GROUP 2 -HOMEWORK #3

Data 621 – Business Analytics
Logistic Regression
April 9, 2018

Group 2
Sharon Morris
Brian Kreis
Keith Folsom
Michael D'Acampora
Valerie Briot

Objective

Our Data Analytics team has been tasked with the creation of model that will be used to predict whether a neighborhood will be at risk for high crime levels. This model could presumably be used to deploy the crime prevention resources of a municipality more effectively by targeting the most at risk neighborhoods.

Our objective is to predict whether a neighborhood will have a higher than median crime rate (1) or not (0). Since we are attempting to predict a binary outcome, we will build a binary logistic regression model on the provided data to predict whether the neighborhood will be at risk for high crime levels.

Approach

The team met to discuss the project and organized ourselves into task groups to be able to produce the deliverable on time.

The following tasks were assigned:

Data Exploration & Data Preparation
Since the data sets were provided, it was crucial that we understand the data set and determine whether any missing values are present.

Model Building & Model Selection
We will develop multiple models and ensure that the model selection takes into consideration the business requirements.

Github was used to manage the project. Using Github helped with version control and ensured each team member had access to the latest version of the project documentation.

Slack was used for daily communication during the project and for quick access to code and documentation. Meetings were organized at least twice a week and as needed using "Go to Meeting".

We are using R to perform analysis. The R code can be found in Appendix B.

Team Members
- Sharon Morris
- Brian Kreis
- Keith Folsom
- Michael D'acampora
- Valerie Briot

Dataset

For reproducibility of the results, the data was loaded to and accessed from a Github repository. The age variable was rounded to a whole number. The training data set has 13 variables (including the outcome variable) and 466 observations.

Data Exploration

Basic Data Exploration and Statistic measures

Basic statistic measures

The following variables comprise the data set. The response variable (Target) is the variable of interest. The response variable is binary (0, 1) and identifies whether the crime rate is above the median crime rate. The remaining 12 variables are predictors. All variables are numeric.

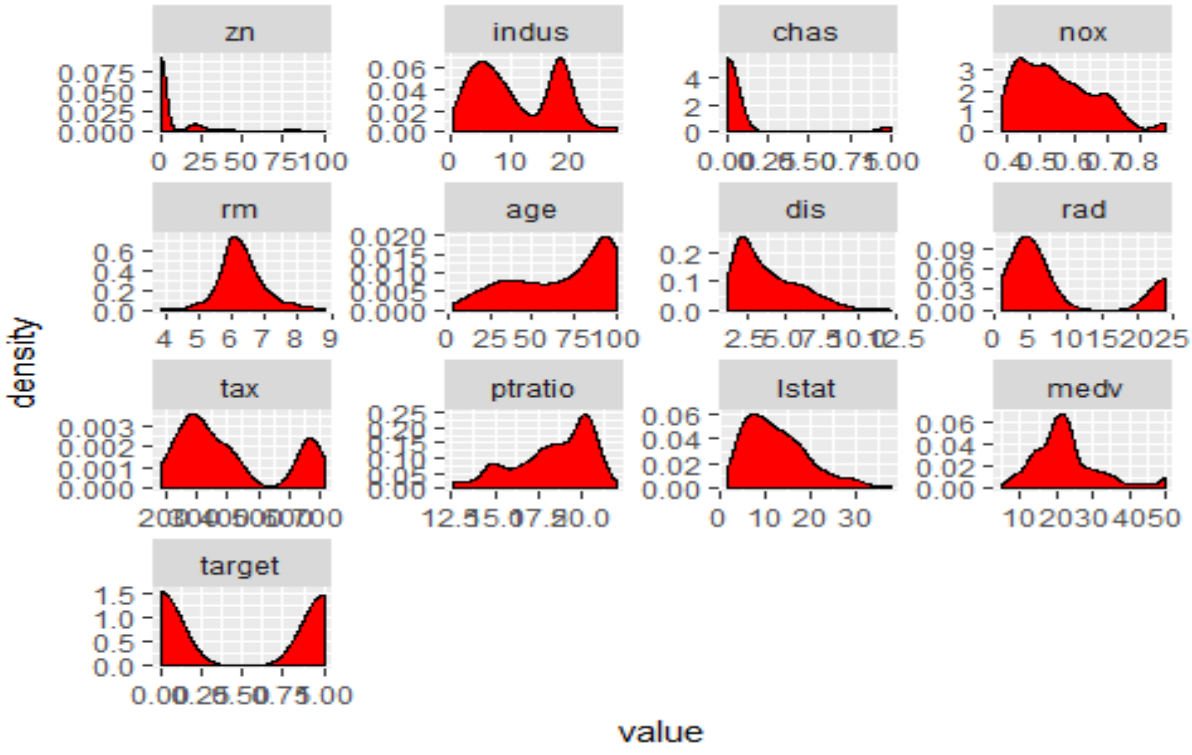
Variable Name	Definition	Variable Type	Data Type
zn	proportion of residential land zoned for large lots (over 25000 square feet)	Predictor	quantitative
indus	proportion of non-retail business acres per suburb	Predictor	quantitative
chas	a dummy var. for whether the suburb borders the Charles River (1) or not (0)	Predictor	categorical
nox	nitrogen oxides concentration (parts per 10 million)	Predictor	quantitative
rm	average number of rooms per dwelling	Predictor	quantitative
age	proportion of owner-occupied units built prior to 1940	Predictor	quantitative
dis	weighted mean of distances to five Boston employment centers	Predictor	quantitative
rad	index of accessibility to radial highways	Predictor	quantitative
tax	full-value property-tax rate per $10,000	Predictor	quantitative
ptratio	pupil-teacher ratio by town	Predictor	quantitative
lstat	lower status of the population (percent)	Predictor	quantitative
medv	median value of owner-occupied homes in $1000s	Predictor	quantitative
target	whether the crime rate is above the median crime rate (1) or not (0)	Response	Categorical

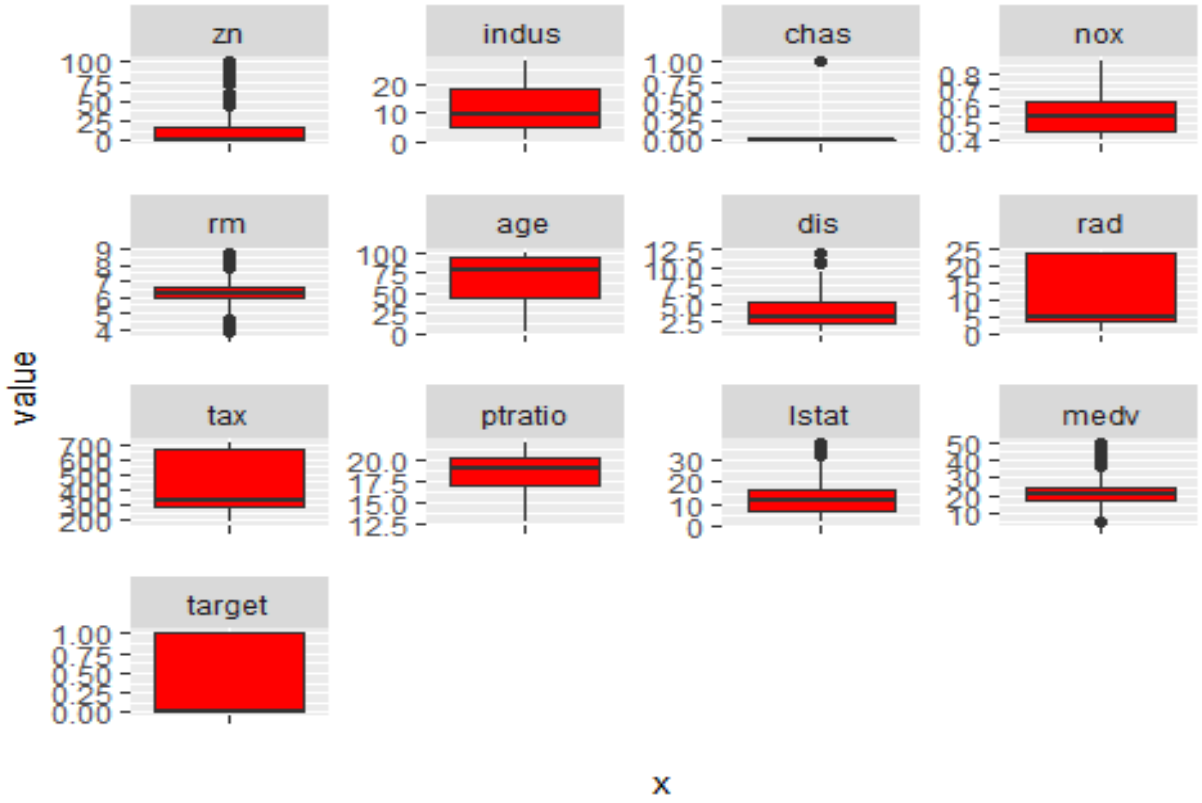
Descriptive statistics were calculated to examine the basic features of the data. Each variable has 466 observations. At first glance, we do not have missing data.

	vars	n	mean	sd	skew	kurtosis	se	IQR	Q0.25	Q0.75
zn	1	466	11.577	23.365	2.1768	3.8136	1.0823	16.25	0	16.25
indus	2	466	11.105	6.8459	0.2885	-1.243	0.3171	12.955	5.145	18.1
chas	3	466	0.0708	0.2568	3.3355	9.1451	0.0119	0	0	0
nox	4	466	0.5543	0.1167	0.7463	-0.036	0.0054	0.176	0.448	0.624
rm	5	466	6.2907	0.7049	0.4793	1.5424	0.0327	0.7425	5.8873	6.6298
age	6	466	68.35	28.324	-0.577	-1.013	1.3121	50	44	94
dis	7	466	3.7957	2.1069	0.9989	0.472	0.0976	3.1132	2.1014	5.2146
rad	8	466	9.53	8.6859	1.0103	-0.862	0.4024	20	4	24
tax	9	466	409.5	167.9	0.6593	-1.148	7.7778	385	281	666
ptratio	10	466	18.398	2.1968	-0.754	-0.4	0.1018	3.3	16.9	20.2
lstat	11	466	12.631	7.1019	0.9056	0.5034	0.329	9.8875	7.0425	16.93
medv	12	466	22.589	9.2397	1.0767	1.3738	0.428	7.975	17.025	25
target	13	466	0.4914	0.5005	0.0342	-2.003	0.0232	1	0	1

From the skewness coefficient and the kurtosis, it appears that variables zn, chas, rad, and medv show some skewness. We will now look at the density plots and box plots for better insight into each variable distribution.

Density and Box Plots





The density plot of predictor variables confirms that the zn, chas, dis, lstat predictor variables are highly skewed. The rm variable is the only predictor that is normally distributed. The Box Plots also show the presence of some outliers.

We will take a closer look at the possible outliers for each variable.

Outliers

zn

This variable is highly skewed to the left. The range is from 85-100.

Outliers for zn: 100, 95, 90, 85, 82.5

indus

This predictor variable is bi-modal.

Outliers for indus: none

nox

This variable is skewed to the left.

Outliers for nox: none

rm

Outliers for rm: 8.78, 8.725, 8.704, 4.138, 3.863

age

Outliers for age: none

dis

Outliers for dis: 12.1265, 10.7103, 10.5857

rad

Outliers for rad: none

tax

Outliers for tax: none

ptratio

Outliers for ptratio: none

lstat

Outliers for lstat: 37.97, 36.98, 34.77, 34.41, 34.37

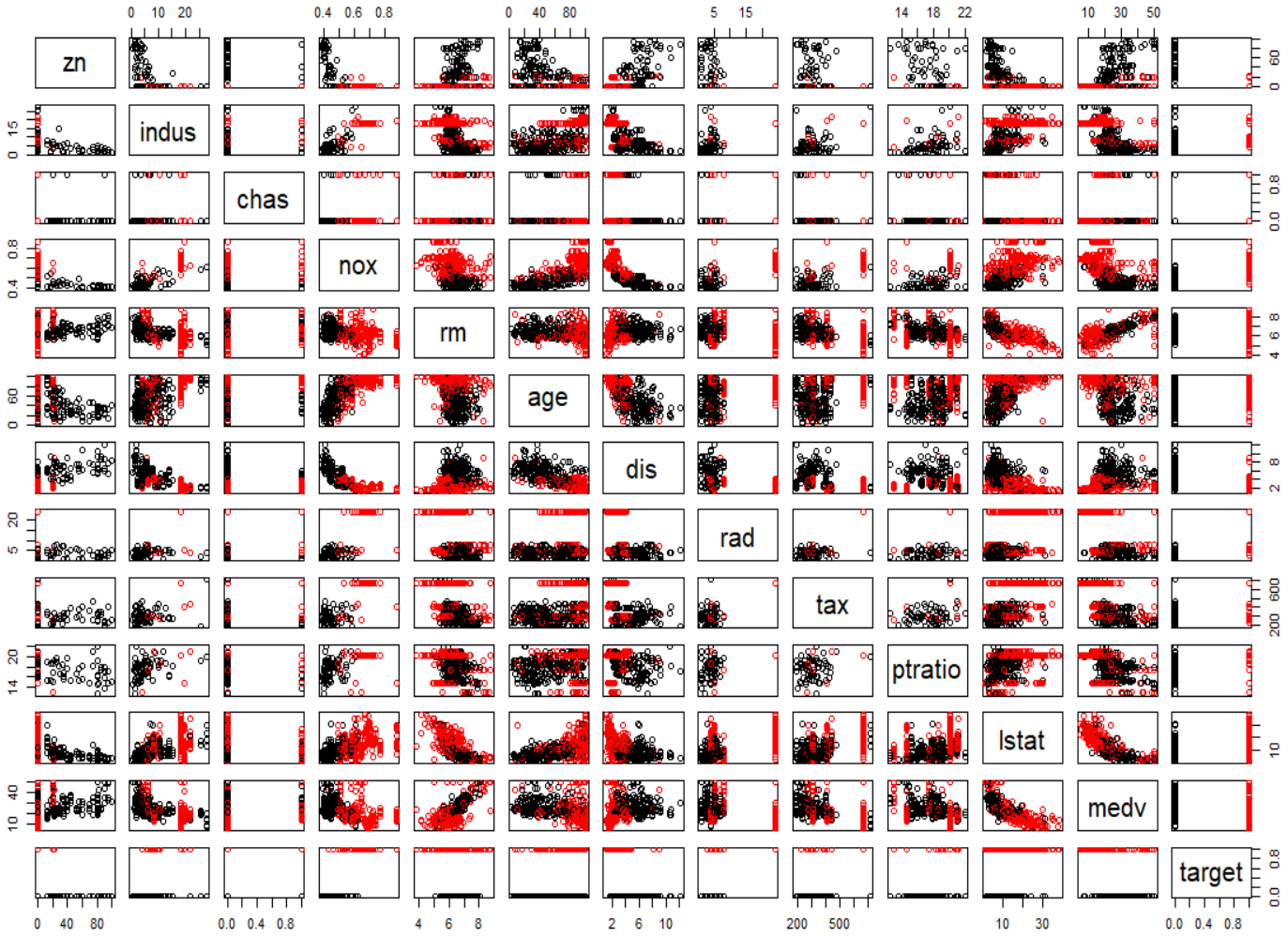
medv

Outliers for lstat :

This completes our univariate exploratory data analysis. We will now look at variables with respect to each other.

Variable-to-Variable Analysis

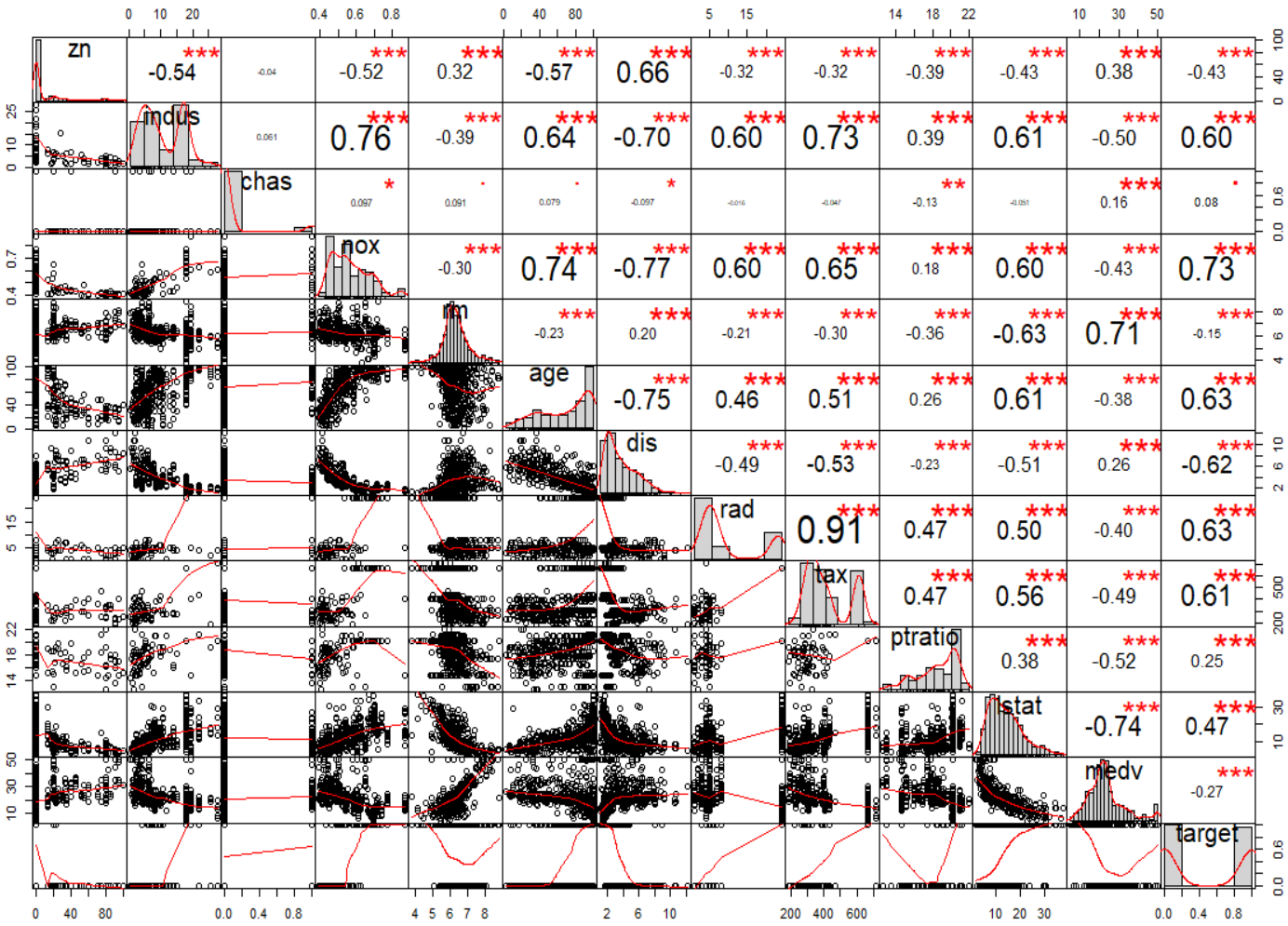
We will now look at all the predictor variables compared to each other and the response, with red values in the scatter plots are showing observations where the crime rate exceeded the median.

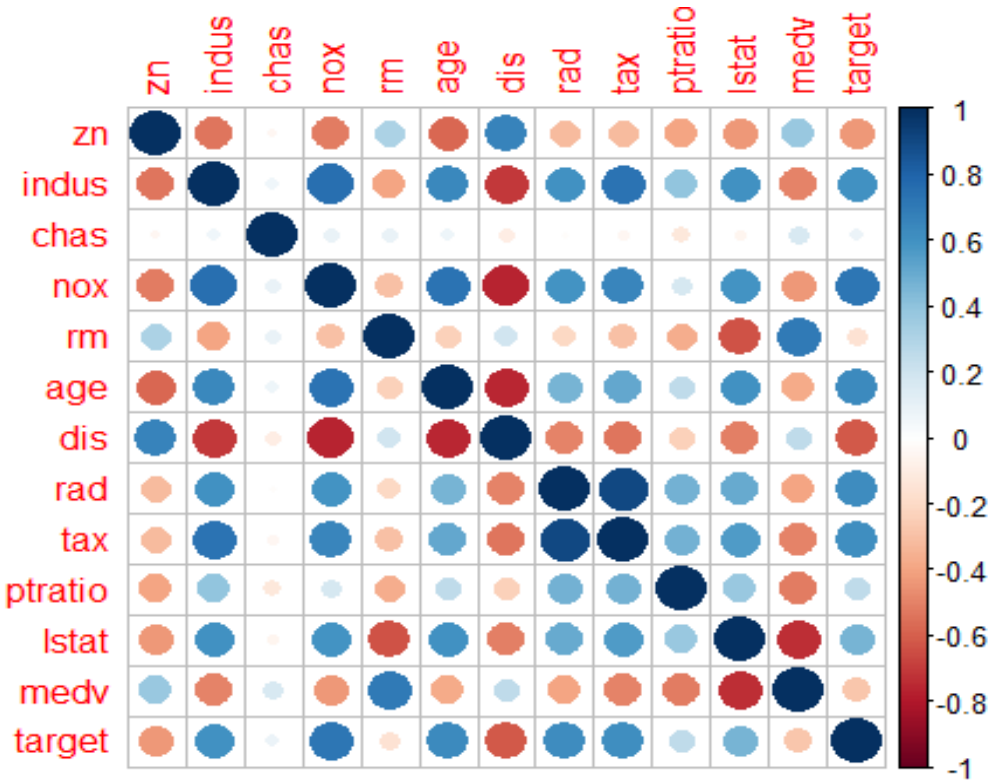


From the scatter plots, we are observing that some predictors are showing relationship with each other such as; (nox; dis), (age, dis), (rm, lstat), (rm, medv), (lstat, medv). This could indicate correlation between variables.

We will now look at the correlation between variables.

Correlation between Variables





	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	0.0	-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	0.0	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

rad and tax have the strongest positive correlation while nox and dis have the strongest negative correlation.

Multicollinearity

This section will test the predictor variables to determine if there is correlation among them. Variance inflation factor (VIF) is used to detect multicollinearity, specifically among the entire set of predictors versus within pairs of variables.

Testing for collinearity among the predictor variables, we see that the following variables may have a problem with collinearity based on their high VIF scores.

Variable Name	VIF
tax	9.217602
nox	4.504675
dis	4.243532
lstat	3.650759
medv	3.667409
indus	4.120617
age	3.142118
ptratio	2.013194

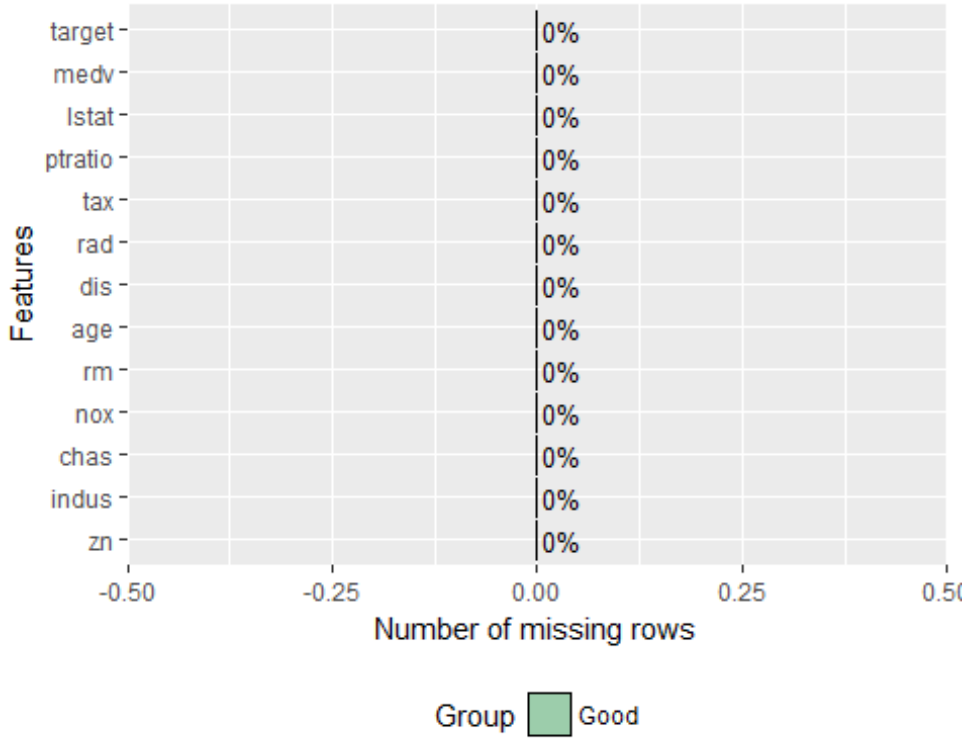
If we set our VIF threshold at 4, the following predictor variables are highly correlated.

Variable Name	VIF
indus	4.120617
dis	4.243532
nox	4.504675
rad	6.782250
tax	9.217602

Data Preparation

There are no NA values in the data; however, it is possible that zero values in a data set may be equivalent to missing information. For instance, we would not expect to see any observation where the average number of rooms per dwelling is equal to zero.

We look at the dataset to determine if zero values exist for each variable and check for reasonableness.



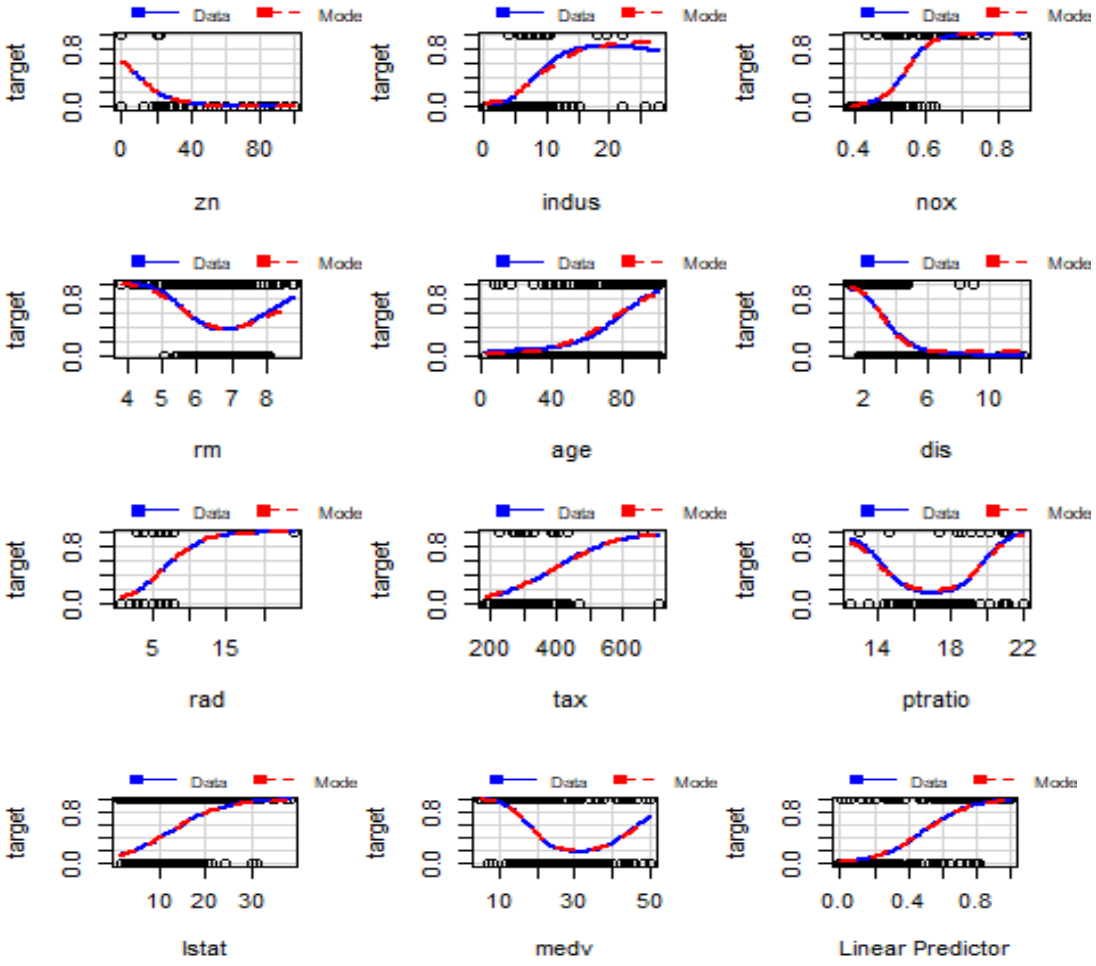
	Count of Zero Values
zn	339
indus	0
chas	433
nox	0
rm	0
age	0
dis	0

rad	0
tax	0
ptratio	0
lstat	0
medv	0
target	237

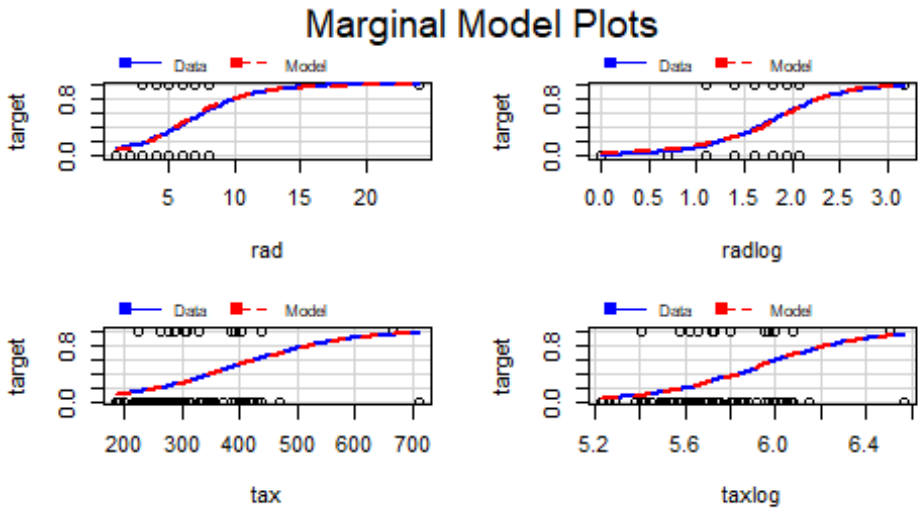
It is reasonable that there could be no land zoned for large lots (zn) in a particular suburb. The chas variable is a binary variable that tells us whether a suburb borders the Charles river, with zero meaning no. The target variable is also binary. It is also feasible that the other variables would not necessarily contain zero values. It appears that this data set did not contain any missing values.

Transformations

In the case of logistic regression, transformations are not necessary as normality of predictors is not required. We can compare the independent variable itself to the dependent variable using marginal model plots to help us determine if transformation improves the fit between the predictor and response.



Two which stand out are rad (index of accessibility to radial highways) and tax (full-value property-tax rate per $10,000). Outliers seem to be having a large impact, with the majority of the data falling to the left side of the plots. We can transform and then compare the use of the transformed variable and the original in our models.



Above shows the marginal model plots before (left) and after (right) transformation. It appears that our fit has improved. We will determine if this improves the overall model in the next section.

Model Building

Model 1: Baseline using all Predictor Variables

As a baseline, the first model build will be a logistic regression model using all predictor variables provided. No transformation has been performed on the predictor variables.

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = dev_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.88220  -0.10094  -0.00031   0.00027   2.94182
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -49.184177   9.422981  -5.220 1.79e-07 ***
## zn           -0.076954   0.044453  -1.731 0.083427 .
## indus        -0.044532   0.064952  -0.686 0.492959
## chas          1.226574   1.082934   1.133 0.257366
## nox          52.509712  10.990233   4.778 1.77e-06 ***
## rm           -0.861188   0.913113  -0.943 0.345612
## age           0.064011   0.019536   3.277 0.001051 **
## dis           0.953227   0.295664   3.224 0.001264 **
## rad           0.976962   0.228521   4.275 1.91e-05 ***
## tax          -0.007383   0.003829  -1.928 0.053857 .
## ptratio       0.623825   0.180634   3.454 0.000553 ***
## lstat        -0.031103   0.064727  -0.481 0.630856
```

```
## medv          0.214037   0.088075   2.430 0.015091 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 118.33  on 314  degrees of freedom
## AIC: 144.33
##
## Number of Fisher Scoring iterations: 9
```

As we can see in our first model, `zn`, `indus`, `chas`, `rm`, `tax`, and `lstat` are not statistically significant. As for the statistically significant variables, `nox` and `rad` have the lowest p-values suggesting a strong association between nitrogen oxide concentration and accessibility to radial highways with the probability of crime rates above the median.

Recall that the estimates from logistic regression characterize the relationship between the predictor and response variable on a log-odds scale. This suggests that for every one unit increase in `nox`, the log-odds of the crime rate increases significantly in orders of magnitude. Access to radial highways, while not nearly to the same magnitude, also increases the log-odds of crime above the median.

It is interesting to note that that `nox` is a significant predictor of crime by orders of magnitude when compared to the other significant predictors. NOx (nitrogen dioxide and nitric oxide) are typically associated with smog and acid rain pollution. NOx has been linked to adverse health effects in humans.

AIC (Akaike Information Criterion) for Model 1 = 144.3266013
BIC (Bayesian Information Criterion) for Model 1 = 193.5960836

Model 2: Baseline using Transformed Variables

In the data preparation section, the log transformation of the `rad` and `tax` predictor variables were determined to be potentially beneficial transformations. This model will use those transformed variables and repeat the modeling process in Model 1.

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2127  -0.0724   0.0000   0.0314   4.0565
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -32.82833   11.11838  -2.953  0.00315 **
## zn           -0.07365    0.04891  -1.506  0.13216
## indus        -0.12025    0.06622  -1.816  0.06939 .
```

```
## chas         -0.25948    0.98736  -0.263  0.79270
## nox          65.61314   12.79735   5.127 2.94e-07 ***
## rm           -0.45659    0.97326  -0.469  0.63898
## age           0.06588    0.02044   3.223  0.00127 **
## dis           0.71536    0.31357   2.281  0.02253 *
## rad           4.36904    1.05771   4.131 3.62e-05 ***
## tax          -3.94655    1.59989  -2.467  0.01363 *
## ptratio       0.44875    0.16876   2.659  0.00784 **
## lstat        -0.08538    0.08047  -1.061  0.28865
## medv          0.11886    0.09095   1.307  0.19123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 117.83  on 314  degrees of freedom
## AIC: 143.83
##
## Number of Fisher Scoring iterations: 8
```

Contrasting against model 1, we now see that `nox`, `age`, and `rad` (log-transformed) are now the most statistically significant variables with `dis`, `tax` (log-transformed), and `ptratio` showing some significance but to a lesser degree.

Model 2 sees an uptick in significance in the tax variable, and the new taxlog variable has one of the lowest p-values suggesting a strong association between property tax rate and crime rates. Of interest here is that this is the only predictor variable which is showing a log-odds decrease in crime for a unit increase in the tax rate.

`ptratio`, the pupil-teacher ratio by town, also saw an increase in significance when running model 2 with the transformed data.

AIC (Akaike Information Criterion) for Model 2 = 143.8252129
BIC (Bayesian Information Criterion) for Model 2 = 193.0946951

Model 1 - Model 2 Comparison

Comparing the two models using a Chi-square test, there's no significance difference detected between the two. However, we do see that Model 2 resulted in a slightly lower AIC value. Consequently, further modeling will be based on the transformed dataset.

```
## Analysis of Deviance Table
##
## Model 1: target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
## Model 2: target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
## 1       314     118.33
## 2       314     117.83  0  0.50139
```

Model 3: AIC Stepwise Variable Selection

The third model used was a stepwise regression, and we chose to use both the "forward" and "backward" methods to obtain the optimal model. As mentioned previously, we have chosen to use the transformed dataset going forward.

After starting from nothing and adding variables one at a time, then repeating the process backwards starting with a full dataset and subtracting variables one at a time, the ideal model chosen included `zn`, `indus`, `nox`, `age`, `dis`, `rad`, `tax`, `ptratio`, and `medv`, with `nox`, `age`, and `rad` having the most statistical significance as shown by the summary below.

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + dis + rad + tax +
##     ptratio + medv, family = binomial(), data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
```

```
## Response: target
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev
## NULL                       326     453.24
## zn        1   83.250       325     369.99
## indus     1   53.372       324     316.62
## nox       1  128.592       323     188.03
## age       1    3.247       322     184.78
## dis       1    4.104       321     180.68
## rad       1   43.048       320     137.63
## tax       1    8.686       319     128.94
## ptratio   1    3.956       318     124.99
## medv      1    5.840       317     119.15
```

AIC (Akaike Information Criterion) for Model 2 = 139.1484546
BIC (Bayesian Information Criterion) for Model 2 = 177.0480563

Model 4: Using VIF Reduction with Transformed Predictor Variables

Since multicollinearity was detected during the EDA phase, Model 4 will select meaningful variables using VIF reduction. The presence of multicollinearity among predictors can lead to overfitting so this modeling approach will attempt to limit that by reducing the predictor variables to those with lower magnitude VIF.

Calculating and reviewing VIF for the predictor variables (below):

Variable	VIF
medv	7.713861
rm	5.607293
nox	4.525322
tax	3.592654
rad	2.975277
indus	2.968206
dis	2.940839
lstat	2.932736
age	2.596930
ptratio	2.214176

zn	1.599503
chas	1.367180

We see that nox, rm, and medv have the high variance inflation factor. However, knowing the significance of nox, we'll keep this variable as a predictor and update the model to remove rm and medv.

In the summary of model 4, several variables are not statistically significant and will be dropped from the final model 4.

Dropped Variables
* zn
* chas
* dis
* ptratio
* lstat

```
##
## Call:
## glm(formula = target ~ indus + nox + age + rad + tax, family = binomial(),
##     data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2531  -0.1493  -0.0013   0.0376   3.2493
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.84525    6.08740  -2.274  0.02294 *
## indus        -0.12492    0.06029  -2.072  0.03828 *
## nox          50.04512   10.67525   4.688 2.76e-06 ***
## age           0.03943    0.01327   2.970  0.00297 **
## rad           3.66138    0.76378   4.794 1.64e-06 ***
## tax          -3.59863    1.24154  -2.899  0.00375 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 135.94  on 321  degrees of freedom
## AIC: 147.94
##
## Number of Fisher Scoring iterations: 8
```

AIC (Akaike Information Criterion) for Model 4 = 147.9402702
BIC (Bayesian Information Criterion) for Model 4 = 170.6800312

Model 5: Using BestGlm using Transformed Predictors

In the final model build the `bestglm` R package is used to determine the best set of predictors using both AIC and BIC as selection criteria.

Using Alkaike Information Criterion (AIC)

```
## Morgan-Tatar search since family is non-gaussian.
```

Looking at the top 5 best models based on lowest AIC, the variables `zn`, `indus`, `nox`, `age`, `dis`, `rad`, `tax`, `ptratio`, and `medv` are selected. AIC values for the top 5 models are shown below:

Model	Criterion
1	137.1485
2	138.1581
3	138.3375
4	138.5338
5	138.9625

The resulting model based on lowest AIC is not dissimilar from previous models. We see `nox`, `age`, and `rad` (again log-transformed) as the most significant predictors.

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
```

```
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8
```

Using Bayesian Information Criterion (BIC)

Calculate the best set of predictors using Bayesian Information Criterion (BIC). The model with the lowest BIC will be selected.

```
## Morgan-Tatar search since family is non-gaussian.
```

Looking at the top 5 best models based on lowest BIC, the variables `indus`, `nox`, `age`, `rad`, and `tax` are selected. The BIC values for the top 5 models are shown below:

Model	Criterion
1	163.5534
2	163.8524
3	163.9032
4	164.8901
5	165.3483

It should be noted that this model based on BIC uses the fewest number of predictors compared to the other model builds. The inclusion of the `indus` variable has a marginal effect on BIC so for simplicity of the second-best model will be used.

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax, family = binomial(),
##     data = dev_train_T)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

```
## -1.93712  -0.17367  -0.00171   0.06190   3.05710
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.43708    4.21211  -1.291  0.19677
## nox         36.91089    6.98000   5.288 1.24e-07 ***
## age          0.03843    0.01301   2.954  0.00313 **
## rad          3.99417    0.77465   5.156 2.52e-07 ***
## tax         -4.13436    1.09974  -3.759  0.00017 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 140.74  on 322  degrees of freedom
## AIC: 150.74
##
## Number of Fisher Scoring iterations: 8
```

The resulting BIC model uses `nox`, `age`, `rad`, and `tax` as the final set of predictors. All are statistically significant.

Model Selection and Evaluation

Model Selection

We will use a structured evaluation of the models on validation data set (we split our training data set between a training set and a model evaluation set) with regards to:

i. parsimonious fit,
ii. goodness-of-fit,
iii. predictive accuracy, and
iv. more subjectively satisfying business requirements

(i) Parsimony

Parsimonious models have optimal parsimony, or just the right number of predictors needed to explain the model well. There is generally a tradeoff between goodness-of-fit and parsimony: low parsimony models tend to have a better fit than high parsimony models.

We will use Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC)

> BIC = LN(number of observations) * number of variables in your model- 2 Log Likelihood
>
> AIC = 2*number of variables in your model = 2 Log Likelihood

(ii) Goodness-of-fit

The Goodness-of-fit of a model describes how well it fits a set of observations. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the model in question.

We will use McFadden's R^2 and the Hosmer-Lemeshow test

McFadden's R^2: Higher value (0.2 to 0.4) indicates a good fit

Hosmer_Lemeshow Test: Small values with large p-values indicate a good fit to the data while large values with p-values below 0.05 indicate a poor fit.

(iii) Predictive accuracy

Predictive accuracy of a model is how well a model is predicting correctly the outcome and is also a measure of the incorrect predictions.

We will use Cohen's Kappa (or Kappa), Youden's Index, F1_Score, Percentage of False Positive, and AUC/ROC Curves

Kappa

Kappa takes into consideration the accuracy that would be generated purely by chance. The form of the measure is:

$$Kappa = \frac{TotalAccuracy - RandomAccuracy}{1 - RandomAccuracy}$$

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

$$RandomAccuracy = \frac{(TN + FP)(TN + FN) + (FN + TP)(FP + TP)}{(TP + TN + FP + FN)^2}$$

Kappa takes on values from -1 to +1, with a value of 0 meaning there is no agreement between the actual and classified classes. A value of 1 indicates perfect concordance of the model prediction and the actual classes and a value of 0 indicates total disagreement between prediction and the actual

Youden's Index

Youden's index evaluates the ability of a classifier to avoid misclassifications. This index puts equal weights on a classifier's performance on both the positive and negative cases. Thus:

$$Youden'sIndex(\gamma) = Sensitivity - (1 - Specificity)$$

We selected to look at False Positive instead of classification error rate since we think this measure is better aligned with the business requirements.

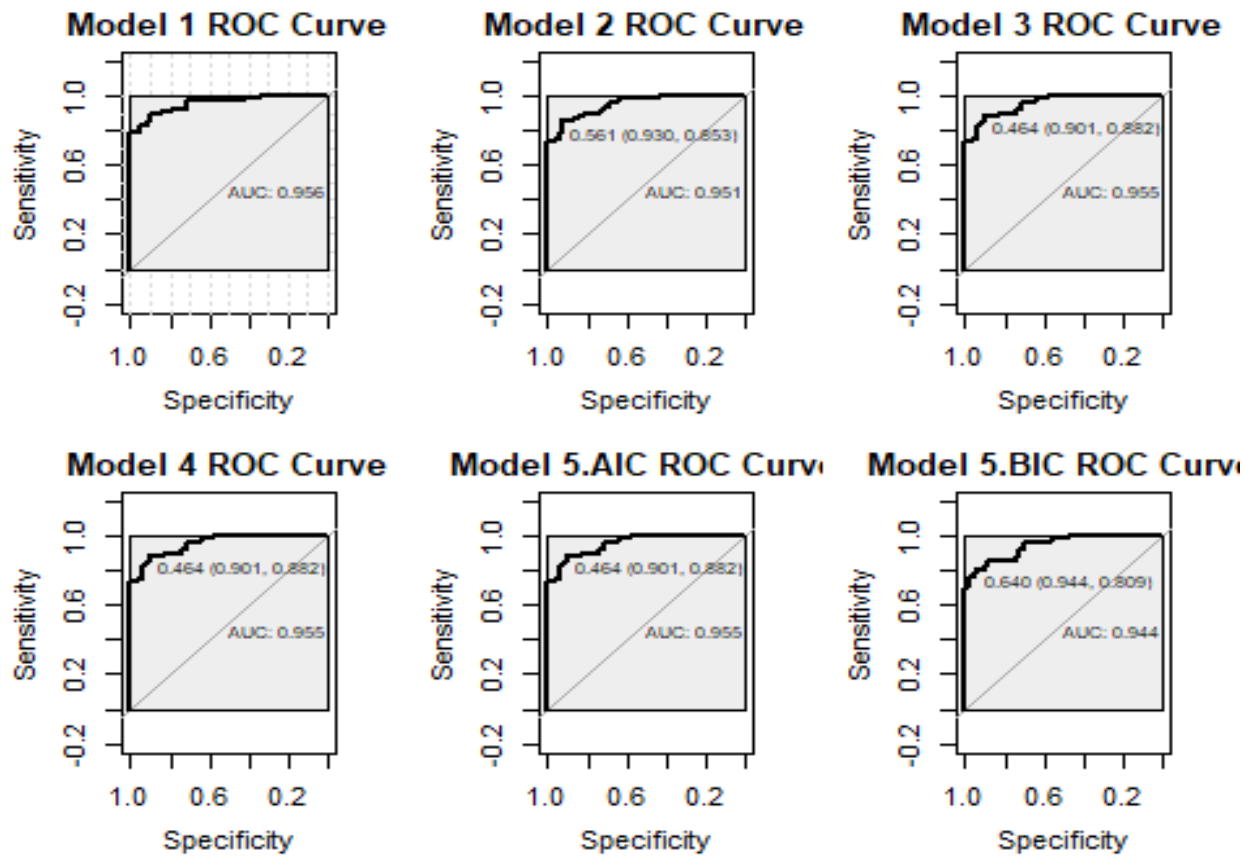
Model	AIC	BIC	McFadenR2	HL_Chi	HL_p	X	Kappa	Youden	F1Score	FPPrct	AUC
Model1	144.33	193.6	0.739	327	0	*	0.755	0.757	0.872	7.19	0.956
Model2	143.83	193.1	0.739	327	0	*	0.755	0.757	0.872	7.19	0.951
Model3	139.15	177.05	0.739	327	0	*	0.755	0.757	0.872	7.19	0.955
Model4	147.94	170.68	0.739	327	0	*	0.74	0.747	0.862	8.63	0.942
Model5.AIC	139.15	177.05	0.739	327	0	*	0.755	0.757	0.872	7.19	0.955
Model5.BIC	150.74	169.69	0.739	327	0	*	0.712	0.718	0.846	9.35	0.944

Note: The 'X' column indicates that the p-value was smaller than 2.2 x 10^{-16}

From the various measurements matrix, we noticed that some of the measures do not come into play since they do not differentiate any of our models: McFadden R^2 and Hosmer-Lemeshow test.

The remaining measures clearly indicate that Model3 and Model5.AIC are superior models.

Let us now consider the ROC curves for all the models.



The side-by-side comparison of the ROC curve is showing the trade-off between Sensitivity and Specificity. The closer the area under to 1, the better fit of the model. The ROC Curves plot support our selection of Model 3 or Model 5

We will compare the 2 models.

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + dis + rad + tax +
##     ptratio + medv, family = binomial(), data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
```

```
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8

##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8
```

Side-by-side comparison reveals that these two models are actually the same. Since both models were built based on best AIC score, this is understandable.

We will recommend one of them as our best model: **model5.AIC**.

Evaluation

We will now run our model against our evaluation data set. However, before we can do so, we need to transform our evaluation data set since Model5.AIC contains transformed predictors.

Load & Transformation of Data Set

Below is a summary of the evaluation data set. There are no missing values. There are only 40 observations in the evaluation data set.

```
##        zn             indus             chas             nox        
##  Min.   : 0.000   Min.   : 1.760   Min.   :0.00   Min.   :0.3850  
##  1st Qu.: 0.000   1st Qu.: 5.692   1st Qu.:0.00   1st Qu.:0.4713  
##  Median : 0.000   Median : 8.915   Median :0.00   Median :0.5380  
##  Mean   : 8.875   Mean   :11.507   Mean   :0.05   Mean   :0.5592  
##  3rd Qu.: 0.000   3rd Qu.:18.100   3rd Qu.:0.00   3rd Qu.:0.6258  
##  Max.   :90.000   Max.   :25.650   Max.   :1.00   Max.   :0.7400  
##        rm              age              dis              rad        
##  Min.   :3.561   Min.   :  7.00   Min.   :1.202   Min.   : 1.000  
##  1st Qu.:5.874   1st Qu.: 56.75   1st Qu.:2.041   1st Qu.: 4.000  
##  Median :6.143   Median : 83.00   Median :3.373   Median : 5.000  
##  Mean   :6.214   Mean   : 71.00   Mean   :3.787   Mean   : 9.775  
##  3rd Qu.:6.532   3rd Qu.: 93.00   3rd Qu.:4.527   3rd Qu.:24.000  
##  Max.   :8.247   Max.   :100.00   Max.   :9.089   Max.   :24.000  
##       tax           ptratio          lstat            medv      
##  Min.   :188.0   Min.   :14.70   Min.   : 2.960   Min.   : 8.40  
##  1st Qu.:276.8   1st Qu.:18.40   1st Qu.: 6.435   1st Qu.:16.98  
##  Median :307.0   Median :19.60   Median :11.685   Median :20.55  
##  Mean   :393.5   Mean   :19.12   Mean   :12.905   Mean   :21.88  
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:17.363   3rd Qu.:25.00  
##  Max.   :666.0   Max.   :21.20   Max.   :34.020   Max.   :50.00
```

We will now run the prediction on our transformed evaluation data set and write the results to a .csv file. The prediction will also be in Appendix A.

Our classification predictions, which are based on a probability threshold of 0.5, indicate that exactly half of the neighborhoods represented in the evaluation set would be identified as high crime areas (above the median crime rate) based on our model.

Conclusion

The data we were provided with was relatively straight forward, there was no missing data and we did not find that there were any additional variables that could be derived from the data. We applied two log transformations to improve the distribution of the most skewed predictors without making the final model too difficult to interpret.

We took measures to avoid possible overfitting with our inclusion of parsimonious measures in the model selection process and our use of AIC to guide our selection of variables.

We split the training data set to reserve a subset for model evaluation and used predictive measures to help select the best model. Based on the thoroughness of our model creation and selection process, we are confident in our final result.

Area for Further Study:

One area of concern is that our test-evaluation data set happened to provide results that are not applicable to another evaluation set. This could have been alleviated by adopting a K-Fold Cross Validation method with randomization to prevent overfitting.

Reference

https://www.researchgate.net/post/Should_I_transform_non-normal_independent_variables_in_logistic_regression
http://www.statisticshowto.com/parsimonious-model/
http://thestatsgeek.com/2014/02/16/the-hosmer-lemeshow-goodness-of-fit-test-for-logistic-regression/
https://www.r-bloggers.com/logistic-regression-in-r-part-two/
https://www.r-bloggers.com/evaluating-logistic-regression-models/
http://support.sas.com/resources/papers/proceedings17/0942-2017.pdf

Appendix A – Prediction results

observation	probability	classification
1	0.005	0
2	0.776	1
3	0.861	1
4	0.556	1
5	0.057	0
6	0.109	0
7	0.236	0
8	0.003	0
9	0.001	0
10	0.000	0
11	0.247	0
12	0.184	0
13	0.900	1
14	0.761	1
15	0.686	1
16	0.091	0
17	0.116	0
18	0.955	1
19	0.004	0
20	0.000	0
21	0.000	0
22	0.032	0
23	0.099	0
24	0.083	0
25	0.059	0
26	0.808	1
27	0.000	0
28	1.000	1
29	1.000	1
30	0.995	1
31	1.000	1
32	1.000	1
33	1.000	1
34	1.000	1
35	1.000	1
36	1.000	1
37	1.000	1
38	1.000	1
39	0.904	1
40	0.050	0

Appendix B - R Code:

Library

```
library(psych)
library(dplyr)
library(ggplot2)
library(DataExplorer)
library(PerformanceAnalytics)
library(corrplot)
library(knitr)
library(car)
library(reshape2)
library(usdm)        # for VIF tests
library(caret)
library(bestglm)
library(pscl)
library(MKmisc)
library(pROC)
```

Read data

```
crime_trainData <-
read.csv("https://raw.githubusercontent.com/621-
Group2/HW3/master/crime-training-data_modified.csv",
header = TRUE)

crime_trainData$age <- round(crime_trainData[,6], digits
= 0)
```

Basic Data Exploration and Statistic measures

```
Variable_names <- c("zn","indus", "chas", "nox", "rm",
"age", "dis", "rad", "tax",
"ptratio", "lstat", "medv", "target")

Definitions <- c("proportion of residential land zoned
for large lots (over 25000 square feet)", "proportion of
non-retail business acres per suburb", "a dummy var. for
whether the suburb borders the Charles River (1) or not
(0)", "nitrogen oxides concentration (parts per 10
million)", " average number of rooms per dwelling",
"proportion of owner-occupied units built prior to 1940",
```

```
"weighted mean of distances to five Boston employment
centers"," index of accessibility to radial highways",
"full-value property-tax rate per $10,000", " pupil-
teacher ratio by town", " lower status of the population
(percent)", "median value of owner-occupied homes in
$1000s", "whether the crime rate is above the median
crime rate (1) or not (0)")

Variable_type <-c("Predictor", "Predictor", "Predictor",
"Predictor", "Predictor", "Predictor", "Predictor",
"Predictor", "Predictor", "Predictor", "Predictor",
"Predictor", "Response")

Data_type <- c("quantitative", "quantitative",
"categorical", "quantitative", "quantitative",
"quantitative", "quantitative", "quantitative",
"quantitative", "quantitative", "quantitative",
"quantitative", "Categorical")

df_crime_md <- cbind.data.frame (Variable_names,
Definitions, Variable_type, Data_type)

colnames(df_crime_md) <- c("Variable Name", "Definition",
"Variable Type", "Data Type")

knitr::kable(df_crime_md)



Use Describe Package to calculate Descriptive Statistic
df_crime_des <- describe(crime_trainData, na.rm=TRUE,
interp=FALSE, skew=TRUE, ranges=FALSE, trim=.1, type=3,
check=TRUE, fast=FALSE, quant=c(.25,.75), IQR=TRUE)

knitr::kable(df_crime_des)

#Calculate mean missing values per variable
missing_data <- crime_trainData %>%
summarize_all(funs(sum(is.na(.)) / length(.)))


#Density
```

```
par(mfrow = c(3, 3))
d = melt(crime_trainData)
ggplot(d, aes(x= value)) +
    geom_density(fill='red') + facet_wrap(~variable,
scales = 'free')

#Boxplot
par(mfrow = c(3, 3))
boxdata = melt(crime_trainData)
ggplot(boxdata, mapping = aes(x= "", y = value)) +
    geom_boxplot(fill="red") + facet_wrap(~variable,
scales = `free')


#Examine outliers
get_outliers <-  function(x, n = 10) {

  v <- abs(x-mean(x,na.rm=TRUE)) > 3*sd(x,na.rm=TRUE)

  # capture all observations falling into outlier
definition sort descending
  obs <- sort(unique(x[v]), decreasing = T)

  # handle cases where the number of observations is less
than
  # the parameter n to return for the top and bottom n
values
  if (length(obs) < 2*n) {n <- floor(length(obs)/2)}

  hi <- obs[1:n]

  low <- obs[length(obs):(length(obs)-n +1)]

  # remove duplicate entries from the lower bound
outliers
  low <- setdiff(low, hi)

  return (list(Obs=obs, Hi=hi, Low=low))

}

o1 <- get_outliers(crime_trainData$zn)
```

```
if (length(o1[[1]])==0){
  o1 <- "none"
}


o2 <- get_outliers(crime_trainData$indus)

if (length(o2[[1]])==0){
  o2 <- "none"
}


o4 <- get_outliers(crime_trainData$nox)

if (length(o4[[1]])==0){
  o4 <- "none"
}


o5 <- get_outliers(crime_trainData$rm)

if (length(o5[[1]])==0){
  o5 <- "none"
}


o6 <- get_outliers(crime_trainData$age)

if (length(o6[[1]])==0){
  o6 <- "none"
}


o7 <- get_outliers(crime_trainData$dis)

if (length(o7[[1]])==0){
  o7 <- "none"
}


o8 <- get_outliers(crime_trainData$rad)
```

```
if (length(o8[[1]])==0){
  o8 <- "none"
}


o9 <- get_outliers(crime_trainData$tax)

if (length(o9[[1]])==0){
  o9 <- "none"
}


o10 <- get_outliers(crime_trainData$ptratio)

if (length(o10[[1]])==0){
  o10 <- "none"
}


o11 <- get_outliers(crime_trainData$lstat)

if (length(o11[[1]])==0){
  o11 <- "none"
}


o12 <- get_outliers(crime_trainData$medv)

if (length(o11[[1]])==0){
  o12 <- "none"
}
```

Variable-to-Variable Analysis

```
#need to add one to the color command because 0 sets the
color to white.
pairs(crime_trainData, col = crime_trainData$target+1)
```

Correlation between Variables

```
par(mfrow = c(2, 1))
chart.Correlation(crime_trainData[1:4])
chart.Correlation(crime_trainData[5:8])
chart.Correlation(crime_trainData[9:13])

crimeCorr <- cor(crime_trainData)
par(mfrow=c(2,1))
corrplot(crimeCorr, method = "circle")
corrplot(crimeCorr, method = “number")
```

Multicollinearity

```
vifcor(crime_trainData[, 1:12],th=0.4)


vif(crime_trainData[, 1:12])


plot_missing(crime_trainData)

#Count of zero values
kable(colSums(crime_trainData==0), col.names = "Count of
Zero Values”)
```

Transformations

```
dataT <- crime_trainData

x1 <- glm(target ~. -chas, family= binomial(), data =
dataT)
mmps(x1)


#transform predictors
dataT$radlog <- log(dataT$rad)
dataT$taxlog <- log(dataT$tax)

x2 <- glm(target ~ rad+radlog+tax+taxlog, family=
binomial(), data = dataT)

mmps(x2)
```

```
all_model_metrics <- data.frame()
all_roc_curves <- list()
all_predictions <- list()

calc_metrics <- function(model_name, model, test, train,
show=FALSE) {


  pred_model <- predict(model, test, type = 'response')
  y_pred_model <- as.factor(ifelse(pred_model > 0.5, 1,
0))

  # psedo R2 value (McFaden):
  McFadenR2_value <- pR2(model1)[[4]]

  # Hosmer L    Test:
  HosmerL_value <- HLgof.test(fit = fitted(model), obs =
train$target)
  HL_Chi_value <- unname(HosmerL_value$C[1]$statistic[1])
  HL_p_value <- unname(HosmerL_value$C[3]$p.value[1])
  # Handle very low p-value
  HL_p_value_limit <- 2.2*(10^(-16))
  HL_p_value_flag <- ' '
  if (HL_p_value <= HL_p_value_limit) {
    HL_p_value_flag <- '*'
    HL_p_value <- HL_p_value_limit
  }

  # Confusion Matrix
  cm <- confusionMatrix(test$target, y_pred_model,
positive = "1", mode="everything" )

  kappa_value <- cm$overall[[2]]
  youden_value <- cm$byClass[[1]] - (1 - cm$byClass[[2]])
  F1Score_value <- cm$byClass[[7]]
  FP_value <- (cm$table[2,1]/nrow(test))*100

  #AUC
  AUC_value <- auc(test$target, pred_model)

  cm_df <- data.frame(Model=model_name,
```

```
                          AIC=round(AIC(model), 3),
                          BIC=round(BIC(model), 3),
                          McFadenR2 = round(McFadenR2_value,
3),
                          HL_Chi = round(HL_Chi_value, 3),
                          HL_p = HL_p_value,
                          '*' = HL_p_value_flag,
                          Kappa = round(kappa_value, 3),
                          Youden = round(youden_value, 3),
                          F1Score = round(F1Score_value, 3),
                          FPPrct = round(FP_value, 2),
                          AUC = round(AUC_value[[1]], 3))


#cbind(t(cm$overall),t(cm$byClass)))

  # ROC Curves
  roc_model <- roc(target ~ pred_model, data = test)

  # Result
  result <- list(cm_df, roc_model, pred_model)
  if (show) {

      # calculate AIC/BIC
      print(paste("AIC= ", round(AIC(model), 3)))
      print(paste("BIC= ", round(BIC(model), 3)))
      print("")

      print(cm)
  }

  return (result)

}

#model_metrics <- calc_metrics('best',
res.bestglm$BestModel, dev_test_T, show=T)

set.seed(1255)

## TRAIN/TEST Dataset Creation ##
```

```
# convert the target response variable to a factor
crime_trainData$target <-
as.factor(crime_trainData$target)

# create the dev_train and dev_test datasets using the
non-transformed variables
idx <-
createDataPartition(y=crime_trainData$target,p=0.7,list=F
ALSE)
dev_train <-crime_trainData[idx,]
dev_test <-crime_trainData[-idx,]

# create the dev_train and dev_test datasets using the
log-transformed variables

# apply the log transformations
dataT <- crime_trainData
dataT$rad <- log(dataT$rad)
dataT$tax <- log(dataT$tax)

idx <-
createDataPartition(y=dataT$target,p=0.7,list=FALSE)
dev_train_T <-dataT[idx,]
dev_test_T <-dataT[-idx,]
```

Model 1 : Baseline using all Predictor Variables

```
model1 <- glm(target ~ ., family=binomial(),
data=dev_train)

summary(model1)


exp(coef(model1))


#all_model_metrics <- rbind(all_model_metrics,
calc_metrics("Model1", model1, dev_test, dev_train,
show=F))
m1<- calc_metrics("Model1", model1, dev_test, dev_train,
show=F)
all_model_metrics <- rbind(all_model_metrics, m1[[1]])
```

```
all_roc_curves[[1]] <- m1[[2]]

all_predictions[[1]] <- m1[[3]]
```

Model 2 : Baseline using Transformed Variables

```
model2 <- glm(target ~ .,  family=binomial(),
data=dev_train_T)

summary(model2)


exp(coef(model1))


#all_model_metrics <- rbind(all_model_metrics,
calc_metrics("Model2", model2, dev_test_T, dev_train_T,
show=F))
m2 <- calc_metrics("Model2", model2, dev_test_T,
dev_train_T, show=F)
all_model_metrics <- rbind(all_model_metrics, m2[[1]])

all_roc_curves[[2]] <- m2[[2]]

all_predictions[[2]] <- m2[[3]]
```

Model 1 - Model 2 Comparison

```
anova(model1, model2,  test="Chisq")
```

Model 3 : AIC Stepwise Variable Selection

```
mod3 <- glm(target ~ .,  family=binomial(),
data=dev_train_T)

# suppress printing the information during the each step
model3 <- step(mod3, direction="both", trace=0)

summary(model3)

anova(model3)
```

```
#all_model_metrics <- rbind(all_model_metrics,
calc_metrics("Model3", model3, dev_test_T, dev_train_T,
show=F))
m3 <- calc_metrics("Model3", model3, dev_test_T,
dev_train_T, show=F)
all_model_metrics <- rbind(all_model_metrics, m3[[1]])

all_roc_curves[[3]] <- m3[[2]]

all_predictions[[3]] <- m3[[3]]
```

Model 4 : Using VIF Reduction with Transformed Predictor Variables

```
model4 <- glm(target ~ . , family=binomial(),
data=dev_train_T)


vif_df <- data.frame(VIF=car::vif(model4))
x <- cbind(Variable = rownames(vif_df), vif_df)
rownames(x) <-NULL

kable(arrange(x, desc(VIF)))


model4 <- update(model4, . ~ . -rm -medv)


model4 <- update(model4, . ~ . -zn -chas -dis - ptratio -
lstat)

summary(model4)


#all_model_metrics <- rbind(all_model_metrics,
calc_metrics("Model4", model4, dev_test_T, dev_train_T,
show=F))
m4 <- calc_metrics("Model4", model4, dev_test_T,
dev_train_T, show=F)
all_model_metrics <- rbind(all_model_metrics, m4[[1]])
```

```
all_roc_curves[[4]] <- m4[[2]]

all_predictions[[4]] <- m3[[3]]
```

Model 5 : Using BestGlm using Transformed Predictors

```
# dataframe containing the design matrix of X and the
output Y
bestglm_df <- within(dev_train_T, {
  y    <- target           # outcome variable must be
named y
  target <- NULL           # drop target as a variable
after it's been move to y
})

## AIC
res.bestglm.aic <-
  bestglm(Xy = bestglm_df,
          family = binomial(),
          IC = "AIC",                 # AIC Information
criteria for
          method = "exhaustive")


## Show top 5 models
kable(data.frame(Model=rownames(res.bestglm.aic$BestModel
s),

Criterion=res.bestglm.aic$BestModels$Criterion))


model5.aic <- res.bestglm.aic$BestModel

summary(model5.aic)
```

Using Bayesian Information Criterion (BIC)

```
## BIC
res.bestglm.bic <-
  bestglm(Xy = bestglm_df,
          family = binomial(),
```

```
          IC = "BIC",                   # Use BIC
Information
          method = "exhaustive")


## Show top 5 models
kable(data.frame(Model=rownames(res.bestglm.bic$BestModel
s),

Criterion=res.bestglm.bic$BestModels$Criterion))


model5.bic <- glm(target ~ nox + age + rad + tax,
family=binomial(), data=dev_train_T)

summary(model5.bic)


#all_model_metrics <- rbind(all_model_metrics,
calc_metrics("Model5.AIC", model5.aic, dev_test_T,
dev_train_T, show=F))
#all_model_metrics <- rbind(all_model_metrics,
calc_metrics("Model5.BIC", model5.bic, dev_test_T,
dev_train_T, show=F))

m5.AIC <- calc_metrics("Model5.AIC", model5.aic,
dev_test_T, dev_train_T, show=F)
all_model_metrics <- rbind(all_model_metrics,
m5.AIC[[1]])

all_roc_curves[[5]] <- m5.AIC[[2]]

all_predictions[[5]] <- m5.AIC[[3]]

m5.BIC <- calc_metrics("Model5.BIC", model5.bic,
dev_test_T, dev_train_T, show=F)
all_model_metrics <- rbind(all_model_metrics,
m5.BIC[[1]])

all_roc_curves[[6]] <- m5.BIC[[2]]

all_predictions[[6]] <- m5.BIC[[3]]
```

Model Selection and Evaluation

```
kable(all_model_metrics)


par(mfrow=c(2,3))

plot.roc(dev_test$target,
as.numeric(all_predictions[[1]]),
         #print.thres=TRUE,
         grid=T,
         percent=F,  print.auc=TRUE, max.auc.polygon=T,
         #auc.polygon=TRUE,
         main="Model 1 ROC Curve")

plot.roc(dev_test_T$target,
as.numeric(all_predictions[[2]]),
         print.thres=TRUE,
         percent=F,  print.auc=TRUE,
max.auc.polygon=TRUE,
         #auc.polygon=TRUE,
         main="Model 2 ROC Curve")

plot.roc(dev_test_T$target,
as.numeric(all_predictions[[3]]),
         print.thres=TRUE,
         percent=F,  print.auc=TRUE,
max.auc.polygon=TRUE,
         #auc.polygon=TRUE,
         main="Model 3 ROC Curve")

plot.roc(dev_test_T$target,
as.numeric(all_predictions[[4]]),
         print.thres=TRUE,
         percent=F,  print.auc=TRUE,
max.auc.polygon=TRUE,
         #auc.polygon=TRUE,
         main="Model 4 ROC Curve")

plot.roc(dev_test_T$target,
as.numeric(all_predictions[[5]]),
```

```
          print.thres=TRUE,
          percent=F,  print.auc=TRUE,
max.auc.polygon=TRUE,
          #auc.polygon=TRUE,
          main="Model 5.AIC ROC Curve")


plot.roc(dev_test_T$target,
as.numeric(all_predictions[[6]]),
          print.thres=TRUE,
          percent=F,  print.auc=TRUE,
max.auc.polygon=TRUE,
          #auc.polygon=TRUE,
          main="Model 5.BIC ROC Curve”)


summary(model3)
summary(model5.aic)
```

Evaluation

```
# Loading and transforming Evaluation Data Set
crime_EvalData <-
read.csv("https://raw.githubusercontent.com/621-
Group2/HW3/master/crime-evaluation-data_modified.csv",
header = TRUE)

crime_EvalData$age <- round(crime_EvalData[,6], digits =
0)

summary(crime_EvalData)

#copy Evaluation Data Set prior to transformation
crime_EvalDataT <- crime_EvalData

# Apply Log Transform
crime_EvalDataT$radlog <- log(crime_EvalDataT$rad)
crime_EvalDataT$taxlog <- log(crime_EvalDataT$tax)


pred_model_final <- predict(model5.aic, crime_EvalDataT,
type = 'response')
```

```
y_pred_model_final <- as.factor(ifelse(pred_model_final > 0.5, 1, 0))


write.csv(as.data.frame(y_pred_model_final), file = "group2_project3_results.csv", row.names=F




#plot.roc(all_roc_curves[[1]], auc=T)
#plot.roc(all_roc_curves[[5]])
#plot.roc(all_roc_curves[[6]])
```