Crime Analysis And Prediction Via Logistic Regression

Brian Weinfeld

City University – School of Professional Studies

Abstract

I have been tasked with developing a logistic regression that will accurately determine whether a neighborhood in a major city has a crime rate either above or below the median rate. The training set I have been provided with has 466 observations across 13 predictors. The evaluation set has 40 observations that must be categorizied.

I will develop three regression models, explore each, and ultimately select the strongest model to use on the evaluation set.

I will begin by exploring the data set as a whole and then each individual predictor.

Data Exploration

Initial inspection of the data very helpfully shows that there are no missing values. Thus, we will not need to impute any data or remove/modify any of the predictors before individual analysis.

```
raw.data %>%
  map_df(~sum(is.na(.))/nrow(raw.data)) %>%
  kable()
```

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	target
0	0	0	0	0	0	0	0	0	0	0	0	0	0

I partition the data into a training set and testing set. The training set contains 80% of the total number of observations. Using caret's createDataPartition method ensures that each partition has target values roughly representation of the overall population. With this data set, there is an appoximate even split between below the median crime rate (0) and above (1). This is reflected in both the training and the testing set.

```
set.seed(123)
part <- caret::createDataPartition(raw.data$target, p=0.8, list=FALSE)
training <- raw.data %>%
  filter(row_number() %in% part)
testing <- raw.data %>%
  filter(!row_number() %in% part)
```

I created a function called Predictor.Disp that is designed to help me in displaying each predictor. It produces a density plot of the predictor in order to aid in determining the predictor's distribution and a box plot faceted by the target response variable in order to aid in comparison of the spread between the two groups.

```
Predictor.Disp <- function(x){
  require(gridExtra)
  plot.1 <- ggplot(training, aes_string(x)) +
    geom_density() +
    labs(y = 'Density',
         title = 'Predictor Distribution') +
    theme_bw() +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0, 0.05, 0.05))

  plot.2 <-
    training %>%
      mutate(target = factor(target)) %>%
```

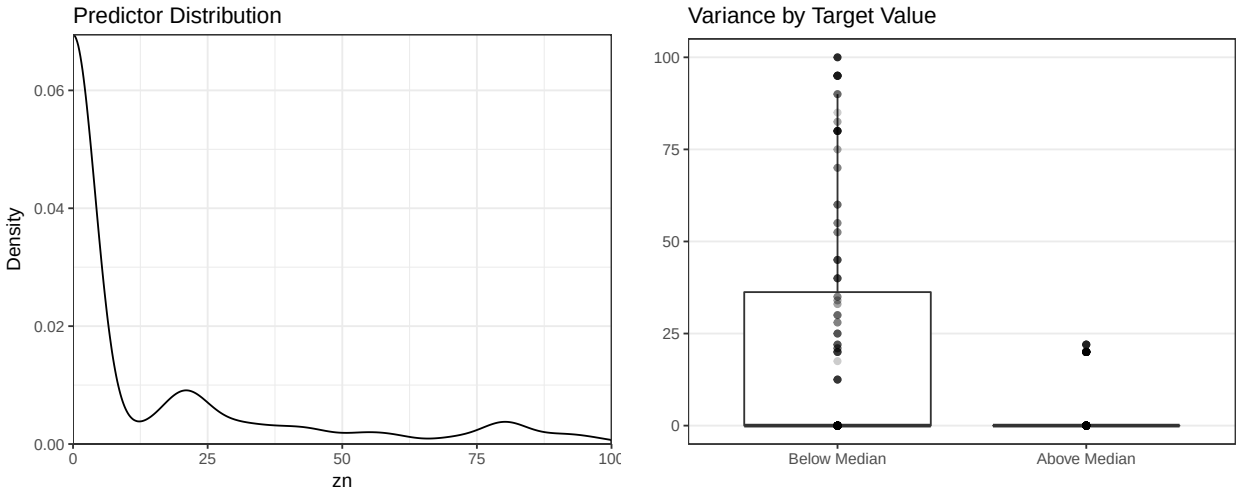
```
      ggplot(aes_string('target', x)) +
      geom_boxplot() +
      geom_point(alpha=0.2) +
      theme_bw() +
      labs(x='',
           y='',
           title='Variance by Target Value') +
      scale_x_discrete(labels = c('Below Median', 'Above Median')) +
      theme(panel.grid.minor = element_blank(),
            panel.grid.major.x = element_blank())

  grid.arrange(plot.1, plot.2, ncol=2)
}
```

zn

Proportion of residential land zoned for large lots (over 25000 square feet)

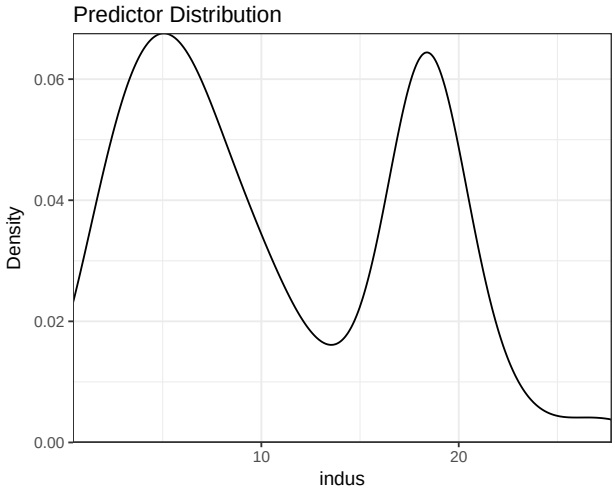
The 'zn' predictor is highly skewed. In fact, 43.7% of all the entries are 0. When examining the box plots it appears that all but two of the above median crime neighborhoods have 0 for 'zn'.

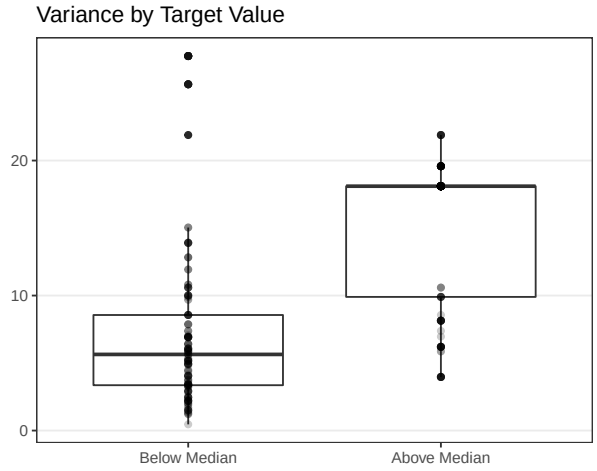


indus

Proportion of non-retail business acres per suburb

The density plot is bimodal indicating that indus is typically either 'low' or 'high'. The spread of the predictor in the box plot is showing that higher indus is associated with higher crime.





chas

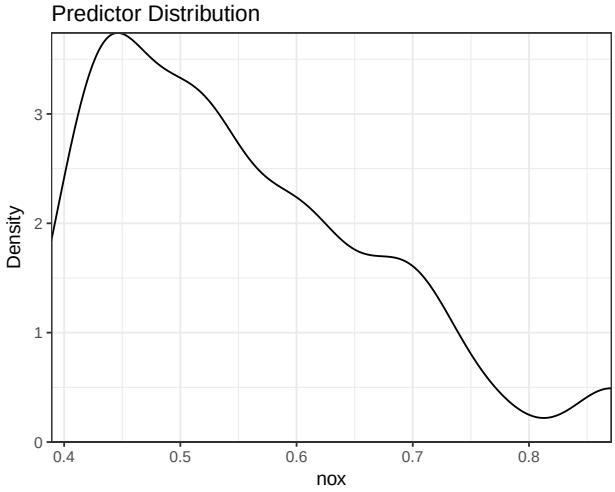
A dummy variable for whether the suburb borders the Charles River

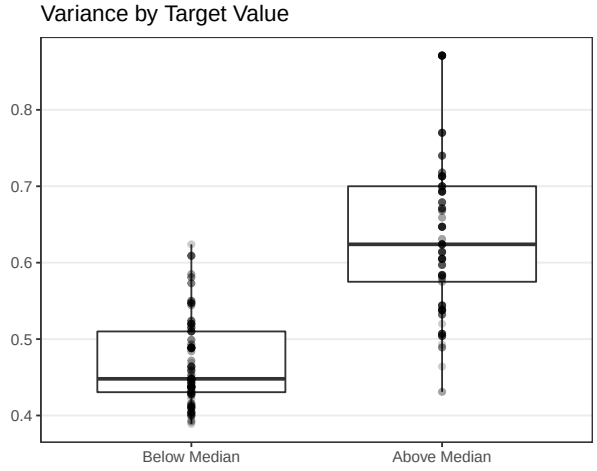
This is the only categorical variable in the predictor set. As per our textbook, 'when x is a dummy variable, it can be shown that the log odds are also a linear function of x'.

nox

Nitrogren oxides concentration (parts per 10 million)

The 'nox' variable has a slightly skew distribution and the fairly even variance in the boxplot. As per the textbook, 'when conducting a binary regression with a skewed predictor, it is often easiest to assess the need for x and log(x) by including them both in the model.'

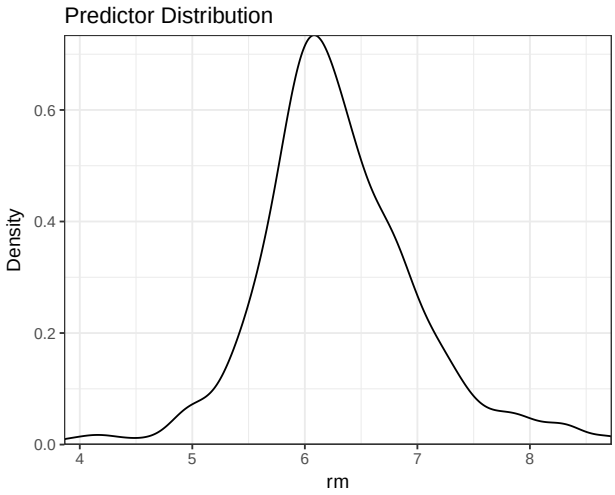


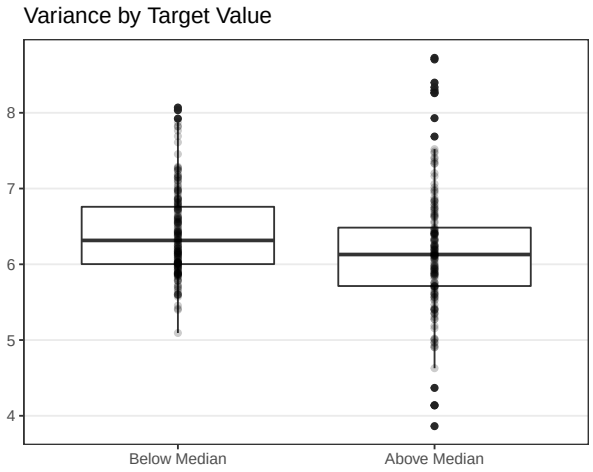


rm

Average number of rooms per dwelling

The 'rm' predictor is very nicely normally distributed with a nearly equal median value when seperated by target. Despite the ease of working with normally distributed data, the fact that the spread and median in the boxplot is so similar suggested that rm may not produce a statistically significant difference.

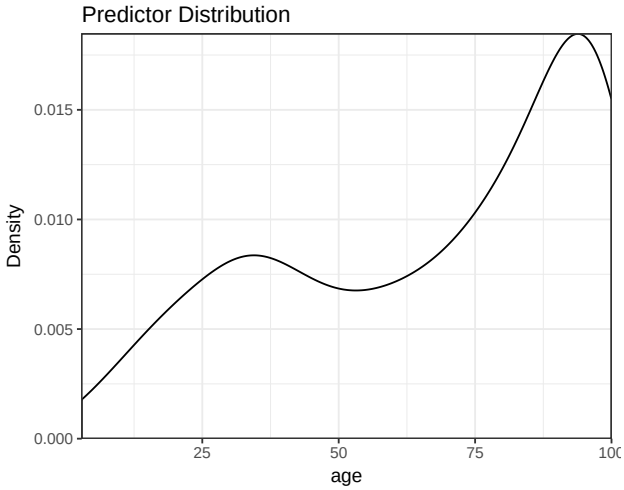


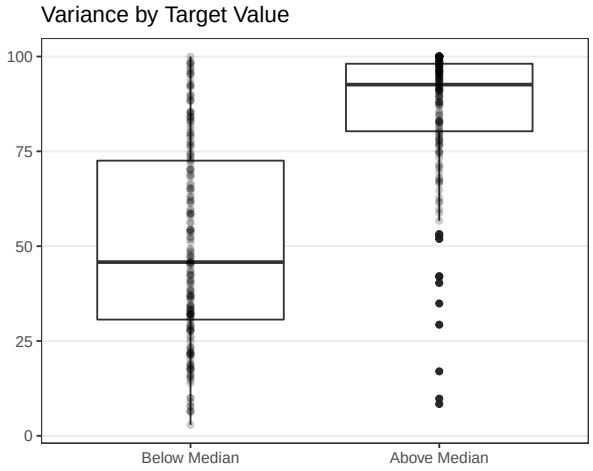


age

Proportion of owner-occupied units built prior to 1940

The 'age' predictor is heavily skewed indicating that a tranformation may be necessary. As with the 'nox' predictor, it may be required to add a log term to the model.

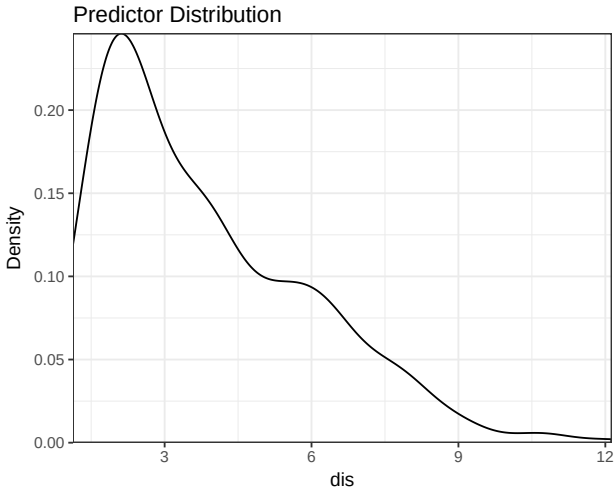


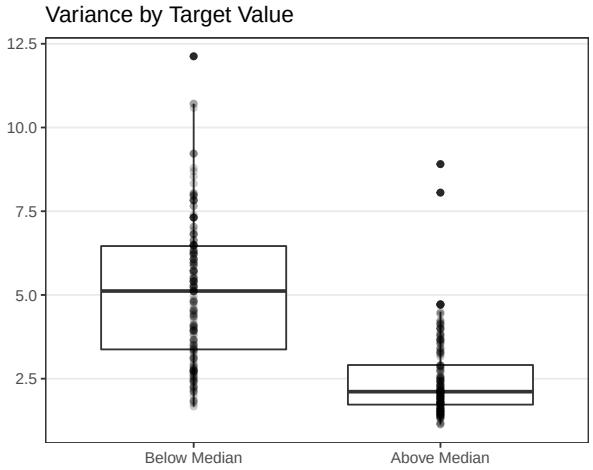


dis

Weighted mean of distances to five Boston employment centers

'dis' is similar to the above skewed predictors. There appears to be a demonstrable difference when considering higher crime locations and lower crime locations.

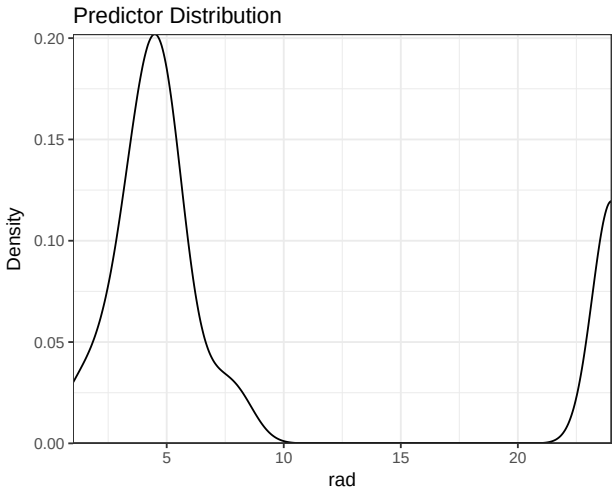


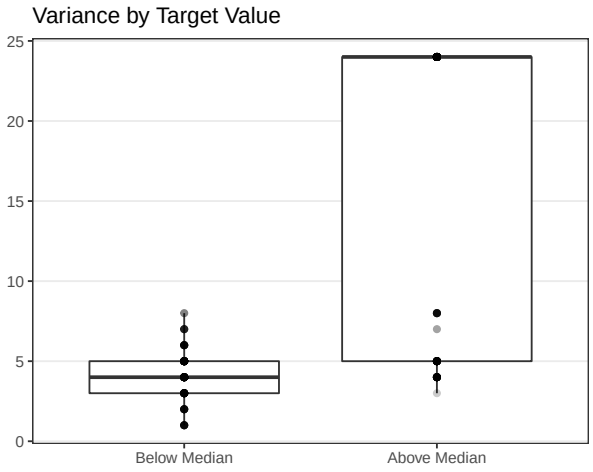


rad

index of accessibility to radial highways

The 'rad' variable, like 'zn' and 'indus' above, is strongly bimodal, perhaps even moreso than those variables. This again indicates that 'rad' can be identified as either 'high' or 'low'.

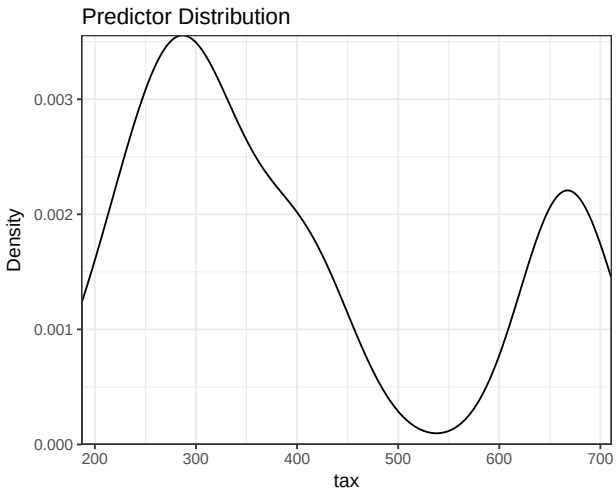


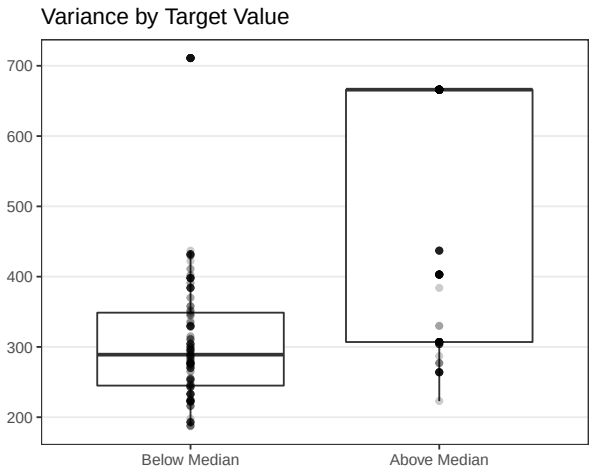


tax

Full-value property-tax rate per $10,000

'tax', like 'rad' is also highly bimodal. This indicates a large number of low tax areas and a small number of high tax areas. Interestingly, it is the high tax areas that seem to exhibit more crime.

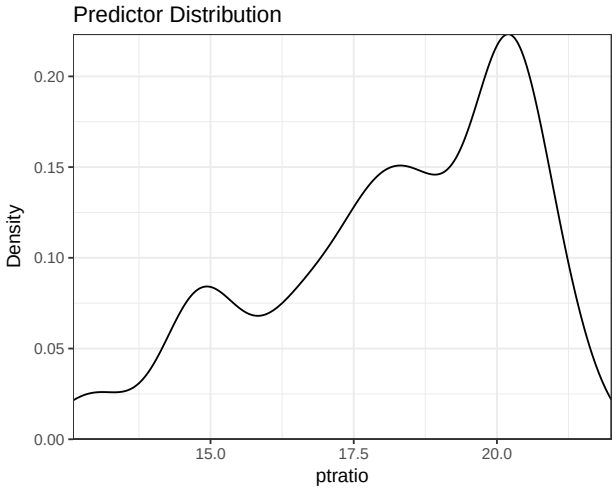


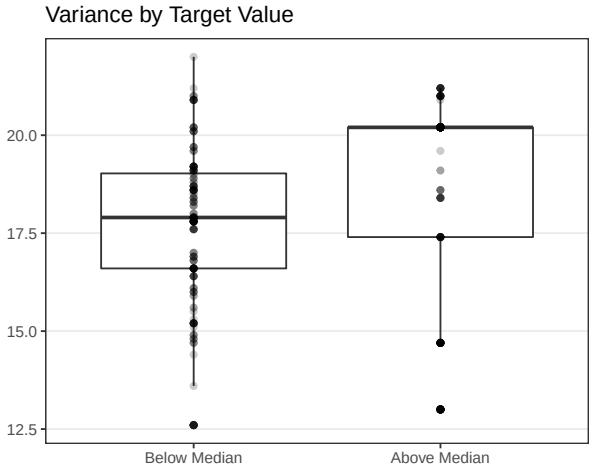


ptratio

pupil-teacher ratio by town

'ptratio' is skewed and may be in need of transforming.

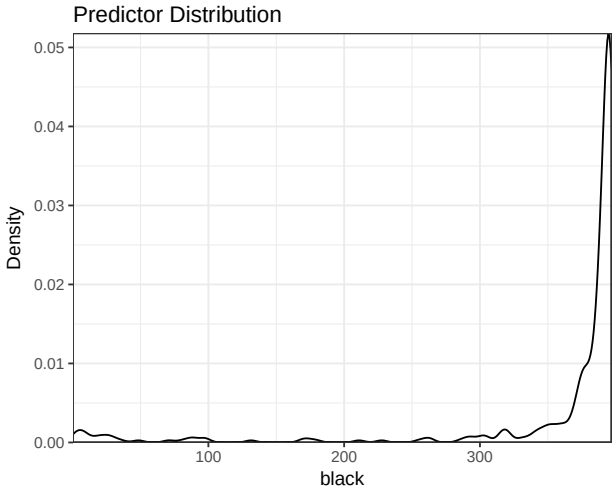


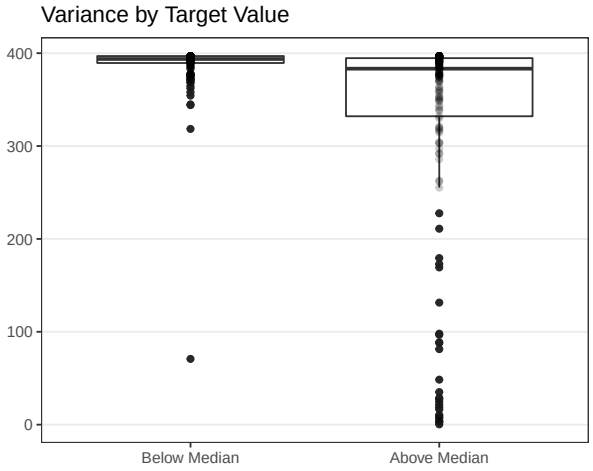


black

$1000(B_k - 0.63)^2$ **where** B_k **is the proportion of blacks by town**

'black' is more similar to the first predictor 'zn' in that nearly all the data is centered around a very small range with a number of outliers.

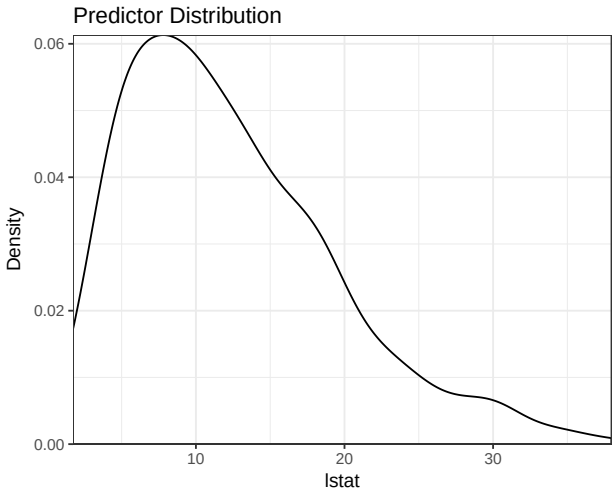


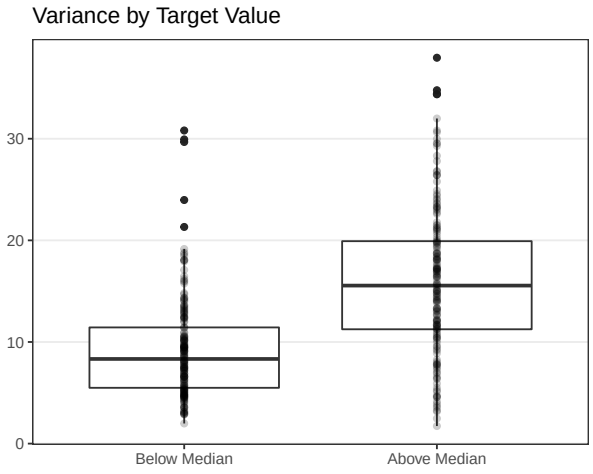


lstat

lower status of the population (percent)

'lstat' is fairly normally, similar to 'rm' however the boxplots indicate that the difference between low and high crime areas may be significant.

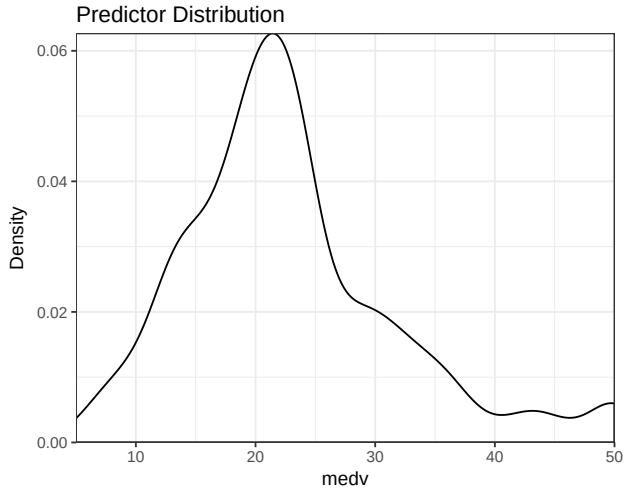


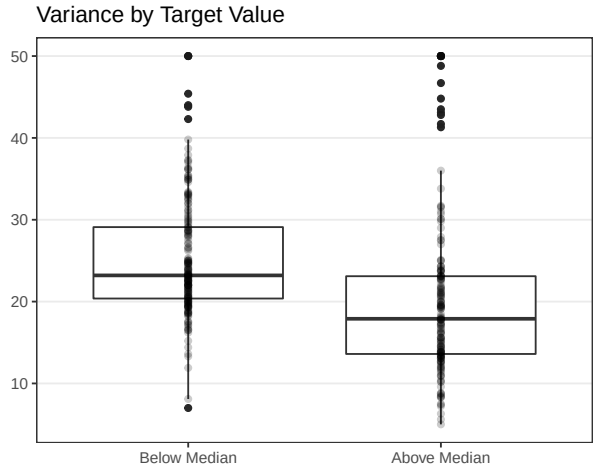


medv

Median value of owner-occupied homes in $1000s

The final predictor is nearly normal and the boxpots have a roughly equal variance.





Summary

Using the information gained by examining each predictor, I am able to make initial decisions in how to prepare the predictors for the first model. The 13 predictors can be grouped roughly as followed:

- Nearly Normal: rm, lstat, medv
- Bimodal / Highly Segregated: zn, indus, rad, tax, black
- Skewed: nox, age, dis, ptratio
- Categorical: chas

Data Preperation

I will begin by altering the Bimodal / Highly Segregated predictors by transforming them into categorical data. I found a number of articles online indicating that this should be done with care because bucketing continuous variables may result in the lose of some of the predictive power that might be present in the variable. These concerns can be eased somewhat due to the fact that the data is highly segregated (the question is almost in essance asking for a 'high'/'low' or 'none'/'any' response) and that there are several other continuous predictors.

I will be converting zn, indus, rad, tax and black into categorical data. The split point was determined by viewing the density plot above and selecting a value that attempted to accurately identify the information contained by the predictor. For 'zn', the split is between 0 and any other value and for the rest a value near the median was selected.

```
training.cat <- training %>%
  mutate(chas.cat = factor(chas),
         zn.cat = factor(ifelse(zn == 0, 0, 1)),
         indus.cat = factor(ifelse(indus < 14, 0, 1)),
         rad.cat = factor(ifelse(rad <= 15, 0, 1)),
         tax.cat = factor(ifelse(tax <= 540, 0, 1)),
         black.cat = factor(ifelse(black <= 390, 0, 1))) %>%
  select(-chas, -zn, -indus, -rad, -tax, -black)
```

There are several other transformations that may aid my model but first I would like to build a model with the data I have collected thus far.

Model 1

For this model I will only be using the 6 categorical predictors. As per the textbook I know that each predictor will be linearly related to the log odds and thus the model will be valid. In addition, I may be able to glean helpful information from the model that will aid in the development of my other two models.

```
model.1 <- glm(target ~ chas.cat + zn.cat + indus.cat + rad.cat + tax.cat + black.cat,
               data=training.cat, family=binomial)
summary(model.1)
```

```
##
## Call:
## glm(formula = target ~ chas.cat + zn.cat + indus.cat + rad.cat +
##     tax.cat + black.cat, family = binomial, data = training.cat)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.11322  -0.69019  -0.40967   0.00011   2.24483
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.3173     0.2707  -1.172  0.24105
## chas.cat1      0.8937     0.5708   1.566  0.11744
## zn.cat1       -1.1224     0.3499  -3.208  0.00134 **
## indus.cat1     2.4368     0.4696   5.189 2.11e-07 ***
## rad.cat1      39.0288  4834.8353   0.008  0.99356
## tax.cat1     -21.1679  4708.9460  -0.004  0.99641
## black.cat1    -0.9959     0.3157  -3.155  0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.12  on 372  degrees of freedom
## Residual deviance: 259.50  on 366  degrees of freedom
## AIC: 273.5
##
## Number of Fisher Scoring iterations: 18
```

Looking at the summary there are two predictors, 'rad.cat' and 'tax.cat', that appear to be completely worthless with p-value of essentially 1. However, upon further inspection, it turns out there is a reasonable explanation for this result.

```
data_frame(name=names(car::vif(model.1)), value=car::vif(model.1)) %>%
  spread(1, 2) %>%
  kable()
```

black.cat	chas.cat	indus.cat	rad.cat	tax.cat	zn.cat
1.010308	1.010948	1.014106	19.45601	19.45601	1.013531

The 'rad.cat' and 'tax.cat' have wide std. error due to high collinearity. In fact, it turns out that these predictors are nearly identical, sharing over 98% of the same values!

```
nrow(training.cat %>% filter(rad.cat == tax.cat)) / nrow(training.cat)
```

```
## [1] 0.9865952
```

Apparently, being close to a radial highway essentially means that you pay higher taxes and vice versa. It is clear that we do not need both of this variables. I am removing 'rad.cat' as 'tax.cat' ends up being statistically significant in the remaining models. I need to be cognizant of the fact that this may be overfitting the training data and explore alternate possibilities later on.

```
model.1.2 <- glm(target ~ chas.cat + zn.cat + indus.cat + tax.cat + black.cat,
                 data=training.cat, family=binomial)
summary(model.1.2)
```

```
##
## Call:
## glm(formula = target ~ chas.cat + zn.cat + indus.cat + tax.cat +
##     black.cat, family = binomial, data = training.cat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6091  -0.6897  -0.4094   0.4214   2.2454
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.3216     0.2644  -1.216 0.223929
## chas.cat1     0.9248     0.5612   1.648 0.099415 .
## zn.cat1      -1.1224     0.3500  -3.207 0.001340 **
## indus.cat1    2.4378     0.4697   5.191  2.1e-07 ***
## tax.cat1      1.2538     0.6375   1.967 0.049226 *
## black.cat1   -0.9932     0.2995  -3.317 0.000911 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.12  on 372  degrees of freedom
## Residual deviance: 297.58  on 367  degrees of freedom
## AIC: 309.58
##
## Number of Fisher Scoring iterations: 5
```

Using `MASS::stepAIC(model.1.2)` reveals no suggested modifications to the selected predictors. Multicollinearity is no longer a problem for this model. Unfortunately, an anova test has indicated that the removal of the single predictor has had a statistically significant impact on our model's predictive ability. [See Appendix for diagnostics]

```
data_frame(name=names(car::vif(model.1.2)), value=car::vif(model.1.2)) %>%
  spread(1, 2) %>%
  kable()
```

black.cat	chas.cat	indus.cat	tax.cat	zn.cat
1.009775	1.015554	1.72262	1.7103	1.051999

```
anova(model.1, model.1.2, test='Chisq')
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
366	259.5037	NA	NA	NA
367	297.5837	-1	-38.07999	0

$$\widehat{y} = 0.92 \times chas + -1.12 \times zn + 2.44 \times indus + 1.25 \times tax + -0.99 \times black + -0.32$$

Model 2

As per the textbook, it is suggested that I should add a log term for skewed predictors along with the original term in an effort to determine which predictors are necessary for a useful model. I have found four skewed predictors (nox, age, dis and ptratio). For the second model I will include all the categorical variables from model 1 and all the continuous variables and their logs.

```
training.log <- training.cat %>%
  mutate(l.nox = log(nox),
         l.age = log(age),
         l.dis = log(dis),
         l.ptratio = log(ptratio))
model.2 <- glm(target ~ ., data=training.log, family=binomial)
```

Using `MASS::stepAIC(model.2)` reduces the number of predictors from 17 to 11.

```
model.2 <- glm(target ~ nox + age + dis + ptratio + medv + chas.cat + rad.cat + tax.cat +
                 black.cat + l.age + l.dis, data=training.log, family=binomial)
```

```
to.disp <- data_frame(name=names(car::vif(model.2)), value=car::vif(model.2)) %>%
  spread(1, 2)
to.disp[, 1:6] %>%
  kable()
```

age	black.cat	chas.cat	dis	l.age	l.dis
8.27422	1.091501	1.120434	24.42482	7.300693	29.85182

```
to.disp[, 7:11] %>%
  kable()
```

medv	nox	ptratio	rad.cat	tax.cat
2.385685	4.279003	1.667345	29.74109	29.74109

Just like before, there is a problem with multi-collinearity between 'rad.cat' and 'tax.cat'. There appears to be a problem with colinearity between dis and log(dis) for obvious reasons. Examining whether this is a valid regression may also aid insight into whether both predictors are required. [See Appendix for diagnostics]

```
model.2 <- glm(target ~ nox + age + dis + ptratio + medv + chas.cat + tax.cat + black.cat
               + l.age + l.dis, data=training.log, family=binomial)
summary(model.2)
```

```
##
## Call:
## glm(formula = target ~ nox + age + dis + ptratio + medv + chas.cat +
##     tax.cat + black.cat + l.age + l.dis, family = binomial, data = training.log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4251  -0.2468  -0.0280   0.2164   3.3259
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -43.09376    7.48749  -5.755 8.64e-09 ***
## nox          54.24371    8.31978   6.520 7.04e-11 ***
## age           0.06918    0.02395   2.888  0.00388 **
## dis          -1.51923    0.60585  -2.508  0.01216 *
## ptratio       0.41281    0.13074   3.157  0.00159 **
## medv          0.18378    0.04154   4.425 9.66e-06 ***
## chas.cat1     1.20827    0.72462   1.667  0.09542 .
## tax.cat1      1.59235    0.71530   2.226  0.02601 *
## black.cat1   -1.37349    0.43525  -3.156  0.00160 **
## l.age        -2.06122    0.92770  -2.222  0.02629 *
## l.dis         9.75739    2.48134   3.932 8.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.12  on 372  degrees of freedom
## Residual deviance: 169.22  on 362  degrees of freedom
## AIC: 191.22
##
## Number of Fisher Scoring iterations: 7
```

```
anova(model.1.2, model.2, test='Chisq')
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
367	297.5837	NA	NA	NA
362	169.2226	5	128.3612	0

The anova test indicates that the second model is statistically significantly more accurate than model 1.

$$\widehat{y} = 54.24 \times nox + 0.07 \times age + -1.52 \times dis + 0.41 \times ptratio +$$
$$0.18 \times medv + 1.21 \times chas + 1.59 \times tax + -1.3 \times black + -2.06 \times log(age) +$$
$$9.75 \times log(dis) + -43.1$$

Model 3

In an effort to reduce the number of predictors without significantly hindering predictive ability, I will use BIC which features a heavier penalty term than AIC (which was used for Model 2)

Using `MASS::stepAIC(model.3, k=log(nrow(training.log)))` produces the following model. Just as with model 1 and 2, the high collinearity between rad.cat and tax.cat will need to be addressed.

```
model.3 <- glm(target ~ nox + age + dis + ptratio + medv + tax.cat + black.cat + l.dis,
               data=training.log, family=binomial)
summary(model.3)
```

```
##
## Call:
## glm(formula = target ~ nox + age + dis + ptratio + medv + tax.cat +
##     black.cat + l.dis, family = binomial, data = training.log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.39320  -0.24511  -0.04091   0.20586   3.12968
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -45.14510    7.10662  -6.353 2.12e-10 ***
## nox          50.87908    7.79984   6.523 6.89e-11 ***
## age           0.02767    0.01239   2.234 0.025494 *
## dis          -1.28043    0.56253  -2.276 0.022835 *
## ptratio       0.36924    0.12418   2.974 0.002944 **
## medv          0.17268    0.04058   4.255 2.09e-05 ***
## tax.cat1      1.48422    0.70195   2.114 0.034479 *
## black.cat1   -1.40124    0.42031  -3.334 0.000857 ***
## l.dis         8.54481    2.28625   3.737 0.000186 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.12  on 372  degrees of freedom
## Residual deviance: 175.85  on 364  degrees of freedom
## AIC: 193.85
##
## Number of Fisher Scoring iterations: 7
```

This model features two fewer predictors than the second model, however an anova test indicates that it's deviance is worse. Howeve the AIC for model 3 is 193.85 and model 2 is 191.22 which is a very small difference for the two additional predictors in model 2. In addition this is the only model where every predictor is statistically significant. [See Appendix for diagnostic]

$$\widehat{y} = 50.88 \times nox + 0.03 \times age + -1.28 \times dis + 0.37 \times ptratio +$$
$$0.17 \times medv + 1.48 \times tax + -1.4 \times black + 8.54 \times log(dis) + -45.15$$

Select Model

I have developed three models, but must select the one to use to run on the evaluation data. It has already been established that model 2 has the best predictive ability on the training data and that difference is statistically significant. Further examination will help determine which model to select.

R^2 does not exist for logistic regression in the traditional sense. However, there are a number of so called pseudo R^2 terms that can be analized.

```
data_frame(name=names(pscl::pR2(model.1.2)), value=pscl::pR2(model.1.2)) %>%
  spread(1, 2) %>%
  kable()
```

G2	llh	llhNull	McFadden	r2CU	r2ML
218.5358	-148.7919	-258.0598	0.4234209	0.5916967	0.443388

```
data_frame(name=names(pscl::pR2(model.2)), value=pscl::pR2(model.2)) %>%
  spread(1, 2) %>%
  kable()
```

G2	llh	llhNull	McFadden	r2CU	r2ML
346.897	-84.61129	-258.0598	0.6721252	0.8079716	0.6054537

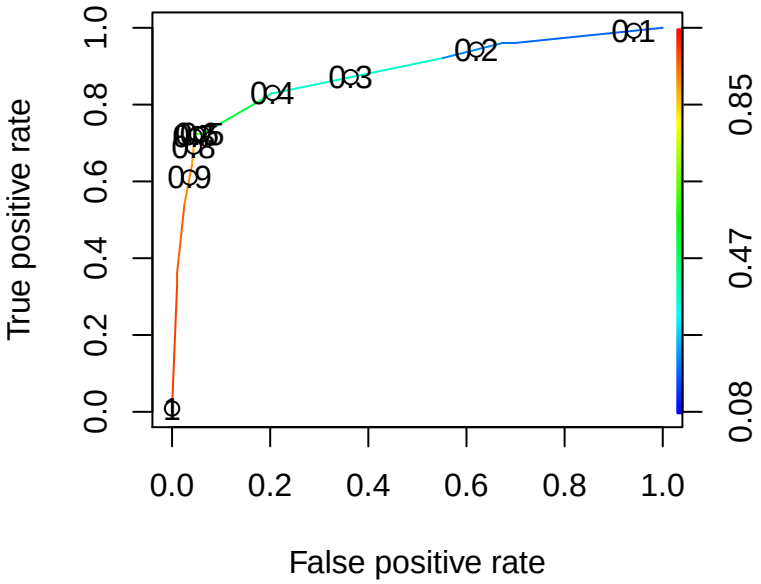
```
data_frame(name=names(pscl::pR2(model.3)), value=pscl::pR2(model.3)) %>%
  spread(1, 2) %>%
  kable()
```

G2	llh	llhNull	McFadden	r2CU	r2ML
340.27	-87.92475	-258.0598	0.6592853	0.7985336	0.5983813

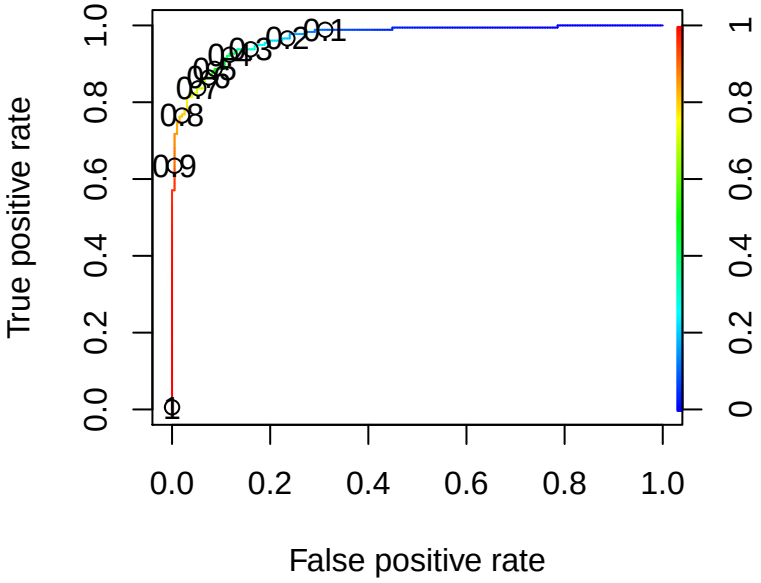
All of these measures, and specifically McFadden, support the anova test's conclusion that model 2 is the strongest with model 3 just slightly behind in predictive ability.

Finally, we will examine the predictive ability of the models by examining the confusion matrix of each model while using the testing data that had been previously set aside. Exploring the ROC curve of all three models indicates that 0.5 is a reasonable threshold for prediction that helps to maximize the true positive rate while minimizing the false positive rate. This follows with the assignment requirement of using a 0.5 threshold.

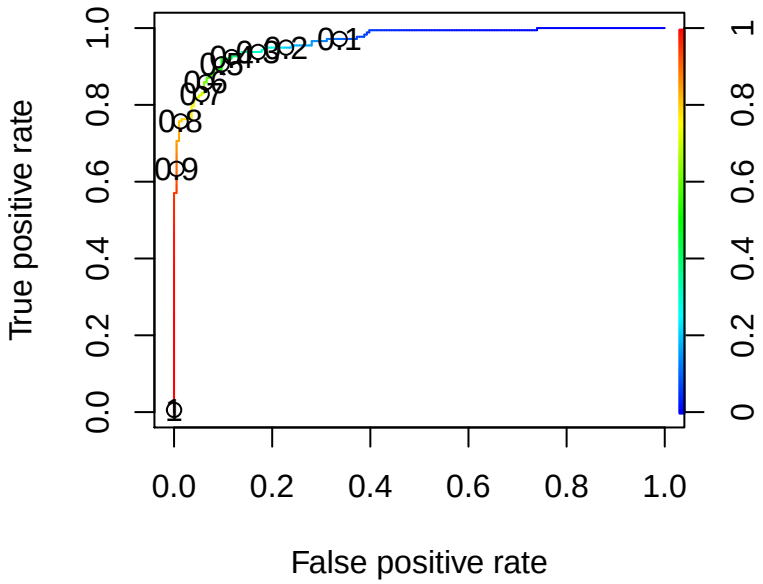
```
ROCRPred <- prediction(predict(model.1.2, type='response'), training.cat$target)
ROCRPref <- performance(ROCRPred, 'tpr', 'fpr')
plot(ROCRPref, colorize=TRUE, print.cutoffs.at = seq(0.1, by=0.1))
```




```
ROCRPred <- prediction(predict(model.2, type='response'), training.log$target)
ROCRPref <- performance(ROCRPred, 'tpr', 'fpr')
plot(ROCRPref, colorize=TRUE, print.cutoffs.at = seq(0.1, by=0.1))
```



```
ROCRPred <- prediction(predict(model.3, type='response'), training.cat$target)
ROCRPref <- performance(ROCRPred, 'tpr', 'fpr')
plot(ROCRPref, colorize=TRUE, print.cutoffs.at = seq(0.1, by=0.1))
```



I created confusion matrices on the previously set aside testing data set. While all three models perform well (79%, 83% and 87% respectively), it is the third model that performs the best on the validation data. This is despite the fact that it has a lower predictive value on the training data. This indicates to me that model 2 may suffer from some overfitting.

```
predictions <- ifelse(predict(model.1.2, newdata=testing.cat, type='response') < 0.5, 0, 1)
caret::confusionMatrix(table(predicted=predictions, actual = testing.cat$target))
```

```
## Confusion Matrix and Statistics
##
##          actual
## predicted  0  1
##         0 38 16
##         1  3 36
##
##                Accuracy : 0.7957
##                  95% CI : (0.6995, 0.8723)
##     No Information Rate : 0.5591
##     P-Value [Acc > NIR] : 1.511e-06
##
##                   Kappa : 0.599
##  Mcnemar's Test P-Value : 0.005905
##
##             Sensitivity : 0.9268
##             Specificity : 0.6923
##          Pos Pred Value : 0.7037
```

```
##          Neg Pred Value : 0.9231
##              Prevalence : 0.4409
##          Detection Rate : 0.4086
##    Detection Prevalence : 0.5806
##       Balanced Accuracy : 0.8096
##
##        'Positive' Class : 0
##
```

```
predictions <- ifelse(predict(model.2, newdata=testing.log, type='response') < 0.5, 0, 1)
caret::confusionMatrix(table(predicted=predictions, actual = testing.log$target))
```

```
## Confusion Matrix and Statistics
##
##          actual
## predicted  0  1
##         0 32  6
##         1  9 46
##
##                Accuracy : 0.8387
##                  95% CI : (0.748, 0.9068)
##     No Information Rate : 0.5591
##     P-Value [Acc > NIR] : 9.557e-09
##
##                   Kappa : 0.6703
##  Mcnemar's Test P-Value : 0.6056
##
##             Sensitivity : 0.7805
##             Specificity : 0.8846
##          Pos Pred Value : 0.8421
##          Neg Pred Value : 0.8364
##              Prevalence : 0.4409
##          Detection Rate : 0.3441
##    Detection Prevalence : 0.4086
##       Balanced Accuracy : 0.8326
##
##        'Positive' Class : 0
##
```

```
predictions <- ifelse(predict(model.3, newdata=testing.log, type='response') < 0.5, 0, 1)
caret::confusionMatrix(table(predicted=predictions, actual = testing.log$target))
```

```
## Confusion Matrix and Statistics
##
##          actual
## predicted  0  1
##         0 34  5
##         1  7 47
##
##                Accuracy : 0.871
##                  95% CI : (0.7855, 0.9315)
##     No Information Rate : 0.5591
##     P-Value [Acc > NIR] : 9.719e-11
##
##                   Kappa : 0.7369
```

```
##  Mcnemar's Test P-Value : 0.7728
##
##             Sensitivity : 0.8293
##             Specificity : 0.9038
##          Pos Pred Value : 0.8718
##          Neg Pred Value : 0.8704
##              Prevalence : 0.4409
##          Detection Rate : 0.3656
##    Detection Prevalence : 0.4194
##       Balanced Accuracy : 0.8666
##
##        'Positive' Class : 0
##
```

I have decided to select model 3. Model 3 has demonstrated strong predictive ability on the testing set and has the added benefit of being less complex (featuring fewer predictors) than model 2. As mentioned above I believe that model 2 may suffer from some overfitting which leads me to believe that model 3 will perform better against the evaluation data.

Below are the results on the evaluation data set.

```
final.data <- read_csv('https://raw.githubusercontent.com/brian-cuny/621week3/master/crime-evaluation-da
  mutate(chas.cat = factor(chas),
         zn.cat = factor(ifelse(zn == 0, 0, 1)),
         indus.cat = factor(ifelse(indus < 14, 0, 1)),
         rad.cat = factor(ifelse(rad <= 15, 0, 1)),
         tax.cat = factor(ifelse(tax <= 540, 0, 1)),
         black.cat = factor(ifelse(black <= 390, 0, 1))) %>%
  select(-chas, -zn, -indus, -rad, -tax, -black) %>%
  mutate(l.nox = log(nox),
         l.age = log(age),
         l.dis = log(dis),
         l.ptratio = log(ptratio))
to.output <- data_frame(prob = predict(model.3, newdata=final.data, type='response'),
                        class = ifelse(prob < 0.5, 0, 1)) %>%
             mutate(id = row_number()) %>%
             select(id, prob, class)
to.output %>%
  kable()
```

id	prob	class
1	0.2511047	0
2	0.9261767	1
3	0.9444495	1
4	0.7765832	1
5	0.0674186	0
6	0.0288519	0
7	0.1415019	0
8	0.0247451	0
9	0.0068603	0
10	0.0044085	0
11	0.5822744	1
12	0.3524640	0
13	0.8055793	1
14	0.8896028	1

id	prob	class
15	0.8080304	1
16	0.1314672	0
17	0.2742860	0
18	0.9771524	1
19	0.0270485	0
20	0.0011449	0
21	0.0006937	0
22	0.0407297	0
23	0.1322441	0
24	0.1704860	0
25	0.1241227	0
26	0.3805960	0
27	0.0035405	0
28	0.9999274	1
29	0.9971820	1
30	0.8715208	1
31	0.9997025	1
32	0.9995160	1
33	0.9993635	1
34	0.9999146	1
35	0.9998795	1
36	0.9999166	1
37	0.9998696	1
38	0.9987898	1
39	0.6885837	1
40	0.6987914	1

In conclusion, I have been able to make predictions on a variety of neighborhoods of a major city in order to determine whether the neighborhood is likely to be above the median crime rate or below the median crime rate. From the initial set of predictors, a subset was selected that demonstrated strong predictive ability. Testing the data on a withheld set produced 87% accuracy. It was this model that was used on the evaluation set. It is my hope that the model wil be able to produce similar results on the evaluation data.

Appendix

I created a function to produce marginal model plots to determine whether each model is valid.

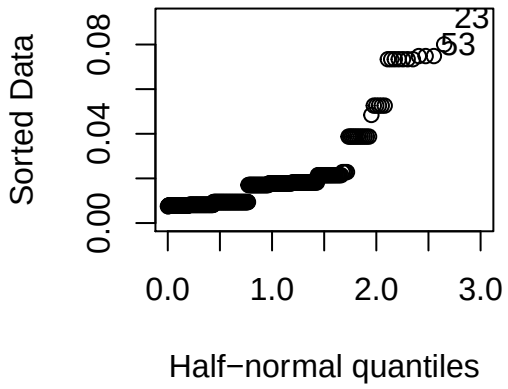
```
Marginal.Model.Two <- function(p){
  yhat <- predict(model.2, type='response')
  p <- ggplot(training.log, aes_string(p, 'target')) +
    geom_point() +
    geom_smooth(method='loess', se=FALSE) +
    geom_smooth(aes(y=yhat), method='loess', se = FALSE, color='red', linetype='dotted')
  return(p)
}

Marginal.Model.Three <- function(p){
  yhat <- predict(model.3, type='response')
  p <- ggplot(training.log, aes_string(p, 'target')) +
    geom_point() +
```

```
    geom_smooth(method='loess', se=FALSE) +
    geom_smooth(aes(y=yhat), method='loess', se = FALSE, color='red', linetype='dotted')
  return(p)
}
```

Model 1

```
faraway::halfnorm(hatvalues(model.1.2))
```

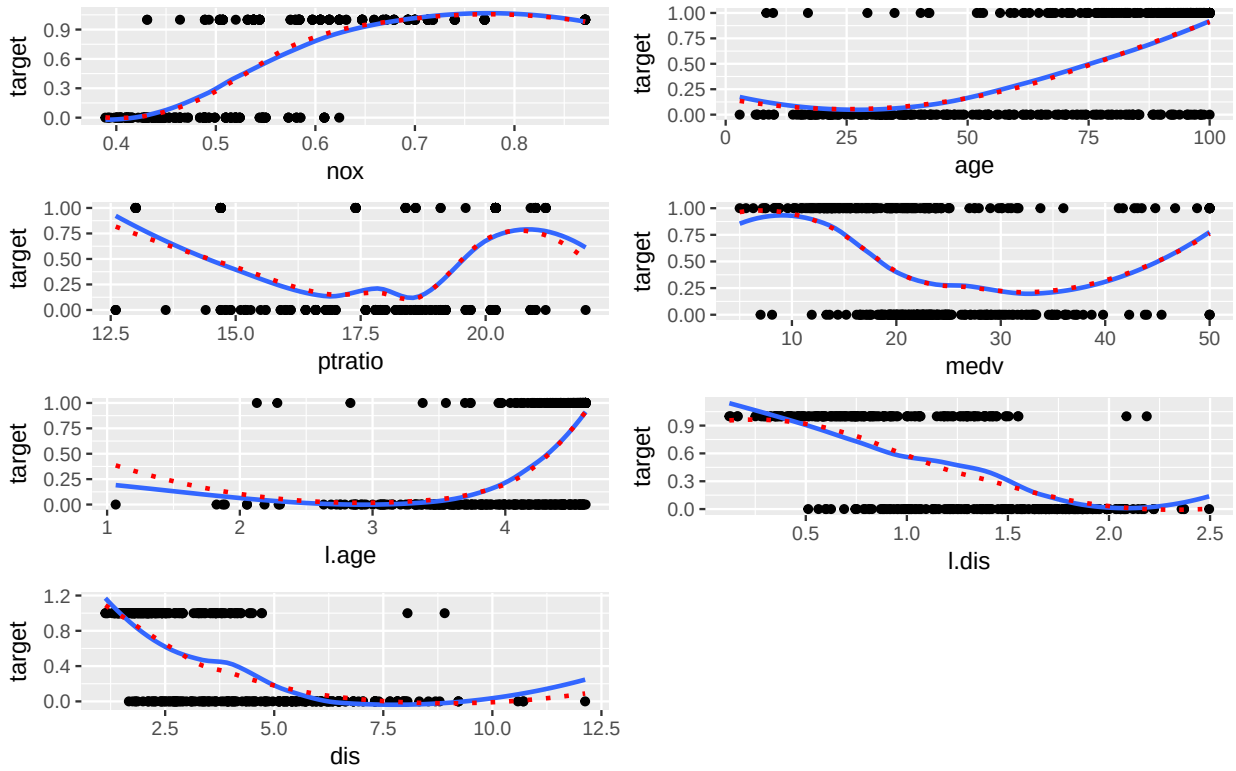


The halfnorm plot for model 1 appears to show a high leverage point in 237. As this is a single point across the entire model, and it does not appear to be too far off the quantile line, it is unlikely to be heavily skewing the model.

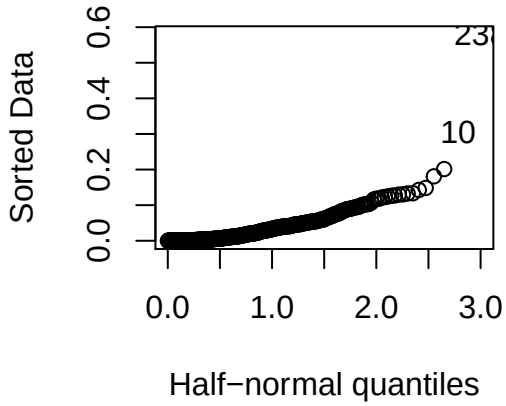
Model 2

I must examine the non-categorical predictor variables' marginal model plots.

```
require(gridExtra)
grid.arrange(Marginal.Model.Two('nox'), Marginal.Model.Two('age'),
             Marginal.Model.Two('ptratio'), Marginal.Model.Two('medv'),
             Marginal.Model.Two('l.age'), Marginal.Model.Two('l.dis'),
             Marginal.Model.Two('dis'), ncol=2)
```



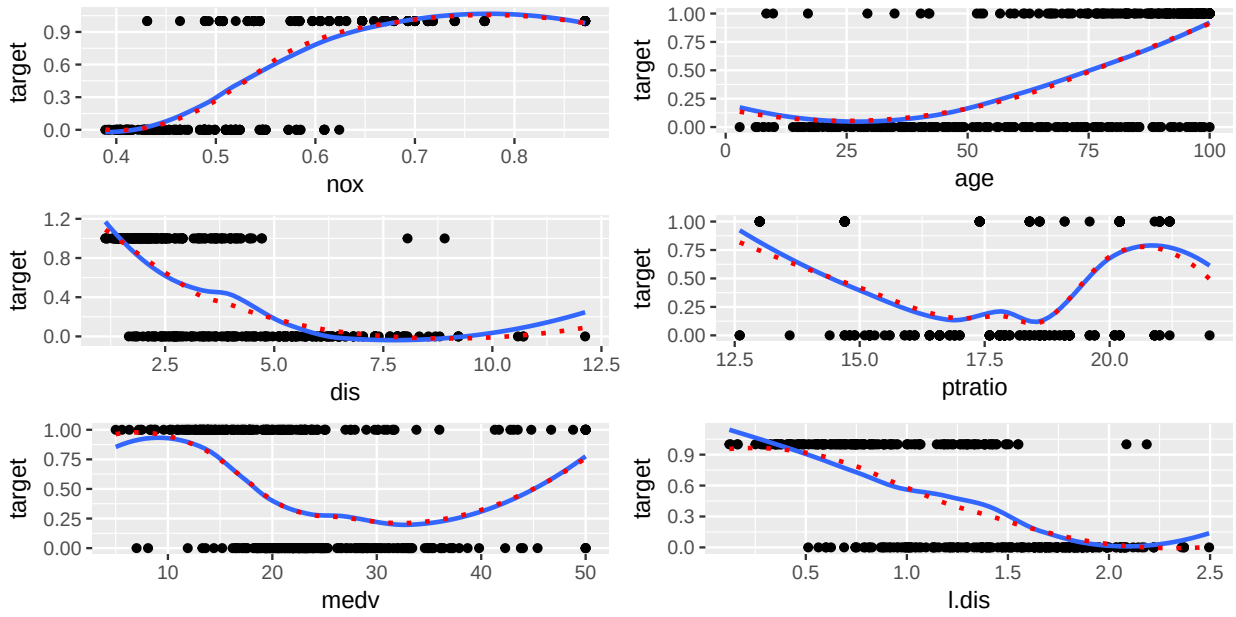

```
faraway::halfnorm(hatvalues(model.2))
```



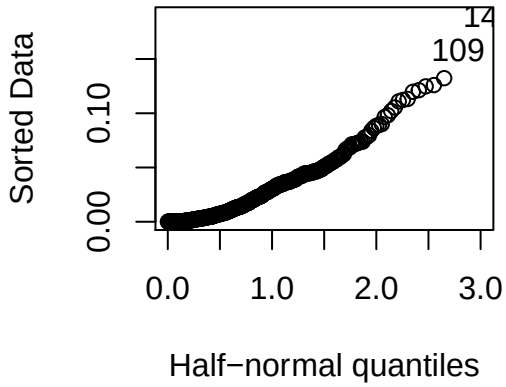
The marginal model plots are all fairly accurate, with a note that l.dis appears to deviate the most. Observation 238 appears to be a significant outlier. I examined this point further on a leverage plot and it sits just below Cook's distance. There does not appear to anything fautly about the data and, combined with it not being past Cook's distance, will be left in the model.

Model 3

```
require(gridExtra)
grid.arrange(Marginal.Model.Two('nox'), Marginal.Model.Two('age'),
             Marginal.Model.Two('dis'), Marginal.Model.Two('ptratio'),
             Marginal.Model.Two('medv'), Marginal.Model.Two('l.dis'), ncol=2)
```




```
faraway::halfnorm(hatvalues(model.3))
```



The marginal model plots for all the predictors are very close to the expected values. This indicates a valid model. There appears to be no bad leverage points in this data set. When view on a leverage plot, observation 223 may be an outlier, but it is a low leverage point and thus will be left alone.