	6.12	25.0	0.0443	0
0	7.38	0	0.493	6.312	28.9	5.4159	5	287	19.6	6.15	23.0	0.0443	0
0	5.19	0	0.515	5.895	59.6	5.6150	5	224	20.2	10.56	18.5	0.3074	0
80	2.01	0	0.435	6.635	29.7	8.3440	4	280	17.0	5.99	24.5	0.0000	0
0	18.10	0	0.718	3.561	87.9	1.6132	24	666	20.2	7.12	27.5	1.0000	1
0	18.10	1	0.631	7.016	97.5	1.2024	24	666	20.2	2.96	50.0	1.0000	1
0	18.10	0	0.584	6.348	86.1	2.0527	24	666	20.2	17.64	14.5	1.0000	1
0	18.10	0	0.740	5.935	87.9	1.8206	24	666	20.2	34.02	8.4	1.0000	1
0	18.10	0	0.740	5.627	93.9	1.8172	24	666	20.2	22.88	12.8	1.0000	1
0	18.10	0	0.740	5.818	92.4	1.8662	24	666	20.2	22.11	10.5	1.0000	1
0	18.10	0	0.740	6.219	100.0	2.0048	24	666	20.2	16.59	18.4	1.0000	1
0	18.10	0	0.740	5.854	96.6	1.8956	24	666	20.2	23.79	10.8	1.0000	1
0	18.10	0	0.713	6.525	86.5	2.4358	24	666	20.2	18.13	14.1	1.0000	1
0	18.10	0	0.713	6.376	88.4	2.5671	24	666	20.2	14.65	17.7	1.0000	1
0	18.10	0	0.655	6.209	65.4	2.9634	24	666	20.2	13.22	21.4	1.0000	1
0	9.69	0	0.585	5.794	70.6	2.8927	6	391	19.2	14.10	18.3	0.2591	0
0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	5.64	23.9	0.0000	0

Split between predicted outcomes is illustrated by tables below.

```
##
##  0  1
## 19 21
```

```
##
##     0     1
## 0.475 0.525
```

APPENDIX B: R Script

```r
# Required libraries
library(knitr)
library(kableExtra)
library(ggplot2)
library(gridExtra)
library(dplyr)
library(caTools)
library(pscl)
library(ROCR)
library(MASS)
library(caret)
library(car)

# Import data
crime <- read.csv(url(paste0("https://raw.githubusercontent.com/",
                             "ilyakats/CUNY-DATA621/master/hw3/",
                             "crime-training-data_modified.csv")))

# Basic statistic
nrow(crime); ncol(crime)
summary(crime)

# Summary table
sumCrime = data.frame(Variable = character(),
                      Min = integer(),
                      Median = integer(),
                      Mean = double(),
                      SD = double(),
                      Max = integer(),
                      Num_NAs = integer(),
                      Num_Zeros = integer())
for (i in 1:13) {
  sumCrime <- rbind(sumCrime, data.frame(Variable = colnames(crime)[i],
                                         Min = min(crime[,i], na.rm=TRUE),
                                         Median = median(crime[,i], na.rm=TRUE),
                                         Mean = mean(crime[,i], na.rm=TRUE),
                                         SD = sd(crime[,i], na.rm=TRUE),
                                         Max = max(crime[,i], na.rm=TRUE),
                                         Num_NAs = sum(is.na(crime[,i])),
                                         Num_Zeros = length(which(crime[,i]==0)))
  )
}
colnames(sumCrime) <- c("", "Min", "Median", "Mean", "SD", "Max",
                        "Num of NAs", "Num of Zeros")
sumCrime

# Proportion of target variable
table(crime$target)
table(crime$target)/sum(table(crime$target))

# Exploratory plots (repeated for each variable)
```

```r
kable(sumCrime[sumCrime[,1]=="zn",2:8], row.names=FALSE)

# Get descriptive plots:
# Variables: zn indus chas nox rm age dis rad tax ptratio lstat medv target
v <- "dis" # Variable to view
pd <- as.data.frame(cbind(crime[, v], crime$target))
colnames(pd) <- c("X", "Y")

# Boxplot
bp <- ggplot(pd, aes(x = 1, y = X)) +
  stat_boxplot(geom ='errorbar') + geom_boxplot() +
  xlab("Boxplot") + ylab("") +
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank())

# Density plot
hp <- ggplot(pd, aes(x = X)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666") +
  ylab("") + xlab("Density Plot with Mean") +
  geom_vline(aes(xintercept=mean(X, na.rm=TRUE)),
             color="red", linetype="dashed", size=1)

# Scatterplot
sp <- ggplot(pd, aes(x=X, y=Y)) +
  geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE) +
  xlab("Scatterplot with Logistic Regression Line")

grid.arrange(bp, hp, sp, layout_matrix=rbind(c(1,2,2),c(1,3,3)))

# Correlation matrix
cm <- cor(crime, use="pairwise.complete.obs")
cm <- round(cm, 2)
cmout <- as.data.frame(cm) %>% mutate_all(function(x) {
  cell_spec(x, "html", color = ifelse(x>0.5 | x<(-0.5),"blue","black"))
  })
rownames(cmout) <- colnames(cmout)
cmout %>%
  kable("html", escape = F, align = "c", row.names = TRUE) %>%
  kable_styling("striped", full_width = F)

pairs(crime)

# Force categorical variables to factors
crime[,'rad'] <- as.factor(crime[,'rad'])
crime[,'chas'] <- as.factor(crime[,'chas'])

# Split into train and validation sets
set.seed(88)
split <- sample.split(crime$target, SplitRatio = 0.75)
crimeTRAIN <- subset(crime, split == TRUE)
crimeTEST <- subset(crime, split == FALSE)
```

```r
# Model 1
model <- glm (target ~ ., data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
pred <- predict(model, newdata=subset(crimeTEST, select=c(1:12)),
                type='response')
cm <- confusionMatrix(as.factor(crimeTEST$target),
                      as.factor(ifelse(pred > 0.5,1,0)))
cm$table
cm$overall['Accuracy']
pR2(model) # McFadden R^2

# ROC
pr <- prediction(pred, crimeTEST$target)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, colorize = TRUE, text.adj = c(-0.2,1.7))
auc <- performance(pr, measure = "auc")
(auc <- auc@y.values[[1]])


# Model 2
model <- glm (target ~ ., data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
model <- glm (target ~ .-rm, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
model <- glm (target ~ .-rm-chas, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
model <- glm (target ~ .-rm-chas-lstat, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
model <- glm (target ~ .-rm-chas-lstat-indus, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
model <- glm (target ~ .-rm-chas-lstat-indus-zn, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
pred <- predict(model, newdata=subset(crimeTEST, select=c(1:12)),
                type='response')
cm <- confusionMatrix(as.factor(crimeTEST$target),
                      as.factor(ifelse(pred > 0.5,1,0)))
cm$table
cm$overall['Accuracy']
pR2(model) # McFadden R^2

# ROC
pr <- prediction(pred, crimeTEST$target)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, colorize = TRUE, text.adj = c(-0.2,1.7))
auc <- performance(pr, measure = "auc")
(auc <- auc@y.values[[1]])
```

```r
# Take out 'tax' because it is highly correlated with 'rad'
model <- glm (target ~ .-tax, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
pred <- predict(model, newdata=subset(crimeTEST, select=c(1:12)),
                type='response')
cm <- confusionMatrix(as.factor(crimeTEST$target),
                      as.factor(ifelse(pred > 0.5,1,0)))
cm$table
cm$overall['Accuracy']
pR2(model) # McFadden R^2

# ROC
pr <- prediction(pred, crimeTEST$target)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, colorize = TRUE, text.adj = c(-0.2,1.7))
auc <- performance(pr, measure = "auc")
(auc <- auc@y.values[[1]])
# Slight improvement

# Step AIC method
model <- glm (target ~ ., data = crimeTRAIN,
              family = binomial(link="logit"))
model <- stepAIC(model, trace=TRUE)
summary(model)
pred <- predict(model, newdata=subset(crimeTEST, select=c(1:12)),
                type='response')
cm <- confusionMatrix(as.factor(crimeTEST$target),
                      as.factor(ifelse(pred > 0.5,1,0)))
cm$table
cm$overall['Accuracy']
pR2(model) # McFadden R^2

# ROC
pr <- prediction(pred, crimeTEST$target)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, colorize = TRUE, text.adj = c(-0.2,1.7))
auc <- performance(pr, measure = "auc")
(auc <- auc@y.values[[1]])


# Bad model for testing of code
model <- glm (target ~ age+tax, data = crimeTRAIN,
              family = binomial(link="logit"))
summary(model)
pred <- predict(model, newdata=subset(crimeTEST, select=c(1:12)),
                type='response')
cm <- confusionMatrix(as.factor(crimeTEST$target),
                      as.factor(ifelse(pred > 0.5,1,0)))
cm$table
cm$overall['Accuracy']
pR2(model) # McFadden R^2
```

```
# ROC
pr <- prediction(pred, crimeTEST$target)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, colorize = TRUE, text.adj = c(-0.2,1.7))
auc <- performance(pr, measure = "auc")
(auc <- auc@y.values[[1]])

# Selected model
model <- glm(target ~ zn+I(nox*100)+rad+tax+indus, data = crimeTRAIN,
             family = binomial(link = "logit"))

# K-Fold cross validation
ctrl <- trainControl(method = "repeatedcv", number = 10,
                     savePredictions = TRUE)
model_fit <- train(target ~ zn + nox + rad + tax,
                   data=crimeTRAIN, method="glm",
                   family="binomial",
                   trControl = ctrl, tuneLength = 5)

pred <- predict(model_fit, newdata=crimeTEST)
confusionMatrix(as.factor(crimeTEST$target),
                as.factor(ifelse(pred > 0.5,1,0)))

# Deviance residuals
anova(model, test="Chisq")

# VIF
vif(model)

# Prediction
eval <- read.csv("crime-evaluation-data_modified.csv")
eval[,'rad'] <- as.factor(eval[,'rad'])
eval[,'chas'] <- as.factor(eval[,'chas'])

pred <- predict(model, newdata=eval, type="response")

eval <- cbind(eval, prob=round(pred,4))
eval <- cbind(eval, predict=round(pred,0))
kable(eval, row.names=FALSE)
```