Multiple linear regression and binary logistic regression models

Abdellah AitElmouden | Gabriel Abreu | Jered Ataky | Patrick Maloney

4/13/2021

Contents

INTRODUCTION

The aim of this assignment is to build a binary logistic regression model to predict whether a neighborhood will be at risk for high crime levels, using a data set containing information on crime for various neighborhoods of a major city. Each record has a response variable indicating whether or not the crime rate is above the median crime rate (1) or not (0). before building the model we will perform some Exploratory Data Analysis (EDA): To visualize distributions and draw correlations between attributes. There are few issues in the data. Although there are no missing values but most of the variables seem to be skewed and not normally distributed, hence we use log transformation to make them symmetric. The obtained model were tested on criminal evaluation data. We compared the results of model predictions and selected the best binary logistic regression model.

DATA EXPLORATION

In this section, we are going to explore the data to see the data type and data structure, We will also check the correlation among the variables and most importantly to see if there are missing values in the data.

Both training and evaluation datasets have been read using read.csv function and above table is a sample of training dataset. we can see that the data is composed of 466 observations and 12 predictor variables. The response variable target is binary (0 or 1). All observations in this dataset are complete.

```
# load the data

raw_train_data <- read.csv("https://raw.githubusercontent.com/aaitelmouden/DATA621/master/Homework3/data

raw_test_data <- read.csv("https://raw.githubusercontent.com/aaitelmouden/DATA621/master/Homework3/data/
```

```r
cp_data <- raw_train_data


glimpse(raw_train_data)
```

```
## Rows: 466
## Columns: 13
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 100, 20...
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5.19, ...
## $ chas    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ nox     <dbl> 0.605, 0.871, 0.740, 0.428, 0.488, 0.520, 0.693, 0.693, 0.5...
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519, 6.3...
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38.1, 19...
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896, 1.6...
## $ rad     <int> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5, 5, 2...
## $ tax     <int> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330, 398,...
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, 16.4,...
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5.68, 9...
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20.9, 2...
## $ target  <int> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,...
```

Now let's explore the data structure using skim function from skimr package. This is an efficient function which not only produces the statistics summary but also builds histogram for each numberic variable, show number of missing values and quantiles. This gives a bird eye view of the training dataset.

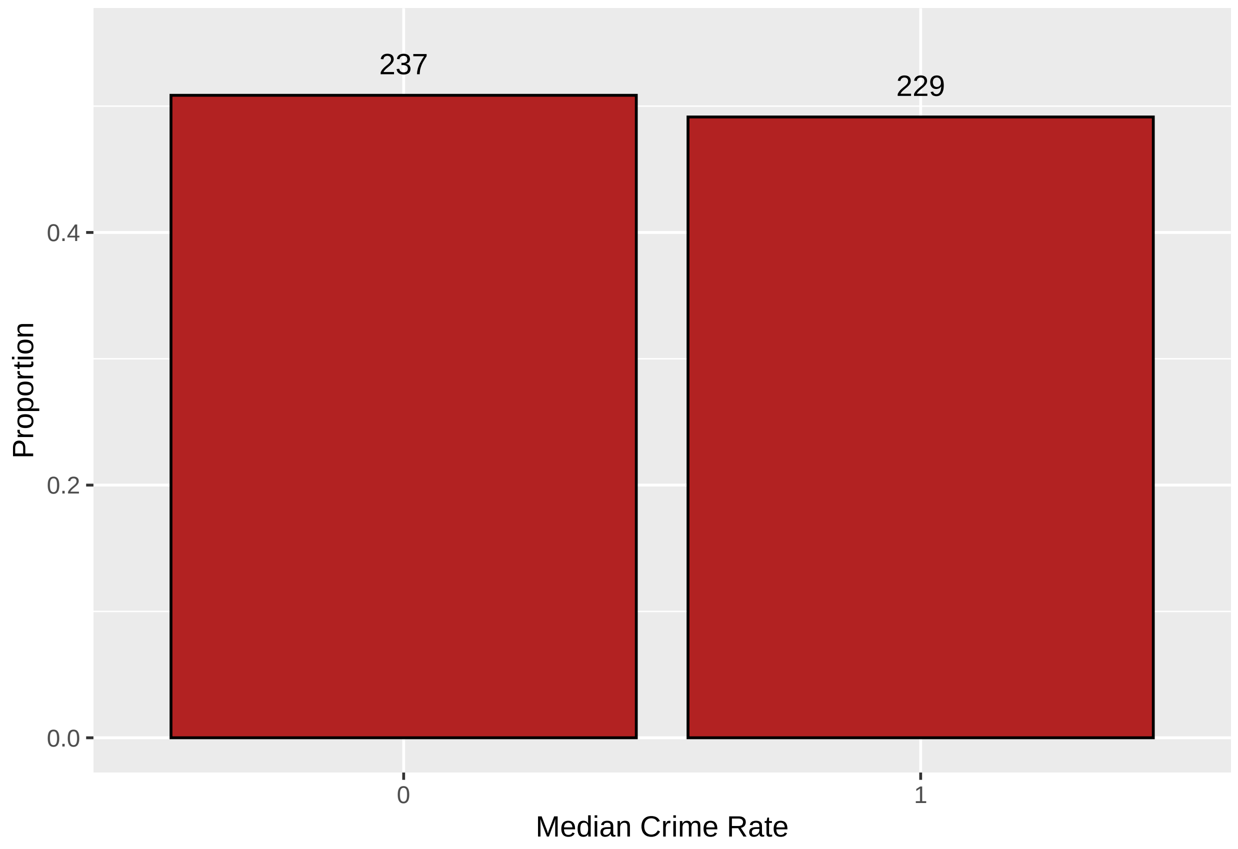
```r
# getting useful summary statistics
#skim(raw_train_data)

table1 <- tbl_summary(raw_train_data,
          statistic = list(all_continuous() ~ "{mean} ({sd}) {median} {min} {max}"), missing = "no")
table1
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 466**
zn	12 (23) 0 0 100
indus	11.1 (6.8) 9.7 0.5 27.7
chas	33 (7.1%)
nox	0.55 (0.12) 0.54 0.39 0.87
rm	6.29 (0.70) 6.21 3.86 8.78
age	68 (28) 77 3 100
dis	3.80 (2.11) 3.19 1.13 12.13
rad	
1	17 (3.6%)
2	20 (4.3%)
3	36 (7.7%)
4	103 (22%)
5	109 (23%)
6	25 (5.4%)
7	15 (3.2%)
8	20 (4.3%)
24	121 (26%)
tax	410 (168) 334 187 711
ptratio	18.40 (2.20) 18.90 12.60 22.00
lstat	13 (7) 11 2 38
medv	23 (9) 21 5 50
target	229 (49%)

Proportion of target among the data. 0 means the crime rate below the median crime rate while 1 is above the median crime rate.

```
raw_train_data$target <- factor(raw_train_data$target)

raw_train_data %>%
  ggplot(aes(x=target, y = ..prop.., group = 1)) +
  geom_bar(fill = 'firebrick', color = 'black')  +
  geom_text(stat='count', aes(label=..count..), vjust=-1) +
  xlab("Median Crime Rate") +
  ylab("Proportion") +
  ylim(0, 0.55)
```

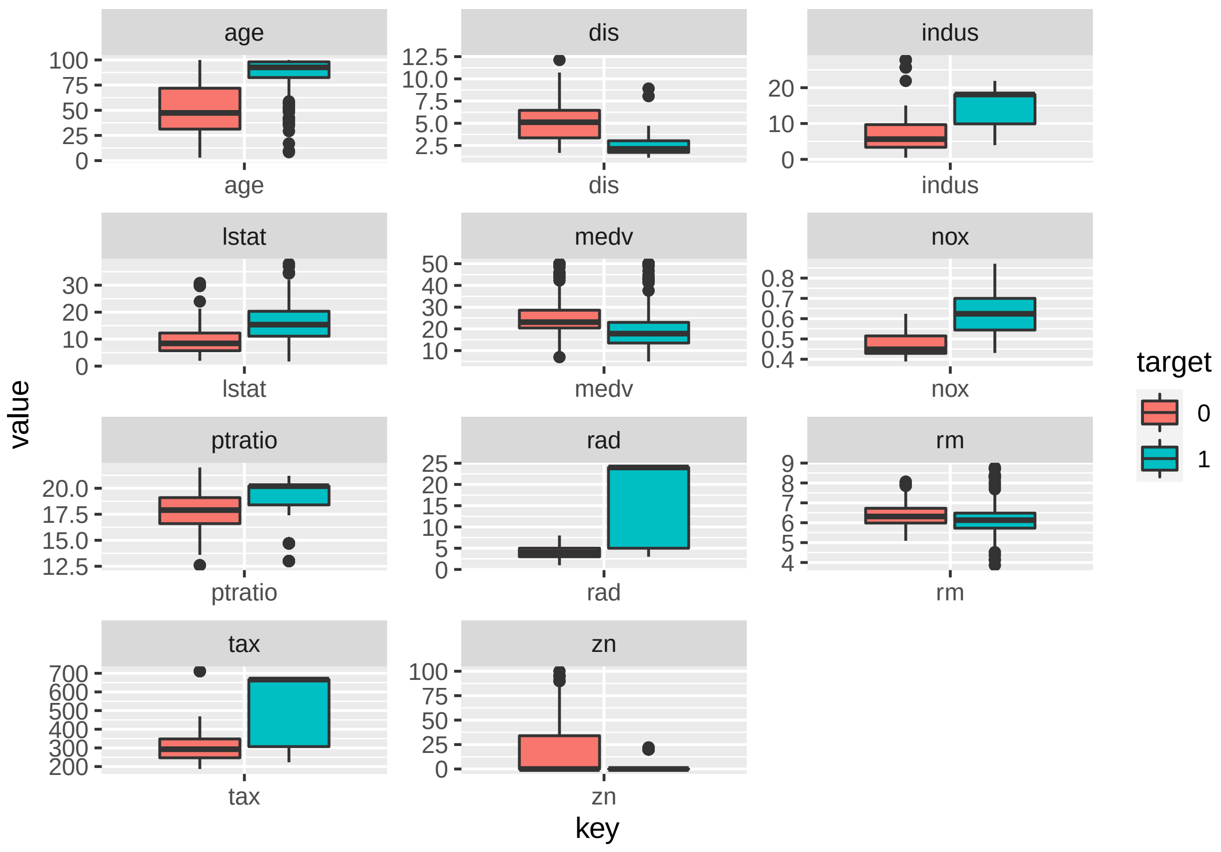


Looking at the histograms and box plots of each we can observe that:

- **age**: The box plot shows that generally older homes in neighborhoods are see as associated with higher crime.
- **chas**: most homes in the dataset border the Charles River, thus this may not be a good predictor variable
- **dis**: a right skewed distribution where a lower distance to employment centers shows a higher crime indicator
- **indus**: bi-modal distribution of industrial sectors and generally seen by the box plots that the higher industrial activity results in an increased crime factor
- **lstat**: a predictor variable based on “status” of population. However it is ambigious what the sale in this factor reflects, but the observation is that the higher on the lstat scale the more indicator of crime
- **mdev**: median value of homes, and seems correct that we would see higher value homes associated with lower crimes

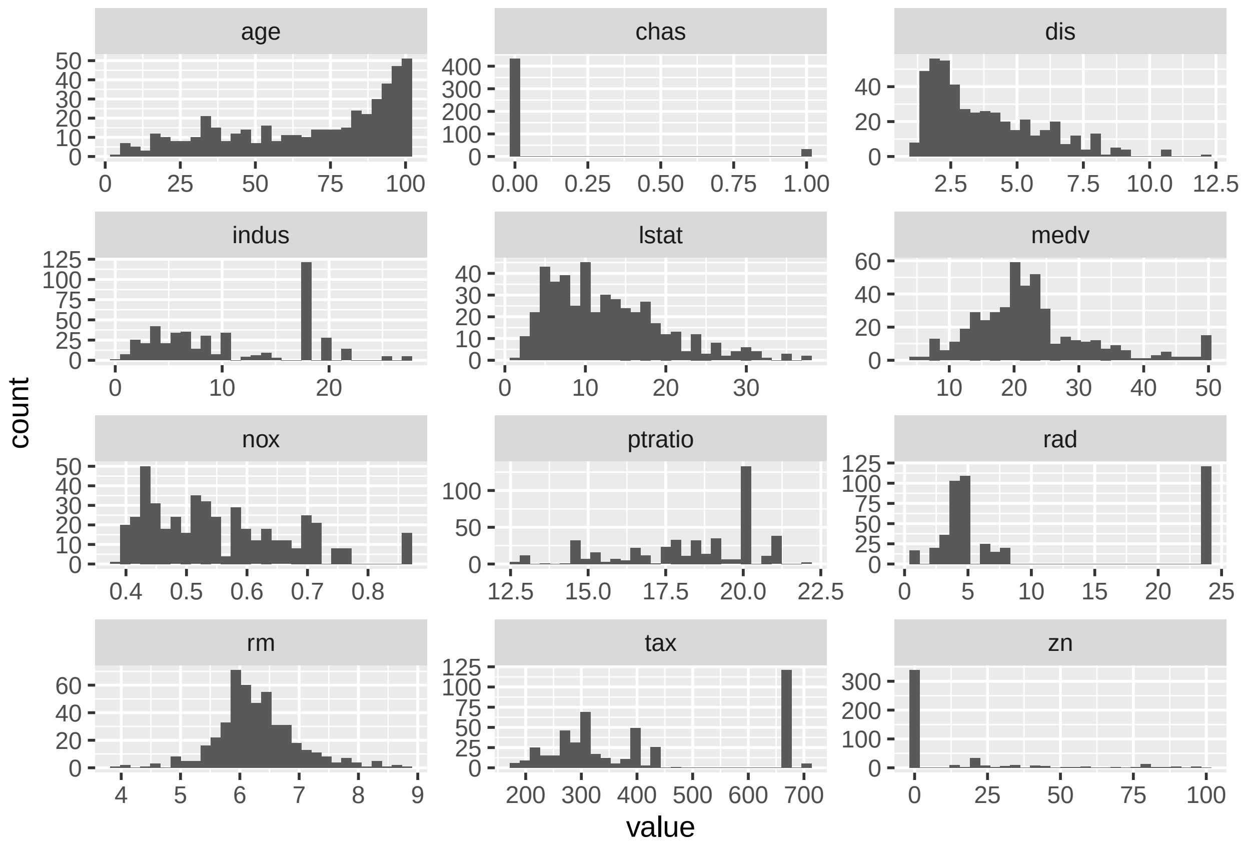
- **nox**: the amount of nitrogen oxides concentrations is right skewed with most locations not having a "high" amount, and as the concentration increases as does the crime
- **ptratio**: student to teacher ratio, as convention and observation show a high student to teacher ratio is indicative of higher crimes
- **rad**: the distance to highways seems slightly bi-modal, and higher distance from highways seems to be associated with higher crime, however the variability on the positive crime indicator is very large
- **rm**: the average number of rooms per home looks normally distributed and the association with crime seems evenly distributed as per the box plot
- **tax**: the property tax variable is bi-modal, the box plot shows that the variability of a positive crime indicator is fairly large
- **zn**: large lot zones show most values as 0 and lower proportions seems associated with higher crime

```
raw_train_data %>%
  dplyr::select(-chas) %>%
  gather(key, value, -target) %>%
  mutate(key = factor(key)) %>%
  ggplot(aes(x = key, y = value)) +
  geom_boxplot(aes(fill = target)) +
  facet_wrap(~ key, scales = 'free', ncol = 3)
```

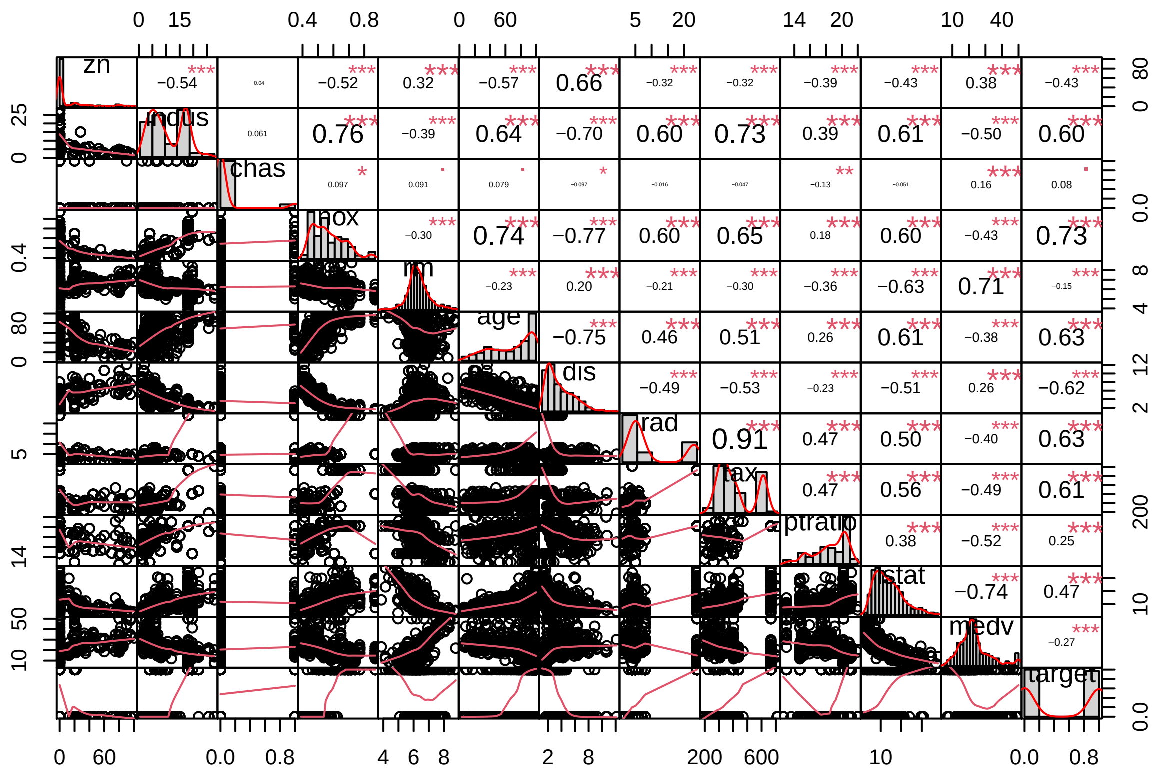


```
raw_train_data %>%
  gather(key, value, -c(target)) %>%
  ggplot(aes(value)) +
  geom_histogram() +
  facet_wrap(~ key, scales = 'free', ncol = 3)
```

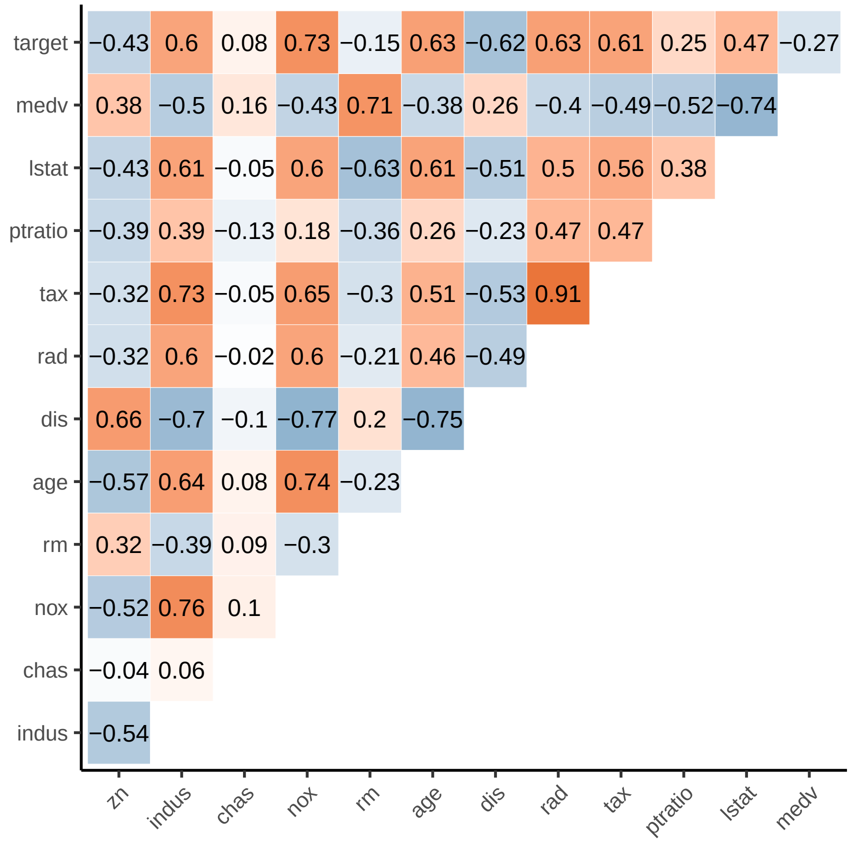
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Correlations among predictors and Variable Selection



Correlation using ggcorrplot



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

According to correlation plot the target variable is positively correlated with nox(.73), age(.63), rad(.63), and tax(.61). Also we can see the target variable is negatively correlated with dis(-.62) as seen in the histogram and box plots, the chas variable as a very weak correlation with all the other variables, and including the target. Therefore we can look to eliminate it from the analysis. We also noticed that there is present a amount of correlation amongst the predictor variables and this is suspect for multicollinearity issues

DATA PREPARATION

```r
set.seed(123)

training.samples <- train_data$target %>% createDataPartition(p= 0.8, list = FALSE)

train.data <- train_data[training.samples, ]
test.data <- train_data[-training.samples, ]
```

BUILD MODELS

Model 1 : Base Model Subsets

The first model is the simplest. Run the logistic regression formula with all the variables as predictors and manually eliminate variables based on significance.

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

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = raw_train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8464  -0.1445  -0.0017   0.0029   3.4665
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas          0.910765   0.755546   1.205  0.22803
## nox          49.122297   7.931706   6.193 5.90e-10 ***
## rm           -0.587488   0.722847  -0.813  0.41637
## age           0.034189   0.013814   2.475  0.01333 *
## dis           0.738660   0.230275   3.208  0.00134 **
## rad           0.666366   0.163152   4.084 4.42e-05 ***
## tax          -0.006171   0.002955  -2.089  0.03674 *
## ptratio       0.402566   0.126627   3.179  0.00148 **
## lstat         0.045869   0.054049   0.849  0.39608
## medv          0.180824   0.068294   2.648  0.00810 **
## ---
## 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: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

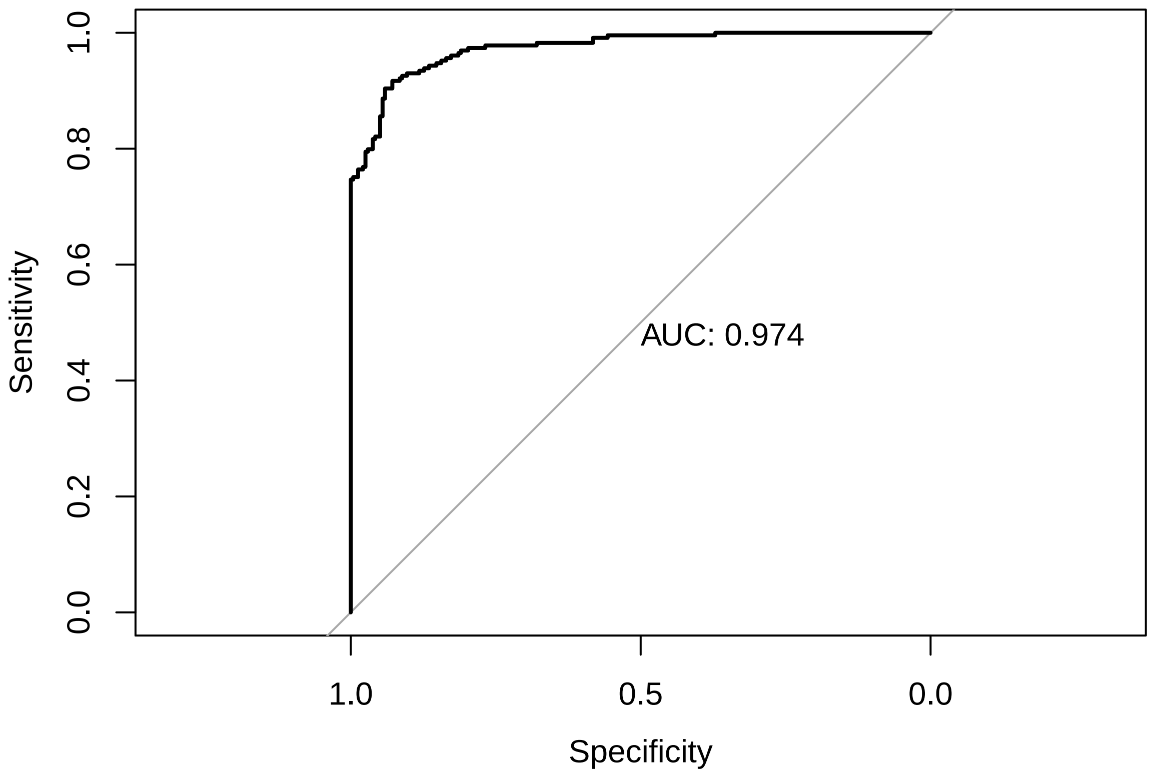
```
glm.probs <- predict(model1, type="response")

# Confirm the 0.5 threshold
glm.pred <- ifelse(glm.probs > 0.5, 1, 0)
results <- tibble(target=raw_train_data$target, pred=glm.pred)
results <- results %>%
mutate(pred.class = as.factor(pred), target.class = as.factor(target))
```

As we can see, summary() returns the estimate, standard errors, z-score, and p-values on each of the coefficients. Look like some of the coefficients are non-significant here ($p > 0.05$). chas, lstat, rm, indus, ptratio, tax and dis. And it seems like only age, nox, rad and medv have significant impact on target.

The summary shows also the null deviance (the deviance just for the mean) and the residual deviance (the deviance for the model with all the predictors). There's a large difference between the 2, along with 12 degrees of freedom.

Prediction

We will plot the ROC curve for the predictive result. ROC in logistic regression are used for determining the best cutoff value for predicting whether a new observation is a “failure” (0) or a “success” (1). the ROC curve shows graphically the tradeoff that occurs between trying to maximize the true positive rate vs. trying to minimize the false positive rate. In an ideal situation, you would have sensitivity and specificity near 100% at all cutoffs, meaning you predict perfectly in all cases. which is not the case for our data.



Using the summ() function we can display model1 regression summary. the first model shows an R square of 0.83. The AIC and the BIC are 218.05, 271.92 alternatively.

```
summ(model1)
```

Observations	466
Dependent variable	target
Type	Generalized linear model
Family	binomial
Link	logit

$\chi^2(12)$	453.83
Pseudo-R^2 (Cragg-Uhler)	0.83
Pseudo-R^2 (McFadden)	0.70
AIC	218.05
BIC	271.92

	Est.	S.E.	z val.	p
(Intercept)	-40.82	6.63	-6.15	0.00
zn	-0.07	0.03	-1.90	0.06
indus	-0.06	0.05	-1.36	0.17
chas	0.91	0.76	1.21	0.23
nox	49.12	7.93	6.19	0.00
rm	-0.59	0.72	-0.81	0.42
age	0.03	0.01	2.47	0.01
dis	0.74	0.23	3.21	0.00
rad	0.67	0.16	4.08	0.00
tax	-0.01	0.00	-2.09	0.04
ptratio	0.40	0.13	3.18	0.00
lstat	0.05	0.05	0.85	0.40
medv	0.18	0.07	2.65	0.01

Standard errors: MLE

Model 2 : Using Stepwise Regression

Although we have used backward elimination in which we eliminated insignificant variables one by one from the model as discussed before. We can use step() function which is more robust and it is used for stepwise regression. Basically, it eliminates all the insignificant variables one-by-one under the hood and brings the significant variables. This model is used only to verify the result of Model1 using step-wise regression

```r
model2 <- step(model1)
```

```
## Start:  AIC=218.05
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##
##           Df Deviance    AIC
## - rm       1   192.71 216.71
## - lstat    1   192.77 216.77
## - chas     1   193.53 217.53
## - indus    1   193.99 217.99
## <none>         192.05 218.05
## - tax      1   196.59 220.59
## - zn       1   196.89 220.89
## - age      1   198.73 222.73
## - medv     1   199.95 223.95
## - ptratio  1   203.32 227.32
## - dis      1   203.84 227.84
## - rad      1   233.74 257.74
## - nox      1   265.05 289.05
##
## Step:  AIC=216.71
## target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - chas     1   194.24 216.24
## - lstat    1   194.32 216.32
## - indus    1   194.58 216.58
## <none>         192.71 216.71
## - tax      1   197.59 219.59
## - zn       1   198.07 220.07
## - age      1   199.11 221.11
## - ptratio  1   203.53 225.53
## - dis      1   203.85 225.85
## - medv     1   205.35 227.35
## - rad      1   233.81 255.81
## - nox      1   265.14 287.14
##
## Step:  AIC=216.24
## target ~ zn + indus + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - indus    1   195.51 215.51
## <none>         194.24 216.24
## - lstat    1   196.33 216.33
## - zn       1   200.59 220.59
## - tax      1   200.75 220.75
## - age      1   201.00 221.00
## - ptratio  1   203.94 223.94
## - dis      1   204.83 224.83
## - medv     1   207.12 227.12
## - rad      1   241.41 261.41
## - nox      1   265.19 285.19
##
## Step:  AIC=215.51
## target ~ zn + nox + age + dis + rad + tax + ptratio + lstat +
##     medv
##
##           Df Deviance    AIC
## - lstat    1   197.32 215.32
## <none>         195.51 215.51
## - zn       1   202.05 220.05
## - age      1   202.23 220.23
## - ptratio  1   205.01 223.01
## - dis      1   205.96 223.96
## - tax      1   206.60 224.60
## - medv     1   208.13 226.13
## - rad      1   249.55 267.55
## - nox      1   270.59 288.59
##
## Step:  AIC=215.32
## target ~ zn + nox + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## <none>         197.32 215.32
## - zn       1   203.45 219.45
## - ptratio  1   206.27 222.27
## - age      1   207.13 223.13
## - tax      1   207.62 223.62
## - dis      1   207.64 223.64
## - medv     1   208.65 224.65
## - rad      1   250.98 266.98
## - nox      1   273.18 289.18
```

```
summ(model2)
```

Observations	466
Dependent variable	target
Type	Generalized linear model
Family	binomial
Link	logit

$\chi^2(8)$	448.55
Pseudo-R^2 (Cragg-Uhler)	0.82
Pseudo-R^2 (McFadden)	0.69
AIC	215.32
BIC	252.62

	Est.	S.E.	z val.	p
(Intercept)	-37.42	6.04	-6.20	0.00
zn	-0.07	0.03	-2.14	0.03
nox	42.81	6.68	6.41	0.00
age	0.03	0.01	3.01	0.00
dis	0.65	0.21	3.06	0.00
rad	0.73	0.15	4.84	0.00
tax	-0.01	0.00	-2.92	0.00
ptratio	0.32	0.11	2.91	0.00
medv	0.11	0.04	3.12	0.00

Standard errors: MLE

From the step() output the last Step table is the model proposed, the output also shows the "Call" function **target ~ zn + nox + age + dis + rad + tax + ptratio + medv**, which describes the actual model and what input variables it includes, and the "Coefficients" are the actual parameter estimates for these values.

From the summ() output above, it can be seen that:

- The two models have almost the same Pseudo, meaning that they are equivalent in explaining the outcome. However, the model 2 is more simple than model 1 because it incorporates less variables. All things equal, the simple model is always better in statistics.
- The AIC and the BIC of the model 2 are lower than those of the model1. In model comparison strategies, the model with the lowest AIC and BIC score is preferred.
- Finally, the F-statistic p.value of the model 2 is lower than the one of the model 1. This means that the model 2 is statistically more significant compared to model 1, which is consistent to the above conclusion.

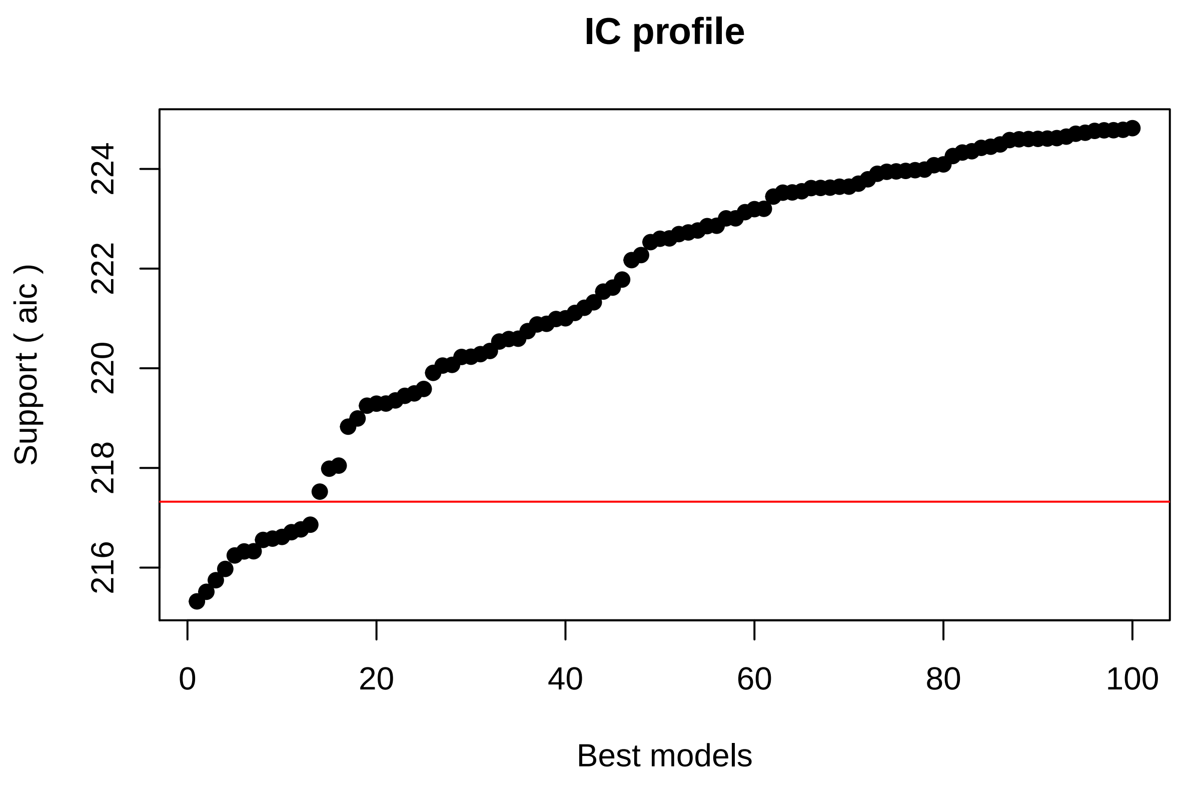
Model 3 Using glmulti:

glmulti finds what are the n best models (the confidence set of models) among all possible models. Models are fitted with the specified fitting function (default is glm) and are ranked with the specified Information Criterion (default is aicc). The output can be used for model selection, variable selection, and multimodel inference. It takes few time to optimize the model though. We will see which model performed best in terms of performance and accuracy in the next section.

```
print(model3)
```

```
## glmulti.analysis
## Method: h / Fitting: glm / IC used: aic
## Level: 1 / Marginality: FALSE
## From 100 models:
## Best IC: 215.322852780351
## Best model:
## [1] "target ~ 1 + zn + nox + age + dis + rad + tax + ptratio + medv"
## Evidence weight: 0.0787039901208588
## Worst IC: 224.81785157674
## 13 models within 2 IC units.
## 53 models to reach 95% of evidence weight.
```

```
plot(model3)
```



The horizontal red line differentiates between models whose AICc value is less versus more than 2 units away from that of the “best” model (i.e., the model with the lowest AICc). The output above shows that there are 14 such models. Sometimes this is taken as a cutoff, so that models with values more than 2 units away are considered substantially less plausible than those with AICc values closer to that of the best model.

```r
top <- weightable(model3)
top <- top[top$aic <= min(top$aic) + 2,]
top
```

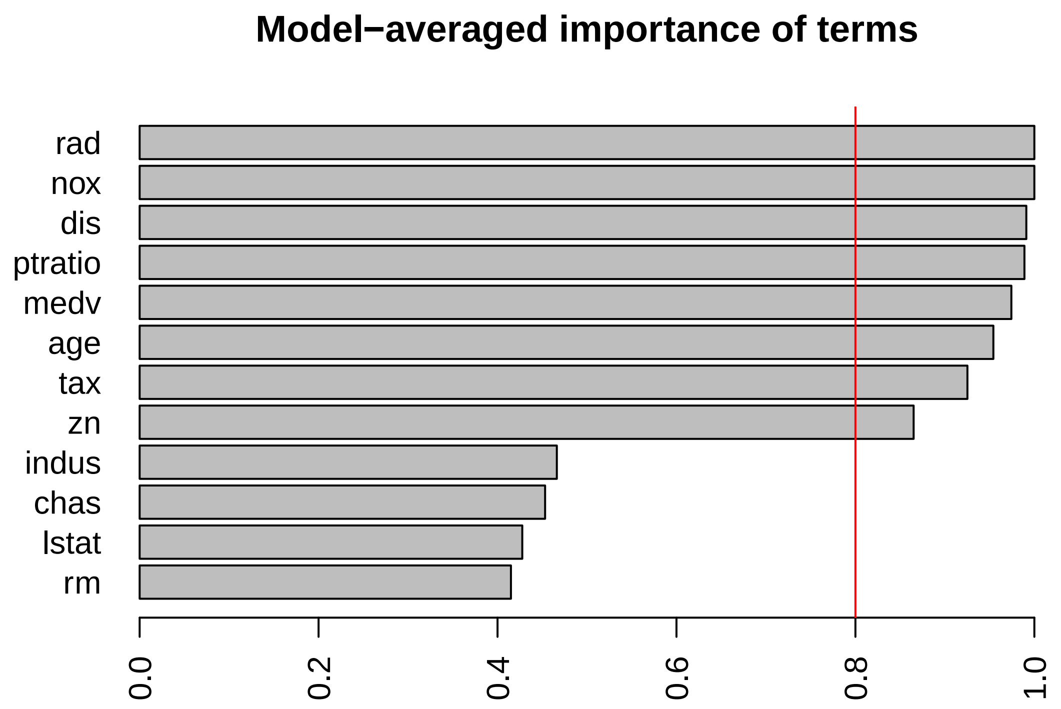
```
##                                                                              model
## 1                            target ~ 1 + zn + nox + age + dis + rad + tax + ptratio + medv
## 2                    target ~ 1 + zn + nox + age + dis + rad + tax + ptratio + lstat + medv
## 3                       target ~ 1 + zn + nox + rm + age + dis + rad + tax + ptratio + medv
## 4                     target ~ 1 + zn + chas + nox + age + dis + rad + tax + ptratio + medv
## 5            target ~ 1 + zn + indus + nox + age + dis + rad + tax + ptratio + lstat + medv
## 6             target ~ 1 + zn + indus + chas + nox + age + dis + rad + tax + ptratio + medv
## 7                    target ~ 1 + zn + indus + nox + age + dis + rad + tax + ptratio + medv
## 8               target ~ 1 + zn + indus + nox + rm + age + dis + rad + tax + ptratio + medv
## 9             target ~ 1 + zn + chas + nox + age + dis + rad + tax + ptratio + lstat + medv
## 10               target ~ 1 + zn + chas + nox + rm + age + dis + rad + tax + ptratio + medv
## 11 target ~ 1 + zn + indus + chas + nox + age + dis + rad + tax + ptratio + lstat + medv
## 12    target ~ 1 + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + medv
## 13              target ~ 1 + zn + nox + rm + age + dis + rad + tax + ptratio + lstat + medv
##         aic    weights
## 1  215.3229 0.07870399
## 2  215.5147 0.07150550
## 3  215.7475 0.06364809
## 4  215.9736 0.05684357
## 5  216.2441 0.04965305
## 6  216.3248 0.04769002
## 7  216.3281 0.04761085
## 8  216.5558 0.04248815
## 9  216.5819 0.04193779
## 10 216.6158 0.04123305
## 11 216.7108 0.03931963
## 12 216.7672 0.03822652
## 13 216.8614 0.03646691
```

We see that the “best” model is the one that only exculde lstat as a moderator. The second best includes lstat. And so on with other variables. The values under weights are the model weights (also called “Akaike weights”). From an information-theoretic perspective, the Akaike weight for a particular model can be regarded as the probability that the model is the best model.

Variable Importance

it may be better to ask about the relative importance of the various predictors more generally, taking all of the models into consideration. We can obtain a plot of the relative importance of the various model terms with:

```r
plot(model3, type="s")
```



The importance value for a particular predictor is equal to the sum of the weights/probabilities for the models in which the variable appears. So, a variable that shows up in lots of models with large weights will receive a high importance value. The vertical red line is drawn at 0.8, which is sometimes used as a cutoff to differentiate between important and not so important variables

SELECT MODELS

In this section, we are going to select the best model out of all through using **compare_performance** and **model_performance** functions from performance package. The functions calculates AIC, BIC, R2 & adjusted r-sq, RMSE, BF and Performance_Score. If we take a look at first two models, model2 is doing great as we saw before the values of AIC and BIC both are lower. RMSE are almost the same. Model3 was calculated through glmulti() package which optimizes the model and gets the best. The value of AIC and BIC, and RMSE are almost the same as model2. We can say that Model2 is the best performing model in terms of AIC, BIC and R2 and hence we will select Model2.

```
compare_performance(model1, model2, rank = TRUE) %>% kable() %>% kable_styling()
```

Name	Model	AIC	BIC	R2_Tjur	RMSE	Sigma	Log_loss	Score_log	Score_spherical
model1	glm	218.0469	271.9213	0.7397840	0.2517121	0.6511103	0.2060589	-Inf	0.0223437
model2	glm	215.3229	252.6205	0.7328891	0.2546085	0.6570987	0.2117198	-Inf	0.0214041

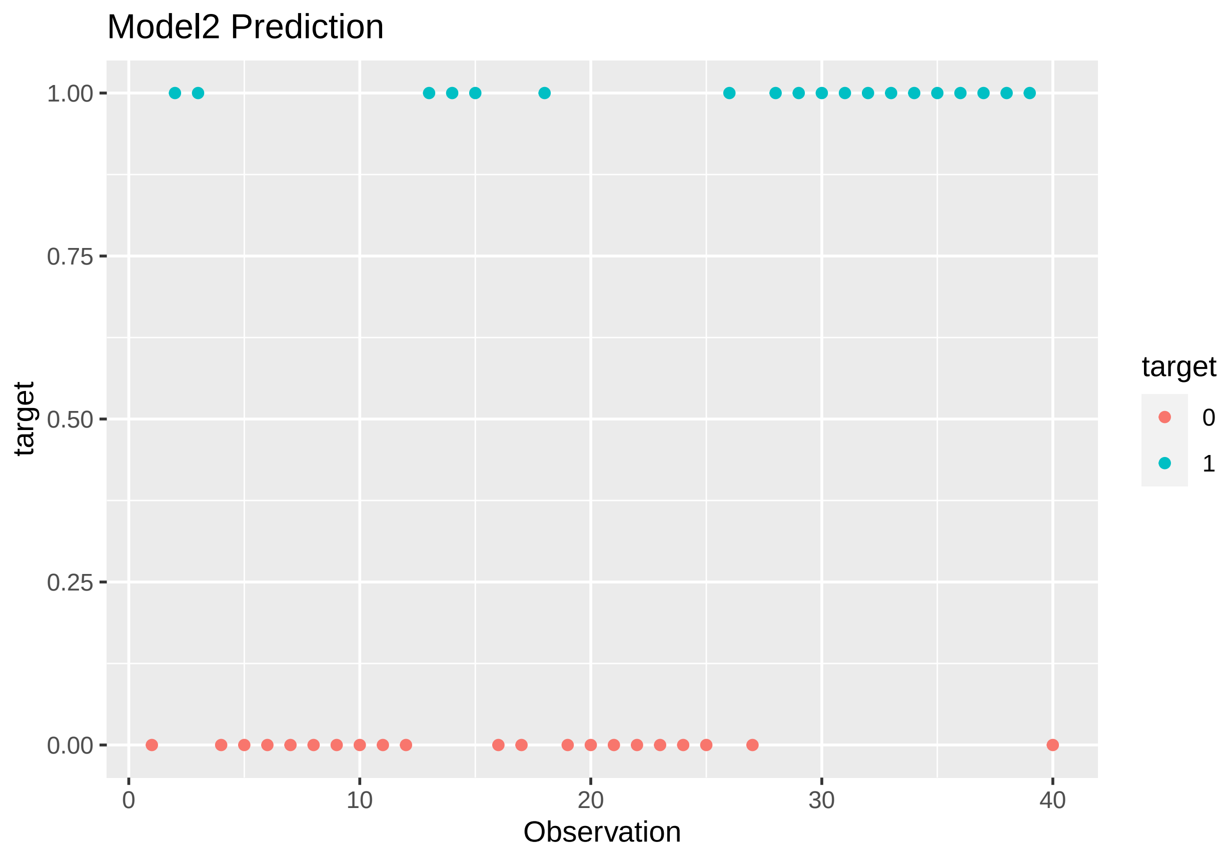
```
model_performance(model3@objects[[1]]) %>% kable() %>% kable_styling()
```

AIC	BIC	R2_Tjur	RMSE	Sigma	Log_loss	Score_log	Score_spherical	PCP
215.3229	252.6205	0.7328891	0.2546085	0.6570987	0.2117198	-Inf	0.0214041	0.8664839

MODEL PERFORMANCE

Based on model2, below is the prediction distribution of the test dataset, we see that the distribution is fairly split between the binary variable target

```r
test_predict <- predict(model2, newdata=raw_test_data)
test_predict <- ifelse(test_predict<.5,0,1)
raw_test_data$target <- test_predict
ggplot(raw_test_data, aes(x=index(raw_test_data), y=target, color=factor(target))) + geom_point() +
  labs(x="Observation", y="target", title = "Model2 Prediction", colour = "target")
```



```r
table(test_predict)
```

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

```r
write.csv(test_predict, "CrimePredictions.csv")
```

REFERENCES

- Model Selection using the glmulti and MuMIn Packages
- Regression Model Validation
- Binary Logistic Regression
- [What are pseudo R-squareds?] (https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/)