Data 621 - Homework 3

Group 2: William Aiken, Donald Butler, Michael Ippolito, Bharani Nittala, and Leticia Salazar

11-06-2022

Contents

Overview:

In this homework assignment, you will explore, analyze and model 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).

Objective:

Build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels. You will provide classifications and probabilities for the evaluation data set using your binary logistic regression model. You can only use the variables given to you (or variables that you derive from the variables provided).

Description:

Below is a short description of the variables of interest in the data set:

- zn: proportion of residential land zoned for large lots (over 25000 square feet)(predictor variable)
- indus: proportion of non-retail business acres per suburb (predictor variable)

- chas: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)
- nox: nitrogen oxides concentration (parts per 10 million) (predictor variable)
- rm: average number of rooms per dwelling (predictor variable)
- age: proportion of owner-occupied units built prior to 1940 (predictor variable)
- dis: weighted mean of distances to five Boston employment centers (predictor variable)
- rad: index of accessibility to radical highways (predictor variable)
- tax: full-value property-tax rate per $10,000 (predictor variable)
- ptratio: pupil-teacher ratio by town (predictor variable)
- black: $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town (predictor variable)
- lstat: lower status of the population (percent)(predictor variable)
- medv: median value of owner-occupied homes in $1000s (predictor variable)
- target: whether the crime rate is above the median crime rate (1) or not (0) (response variable)

Load Libraries:

These are the libraries used to explore, prepare, analyze and build our models

```
library(tidyverse)
library(caret)
library(pROC)
library(corrplot)
library(GGally)
library(psych)
library(car)
library(kableExtra)
library(gridExtra)
library(performance)
library(faraway)
```

Load Data set:

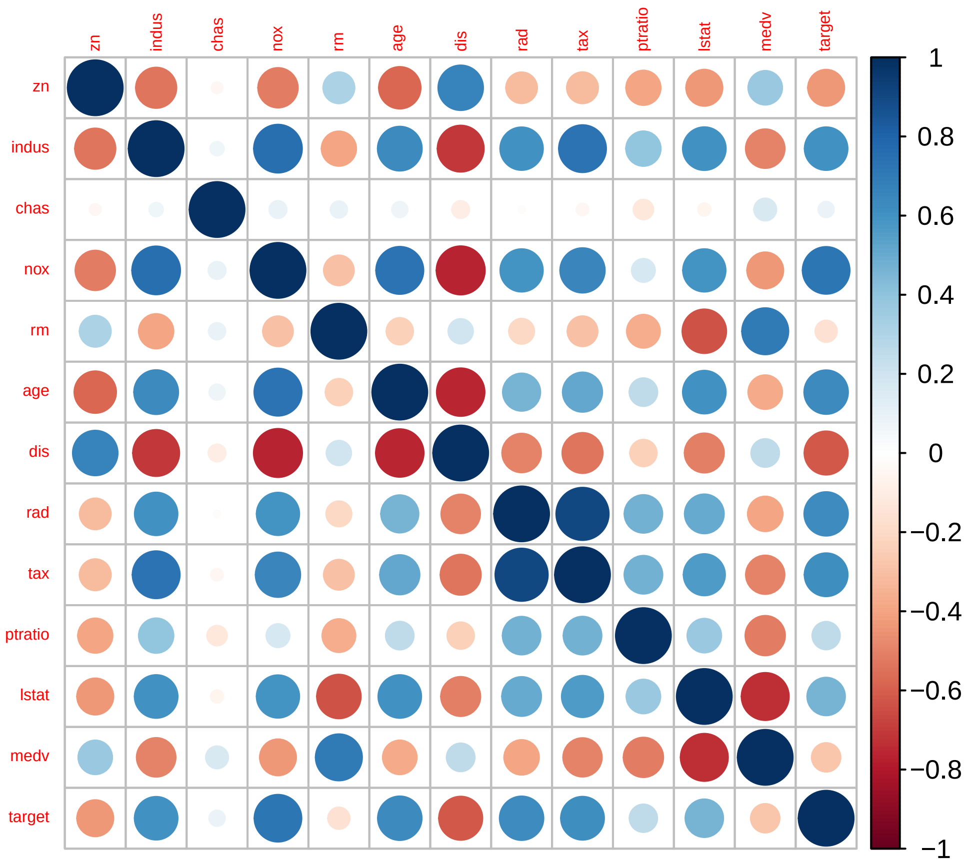
We have included the original data sets in our GitHub account and read from this location. Our data set includes 466 records and 13 variables.

```
## Rows: 466
## Columns: 13
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 100, 20, 0~
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5.19, 3.6~
## $ 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, 0.515,~
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519, 6.316,~
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38.1, 19.1,~
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896, 1.6582~
```

```
## $ rad     <int> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5, 5, 24, ~
## $ tax     <int> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330, 398, 66~
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, 16.4, 19~
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5.68, 9.25~
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20.9, 24.8~
## $ target  <int> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0,~
```

Data Exploration:

The correlation plot below is measuring the degree of linear relationship within the training data set. The values in which this is measured falls between -1 and +1, with +1 being a stronger correlation.



To give more insight on our data set we used the `summary()` and `describe()` functions below:

```
# summarizing data set
summary(dftrain)
```

```
##        zn             indus            chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
```

```
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio         lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
describe(dftrain)
```

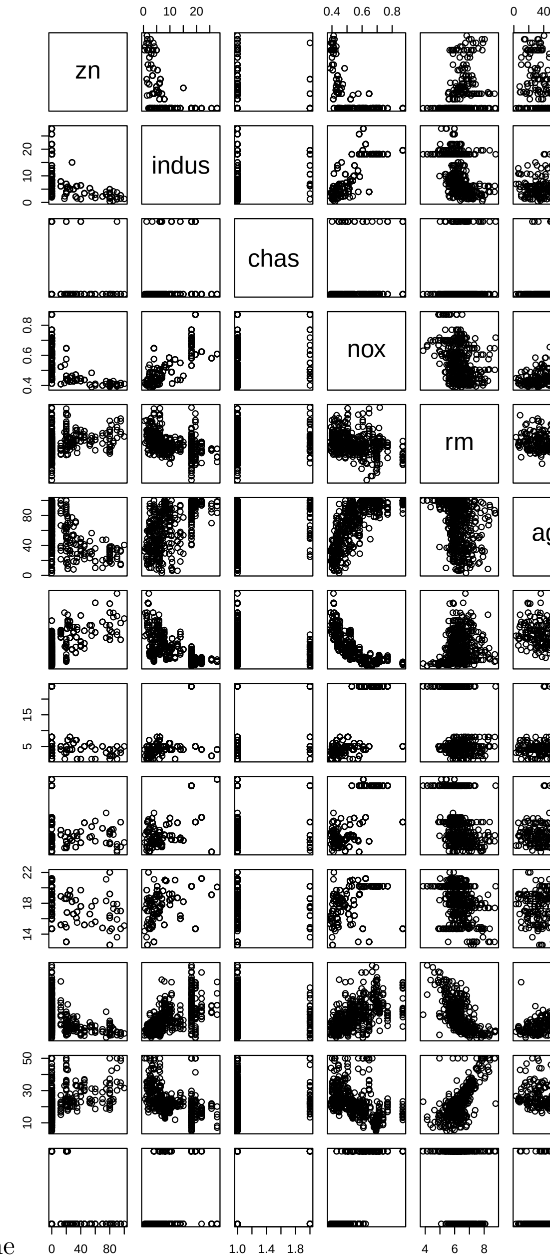
```
##         vars   n   mean     sd median trimmed    mad    min    max  range  skew
## zn         1 466  11.58  23.36   0.00    5.35   0.00   0.00 100.00 100.00  2.18
## indus      2 466  11.11   6.85   9.69   10.91   9.34   0.46  27.74  27.28  0.29
## chas       3 466   0.07   0.26   0.00    0.00   0.00   0.00   1.00   1.00  3.34
## nox        4 466   0.55   0.12   0.54    0.54   0.13   0.39   0.87   0.48  0.75
## rm         5 466   6.29   0.70   6.21    6.26   0.52   3.86   8.78   4.92  0.48
## age        6 466  68.37  28.32  77.15   70.96  30.02   2.90 100.00  97.10 -0.58
## dis        7 466   3.80   2.11   3.19    3.54   1.91   1.13  12.13  11.00  1.00
## rad        8 466   9.53   8.69   5.00    8.70   1.48   1.00  24.00  23.00  1.01
## tax        9 466 409.50 167.90 334.50  401.51 104.52 187.00 711.00 524.00  0.66
## ptratio   10 466  18.40   2.20  18.90   18.60   1.93  12.60  22.00   9.40 -0.75
## lstat     11 466  12.63   7.10  11.35   11.88   7.07   1.73  37.97  36.24  0.91
## medv      12 466  22.59   9.24  21.20   21.63   6.00   5.00  50.00  45.00  1.08
## target    13 466   0.49   0.50   0.00    0.49   0.00   0.00   1.00   1.00  0.03
##         kurtosis   se
## zn          3.81 1.08
## indus      -1.24 0.32
## chas        9.15 0.01
## nox        -0.04 0.01
## rm          1.54 0.03
## age        -1.01 1.31
## dis         0.47 0.10
## rad        -0.86 0.40
## tax        -1.15 7.78
## ptratio    -0.40 0.10
## lstat       0.50 0.33
```

```
## medv        1.37 0.43
## target     -2.00 0.02
```

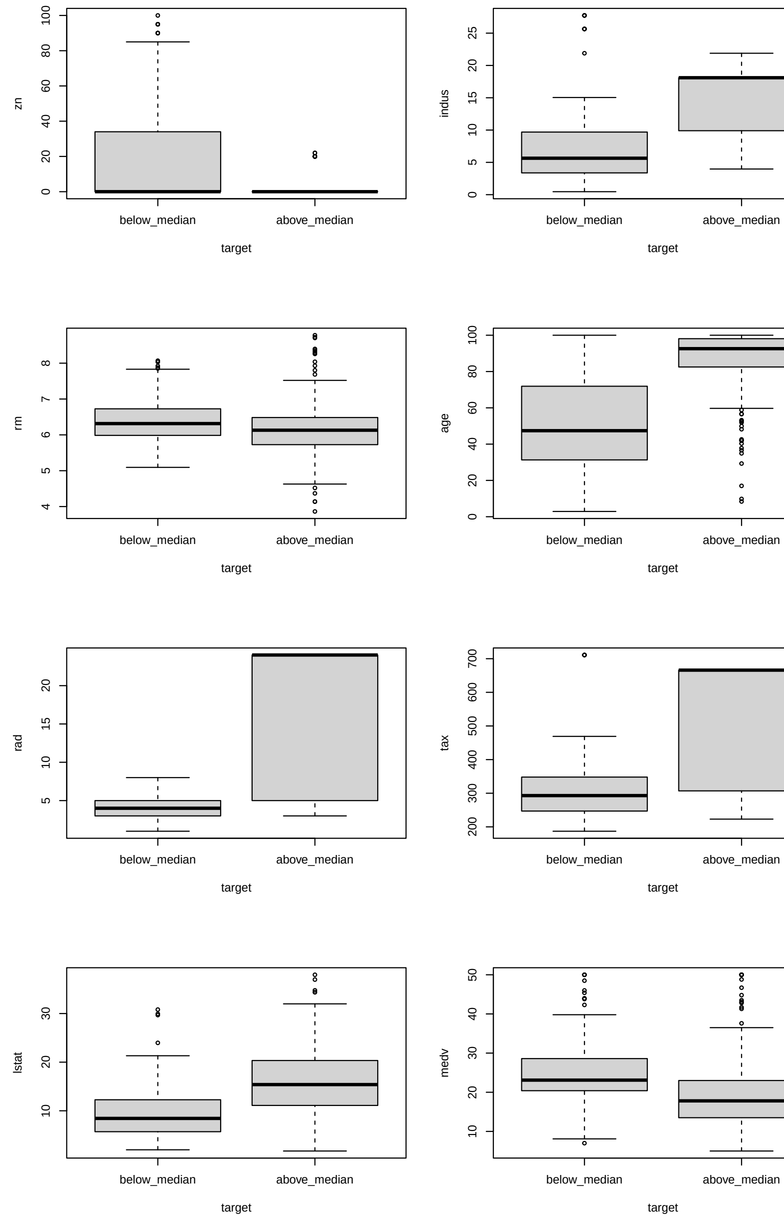
Factor categorical variables

```
# from the training data set; variable: target
dftrain$target <- factor(dftrain$target, levels = c(0, 1))
levels(dftrain$target) <- list(below_median = 0, above_median = 1)

# from the training data set; variable: chas
dftrain$chas <- factor(dftrain$chas)
levels(dftrain$chas) <- list(not_on_charles = 0, on_charles = 1)

# from the evaluation data set; variable: chas
dfeval$chas <- factor(dfeval$chas)
levels(dfeval$chas) <- list(not_on_charles = 0, on_charles = 1)
```

The plot matrix below consists of scatter plots corresponding to each data frame



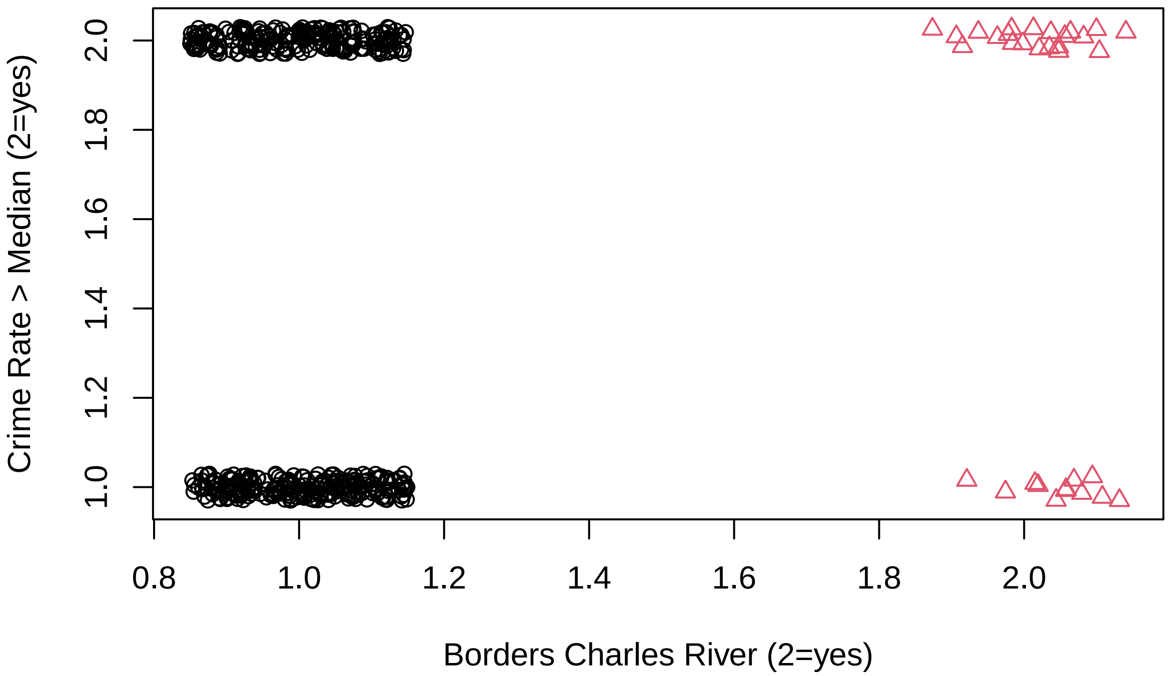
These boxplots below are show plenty of variables in our training data set with outliers. We also notice that



variables `rad` and `tax` have a higher median for crime rate.

We created a contingency table to show the distribution of a variables `target` and `chas`. By using a jitter plot we are trying to visualize the relationship between these two variables.

```
##                 chas
## target           not_on_charles on_charles
##   below_median              225         12
##   above_median              208         21
```



Relationship of median crime rate to the following predictor variables:

Predictor	Definition	Relationship to Median Crime Rate
zn	Proportion of residential land zoned for large lots (over 25000 square feet)	negative
indus	Proportion of non-retail business acres per suburb	positive
chas	Dummy var. for whether the suburb borders the Charles River	unclear
nox	Nitrogen oxides concentration	positive
rm	Average number of rooms per dwelling	unclear
age	Proportion of owner-occupied units built prior to 1940	positive
dis	Weighted mean of distances to five Boston employment centers	negative
rad	Index of accessibility to radial highways	positive
tax	Full-value property-tax rate per $10,000	positive
ptratio	Pupil-teacher ratio by town	positive
lstat	Lower status of the population (percent)	positive
medv	Median value of owner-occupied homes in $1000s	negative

As indicated in the table, several predictors exhibit exhibite an inverse relationship with median crime rate. Based on the zn and medv variables, larger lot sizes and higher median home values correspond to a drop in crime rate, which is expected since larger lots and higher home values typically indicate higher economic status and, hence, lower crime. The same is true for the dis variable, which indicates that the farther a neighborhood is away from a major employment center, the lower the crime rate; this also makes sense, given that employment centers are often located in denser, more urban settings, which typically have higher rates of crime.

For the most part, variables exhibiting positive relationships with median crime rate also make intuitive sense. It would follow that neighborhoods having higher rates of industry (and, therefore, higher concentrations of pollutants like nitrogen oxides–nox–in the air) would also have higher crime rates. Likewise, neighborhoods with older homes (indicated by the age variable) located near radial highways (rad variable) and with a high pupil-to-teacher ratio (ptratio variable) could also be interpreted to have higher rates of crime.

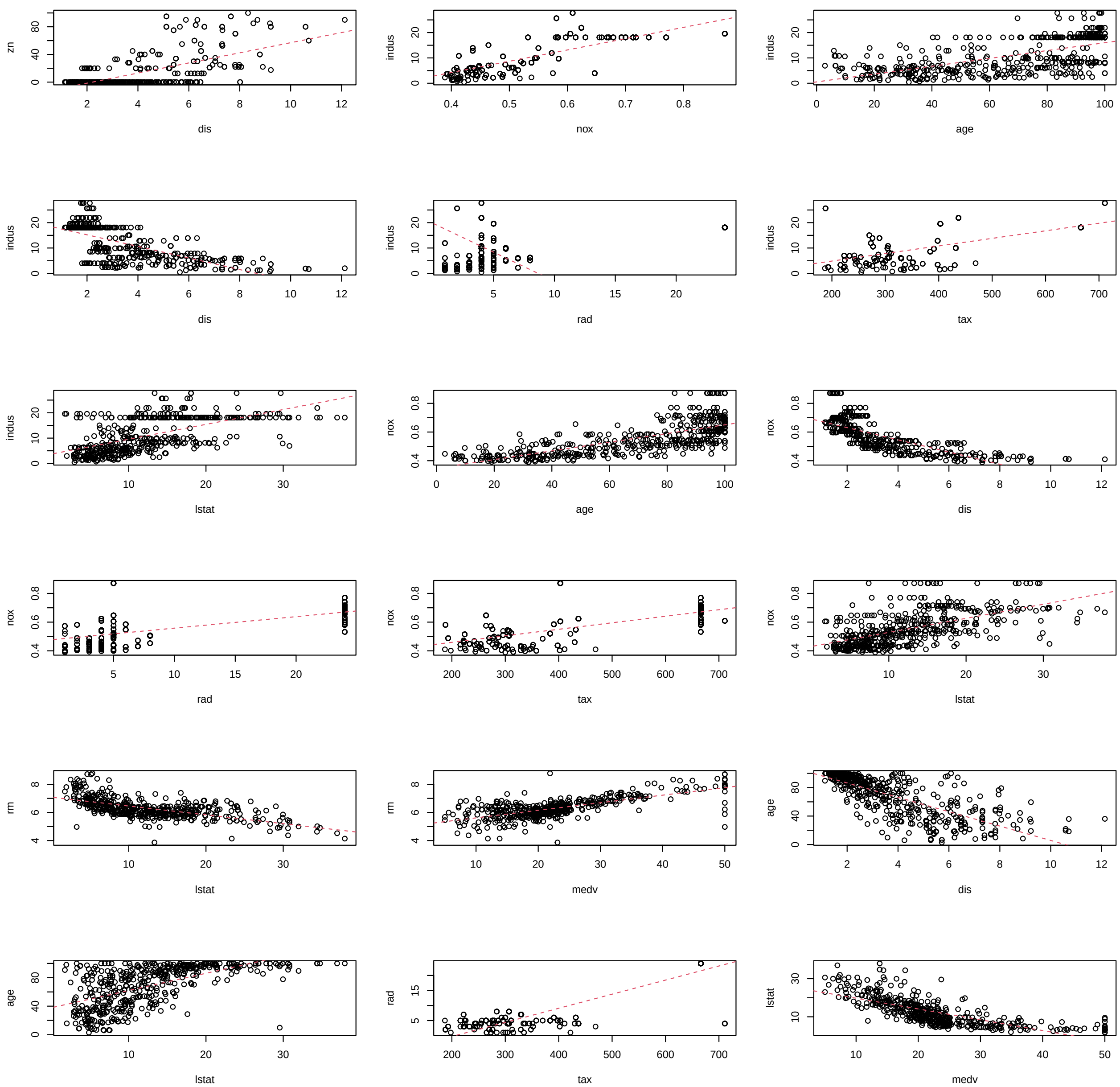
Two variables didn't exhibit a clear relationships to crime rate: whether the neighborhood borders the Charles River (chas) and the average number of rooms per dwelling (rm). In addition, while the the lstat predictor exhibitied a positive relationship with crime rate, the description of the variable ("lower status of the population") didn't clearly state what the data values represent.

Now we'll look for any significant relationships among predictor variables. We considered correlation values above 0.6 to be significant.

Let's explore colinearity of predictor variables with the help of a correlation matrix:

```
## [1] "Correlation matrix (numerical variables):"
```

```
##            zn indus   nox    rm   age   dis   rad   tax ptratio lstat  medv
## zn       1.00 -0.54 -0.52  0.32 -0.57  0.66 -0.32 -0.32   -0.39 -0.43  0.38
## indus   -0.54  1.00  0.76 -0.39  0.64 -0.70  0.60  0.73    0.39  0.61 -0.50
## nox     -0.52  0.76  1.00 -0.30  0.74 -0.77  0.60  0.65    0.18  0.60 -0.43
## rm       0.32 -0.39 -0.30  1.00 -0.23  0.20 -0.21 -0.30   -0.36 -0.63  0.71
## age     -0.57  0.64  0.74 -0.23  1.00 -0.75  0.46  0.51    0.26  0.61 -0.38
## dis      0.66 -0.70 -0.77  0.20 -0.75  1.00 -0.49 -0.53   -0.23 -0.51  0.26
## rad     -0.32  0.60  0.60 -0.21  0.46 -0.49  1.00  0.91    0.47  0.50 -0.40
## tax     -0.32  0.73  0.65 -0.30  0.51 -0.53  0.91  1.00    0.47  0.56 -0.49
## ptratio -0.39  0.39  0.18 -0.36  0.26 -0.23  0.47  0.47    1.00  0.38 -0.52
## lstat   -0.43  0.61  0.60 -0.63  0.61 -0.51  0.50  0.56    0.38  1.00 -0.74
## medv     0.38 -0.50 -0.43  0.71 -0.38  0.26 -0.40 -0.49   -0.52 -0.74  1.00
```



As shown in the graphs above, a number of significant correlations exist. Some of the stronger relationships are discussed here. First, the proportion of area zoned for large lots (zn) has a positive relationship with the distance to employment centers (dis), since it is more difficult to locate large lots close to the city center. A strong positive correlation exists between indus and nox, which is intuitively obvious. Likewise, tax rates in industrial areas are likely to be higher, as shown by the strong positive correlation of 0.73. Another strong correlation that makes obvious intuitive sense is that between median home values (medv) and the average number of rooms per dwelling (rm). The strongest positive correlation (0.91) exists between tax rate (tax) and the index of accessibility to radial highways (rad), which also corresponds to the fact that industrial areas are typically close to radial highways and also exhibit higher tax rates. The strongest negative correlation (-0.77) exists between nox and dis, indicating that the farther away from employment centers (and, hence, industrial areas), the lower the concentration of nitrogen oxide pollutants. Almost equally strong (-0.75) is the correlation between the age of dwellings (age) and the distance from employment centers (dis), indicating that the farther from urban centers, the newer the houses, which makes intuitive sense.

Data Preparation:

There are no missing values for our data sets

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

```
##      zn   indus    chas     nox      rm     age     dis     rad     tax ptratio
##       0       0       0       0       0       0       0       0       0       0
##   lstat    medv
##       0       0
```

The rad predictor is a categorical value and has some unknown meaning for values 1-8, 24. We need to introduce dummy variables rad1, rad2, etc to indicate if the neighborhood is in which category. We will exclude rad24 since we only need N-1 variables to represent each value.

```
# cleaning train data
clean_df <- function(df) {
    df$rad_1 <- ifelse(df$rad == 1, 1, 0)
    df$rad_2 <- ifelse(df$rad == 2, 1, 0)
    df$rad_3 <- ifelse(df$rad == 3, 1, 0)
    df$rad_4 <- ifelse(df$rad == 4, 1, 0)
    df$rad_5 <- ifelse(df$rad == 5, 1, 0)
    df$rad_6 <- ifelse(df$rad == 6, 1, 0)
    df$rad_7 <- ifelse(df$rad == 7, 1, 0)
    df$rad_8 <- ifelse(df$rad == 8, 1, 0)
    df$rad <- NULL
    return(df)
}

dftrain_clean <- clean_df(dftrain)
dftrain_clean <- dftrain_clean %>%
    select(target, everything())
dfeval_clean <- clean_df(dfeval)
```

Model Building:

Logistic Regression: Ippolito

```
# Maximal model for backward elimination
lmod.max <- glm(target ~ ., family = binomial(), data = dftrain)
summary(lmod.max)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = dftrain)
##
## 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
## chason_charles   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
```

```r
# Backward elimination
lmod.back <- step(lmod.max, data = dftrain, direction = "backward")
```

```
## 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
```

```
summary(lmod.back)
```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv, family = binomial(), data = dftrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## zn           -0.068648   0.032019  -2.144  0.03203 *
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## age           0.032950   0.010951   3.009  0.00262 **
## dis           0.654896   0.214050   3.060  0.00222 **
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## medv          0.110472   0.035445   3.117  0.00183 **
## ---
## 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: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

```
# Minimal model for forward elimination
lmod.min <- glm(target ~ 1, family = binomial(), data = dftrain)
summary(lmod.min)
```

```
##
## Call:
## glm(formula = target ~ 1, family = binomial(), data = dftrain)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.163  -1.163  -1.163   1.192   1.192
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.03434    0.09266  -0.371    0.711
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 645.88  on 465  degrees of freedom
## AIC: 647.88
##
## Number of Fisher Scoring iterations: 3
```

```r
lmod.max.form <- formula(lmod.max)

# Forward elimination
lmod.fwd <- step(lmod.min, data = dftrain, direction = "forward",
    scope = lmod.max.form)
```

```
## Start:  AIC=647.88
## target ~ 1
##
##           Df Deviance    AIC
## + nox      1   292.01 296.01
## + rad      1   404.16 408.16
## + dis      1   409.50 413.50
## + age      1   424.75 428.75
## + tax      1   442.38 446.38
## + indus    1   453.23 457.23
## + zn       1   518.46 522.46
## + lstat    1   528.01 532.01
## + medv     1   609.62 613.62
## + ptratio  1   615.64 619.64
## + rm       1   634.82 638.82
## + chas     1   642.86 646.86
## <none>         645.88 647.88
##
## Step:  AIC=296.01
## target ~ nox
##
##           Df Deviance    AIC
## + rad      1   239.51 245.51
## + rm       1   284.63 290.63
## + medv     1   285.86 291.86
## + indus    1   288.11 294.11
## + zn       1   288.29 294.29
## + tax      1   288.40 294.40
## + chas     1   288.47 294.47
## <none>         292.01 296.01
## + ptratio  1   290.13 296.13
```

```
## + age      1   290.62 296.62
## + dis      1   290.91 296.91
## + lstat    1   291.93 297.93
##
## Step:  AIC=245.51
## target ~ nox + rad
##
##           Df Deviance    AIC
## + tax      1   224.47 232.47
## + indus    1   233.09 241.09
## + zn       1   235.19 243.19
## + rm       1   236.60 244.60
## + age      1   236.76 244.76
## + medv     1   236.86 244.86
## + ptratio  1   237.33 245.33
## <none>         239.51 245.51
## + chas     1   237.64 245.64
## + dis      1   237.96 245.96
## + lstat    1   239.47 247.47
##
## Step:  AIC=232.47
## target ~ nox + rad + tax
##
##           Df Deviance    AIC
## + ptratio  1   218.70 228.70
## + zn       1   219.94 229.94
## + age      1   220.44 230.44
## <none>         224.47 232.47
## + dis      1   223.30 233.30
## + indus    1   223.40 233.40
## + chas     1   223.63 233.63
## + lstat    1   223.71 233.71
## + rm       1   223.75 233.75
## + medv     1   224.27 234.27
##
## Step:  AIC=228.7
## target ~ nox + rad + tax + ptratio
##
##         Df Deviance    AIC
## + age    1   214.46 226.46
## + medv   1   215.23 227.23
## + rm     1   216.12 228.12
## + zn     1   216.32 228.32
## <none>       218.70 228.70
## + chas   1   216.81 228.81
## + dis    1   217.79 229.79
## + indus  1   217.82 229.82
## + lstat  1   218.57 230.57
##
## Step:  AIC=226.46
## target ~ nox + rad + tax + ptratio + age
##
##         Df Deviance    AIC
## + medv   1   209.55 223.55
```

```
## + rm     1   212.31 226.31
## + dis    1   212.40 226.40
## <none>       214.46 226.46
## + zn     1   212.67 226.67
## + chas   1   213.24 227.24
## + indus  1   213.38 227.38
## + lstat  1   214.35 228.35
##
## Step:  AIC=223.55
## target ~ nox + rad + tax + ptratio + age + medv
##
##         Df Deviance    AIC
## + dis    1   203.45 219.45
## <none>       209.55 223.55
## + zn     1   207.64 223.64
## + lstat  1   208.07 224.07
## + chas   1   208.33 224.33
## + indus  1   208.58 224.58
## + rm     1   208.79 224.79
##
## Step:  AIC=219.45
## target ~ nox + rad + tax + ptratio + age + medv + dis
##
##         Df Deviance    AIC
## + zn     1   197.32 215.32
## + chas   1   201.29 219.29
## + rm     1   201.35 219.35
## <none>       203.45 219.45
## + lstat  1   202.05 220.05
## + indus  1   202.23 220.23
##
## Step:  AIC=215.32
## target ~ nox + rad + tax + ptratio + age + medv + dis + zn
##
##         Df Deviance    AIC
## <none>       197.32 215.32
## + lstat  1   195.51 215.51
## + rm     1   195.75 215.75
## + chas   1   195.97 215.97
## + indus  1   196.33 216.33
```

```
summary(lmod.fwd)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + age + medv +
##     dis + zn, family = binomial(), data = dftrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## age           0.032950   0.010951   3.009  0.00262 **
## medv          0.110472   0.035445   3.117  0.00183 **
## dis           0.654896   0.214050   3.060  0.00222 **
## zn           -0.068648   0.032019  -2.144  0.03203 *
## ---
## 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: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

Linear Model

Logistic Regression: Butler

```
dftrain_clean_dummy <- dftrain_clean %>%
    mutate(target = as.numeric(target == "above_median"))
olsreg <- lm(data = dftrain_clean_dummy, formula = target ~ .)
summary(olsreg)
```

You can't calculate residuals for a factor so I created a dummy target variable for this model

```
##
## Call:
## lm(formula = target ~ ., data = dftrain_clean_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70893 -0.17857 -0.03795  0.17333  1.02572
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.5216403  0.3719659  -1.402 0.161495
## zn             -0.0009312  0.0008753  -1.064 0.287955
## indus          -0.0022092  0.0039915  -0.553 0.580214
## chason_charles -0.0733915  0.0530829  -1.383 0.167485
## nox             2.1406483  0.2411727   8.876  < 2e-16 ***
## rm              0.0059837  0.0284320   0.210 0.833408
## age             0.0030372  0.0008144   3.729 0.000217 ***
## dis             0.0024358  0.0129550   0.188 0.850948
## tax            -0.0001335  0.0002407  -0.554 0.579618
```

```
## ptratio        -0.0135287  0.0092241  -1.467 0.143171
## lstat           0.0028229  0.0035179   0.802 0.422730
## medv            0.0078729  0.0027222   2.892 0.004014 **
## rad_1          -0.5470800  0.1083242  -5.050 6.44e-07 ***
## rad_2          -0.6743564  0.1112379  -6.062 2.86e-09 ***
## rad_3          -0.5584557  0.1014553  -5.504 6.26e-08 ***
## rad_4          -0.2139619  0.0812526  -2.633 0.008750 **
## rad_5          -0.4966807  0.0825141  -6.019 3.66e-09 ***
## rad_6          -0.6024081  0.0887388  -6.789 3.62e-11 ***
## rad_7          -0.3756072  0.1082244  -3.471 0.000570 ***
## rad_8           0.0565900  0.1017049   0.556 0.578207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2777 on 446 degrees of freedom
## Multiple R-squared:  0.7046, Adjusted R-squared:  0.692
## F-statistic: 55.99 on 19 and 446 DF,  p-value: < 2.2e-16
```

Logit Model

```r
logit <- glm(data = dftrain_clean, formula = target ~ ., family = binomial(link = "logit"))
summary(logit)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = dftrain_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5265  -0.0409   0.0000   0.0001   4.3848
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.123e+01  1.944e+03  -0.006   0.9954
## zn             -1.609e-01  6.574e-02  -2.447   0.0144 *
## indus          -1.562e-01  1.166e-01  -1.340   0.1802
## chason_charles -2.603e-01  9.626e-01  -0.270   0.7869
## nox             6.863e+01  1.362e+01   5.038 4.71e-07 ***
## rm             -1.225e+00  1.010e+00  -1.213   0.2250
## age             1.871e-02  1.569e-02   1.193   0.2330
## dis             5.351e-01  2.671e-01   2.003   0.0452 *
## tax            -9.491e-03  5.442e-03  -1.744   0.0811 .
## ptratio         4.824e-02  2.040e-01   0.236   0.8131
## lstat           6.778e-02  6.441e-02   1.052   0.2927
## medv            2.195e-01  9.964e-02   2.203   0.0276 *
## rad_1          -4.404e+01  5.457e+03  -0.008   0.9936
## rad_2          -4.449e+01  5.328e+03  -0.008   0.9933
## rad_3          -2.620e+01  1.944e+03  -0.013   0.9892
## rad_4          -2.182e+01  1.944e+03  -0.011   0.9910
## rad_5          -2.454e+01  1.944e+03  -0.013   0.9899
## rad_6          -2.666e+01  1.944e+03  -0.014   0.9891
```

```
## rad_7          -1.704e+01  1.944e+03  -0.009   0.9930
## rad_8          -1.840e+01  1.944e+03  -0.009   0.9924
## ---
## 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: 116.98  on 446  degrees of freedom
## AIC: 156.98
##
## Number of Fisher Scoring iterations: 20
```

Logit Model with Backward Elimination

```r
# Backward elimination
lmod.back <- step(logit, data = dftrain_clean, direction = "backward")
```

```
## Start:  AIC=156.98
## target ~ zn + indus + chas + nox + rm + age + dis + tax + ptratio +
##     lstat + medv + rad_1 + rad_2 + rad_3 + rad_4 + rad_5 + rad_6 +
##     rad_7 + rad_8
##
##           Df Deviance    AIC
## - ptratio  1   117.04 155.04
## - chas     1   117.06 155.06
## - lstat    1   118.07 156.07
## - age      1   118.44 156.44
## - rad_7    1   118.47 156.47
## - rm       1   118.50 156.50
## - indus    1   118.82 156.82
## <none>         116.98 156.98
## - rad_8    1   119.91 157.91
## - tax      1   120.42 158.42
## - dis      1   121.06 159.06
## - medv     1   122.98 160.98
## - zn       1   125.74 163.74
## - rad_4    1   137.55 175.55
## - rad_2    1   141.93 179.93
## - rad_3    1   149.43 187.43
## - rad_1    1   156.00 194.00
## - rad_5    1   156.41 194.41
## - rad_6    1   177.89 215.89
## - nox      1   185.39 223.39
##
## Step:  AIC=155.04
## target ~ zn + indus + chas + nox + rm + age + dis + tax + lstat +
##     medv + rad_1 + rad_2 + rad_3 + rad_4 + rad_5 + rad_6 + rad_7 +
##     rad_8
##
##         Df Deviance    AIC
## - chas   1   117.11 153.11
```

```
## - lstat  1   118.14 154.15
## - age    1   118.46 154.46
## - rad_7  1   118.47 154.47
## - rm     1   118.53 154.53
## <none>       117.04 155.04
## - indus  1   119.35 155.35
## - rad_8  1   119.91 155.91
## - tax    1   120.42 156.42
## - dis    1   121.17 157.17
## - medv   1   124.02 160.02
## - zn     1   127.07 163.07
## - rad_4  1   137.67 173.67
## - rad_2  1   142.58 178.58
## - rad_3  1   149.60 185.60
## - rad_1  1   156.42 192.42
## - rad_5  1   158.34 194.34
## - rad_6  1   179.73 215.73
## - nox    1   187.89 223.89
##
## Step:  AIC=153.11
## target ~ zn + indus + nox + rm + age + dis + tax + lstat + medv +
##     rad_1 + rad_2 + rad_3 + rad_4 + rad_5 + rad_6 + rad_7 + rad_8
##
##         Df Deviance    AIC
## - lstat  1   118.17 152.17
## - age    1   118.46 152.46
## - rad_7  1   118.50 152.50
## - rm     1   118.54 152.54
## <none>       117.11 153.11
## - rad_8  1   119.94 153.94
## - indus  1   120.17 154.17
## - tax    1   120.66 154.66
## - dis    1   121.41 155.41
## - medv   1   124.07 158.07
## - zn     1   127.10 161.10
## - rad_4  1   138.03 172.03
## - rad_2  1   144.31 178.31
## - rad_3  1   152.05 186.05
## - rad_1  1   156.55 190.55
## - rad_5  1   159.20 193.20
## - rad_6  1   180.63 214.63
## - nox    1   190.43 224.43
##
## Step:  AIC=152.17
## target ~ zn + indus + nox + rm + age + dis + tax + medv + rad_1 +
##     rad_2 + rad_3 + rad_4 + rad_5 + rad_6 + rad_7 + rad_8
##
##         Df Deviance    AIC
## - rad_7  1   119.97 151.97
## <none>       118.17 152.17
## - age    1   120.74 152.74
## - indus  1   120.93 152.93
## - rm     1   121.05 153.05
## - rad_8  1   121.62 153.62
```

```
## - tax    1   121.73 153.73
## - dis    1   122.35 154.35
## - medv   1   125.18 157.18
## - zn     1   127.58 159.58
## - rad_4  1   138.44 170.44
## - rad_2  1   145.60 177.60
## - rad_3  1   152.90 184.90
## - rad_1  1   159.16 191.16
## - rad_5  1   160.76 192.76
## - rad_6  1   180.95 212.95
## - nox    1   191.60 223.60
##
## Step:  AIC=151.97
## target ~ zn + indus + nox + rm + age + dis + tax + medv + rad_1 +
##     rad_2 + rad_3 + rad_4 + rad_5 + rad_6 + rad_8
##
##         Df Deviance    AIC
## - rad_8  1   121.75 151.75
## <none>       119.97 151.97
## - tax    1   122.40 152.40
## - age    1   122.45 152.45
## - rm     1   123.24 153.24
## - dis    1   124.24 154.24
## - indus  1   125.03 155.03
## - medv   1   127.81 157.81
## - zn     1   140.31 170.31
## - rad_4  1   141.71 171.71
## - rad_2  1   150.79 180.79
## - rad_3  1   159.80 189.80
## - rad_1  1   164.34 194.34
## - rad_5  1   172.56 202.56
## - rad_6  1   186.82 216.82
## - nox    1   208.97 238.97
##
## Step:  AIC=151.75
## target ~ zn + indus + nox + rm + age + dis + tax + medv + rad_1 +
##     rad_2 + rad_3 + rad_4 + rad_5 + rad_6
##
##         Df Deviance    AIC
## - tax    1   123.70 151.70
## <none>       121.75 151.75
## - age    1   124.18 152.18
## - rm     1   124.86 152.86
## - dis    1   125.30 153.30
## - indus  1   127.23 155.23
## - medv   1   129.59 157.59
## - zn     1   140.72 168.72
## - rad_4  1   142.60 170.60
## - rad_2  1   152.11 180.11
## - rad_1  1   165.96 193.96
## - rad_3  1   165.98 193.98
## - rad_5  1   189.53 217.53
## - rad_6  1   194.20 222.20
## - nox    1   209.71 237.71
```

```
##
## Step:  AIC=151.7
## target ~ zn + indus + nox + rm + age + dis + medv + rad_1 + rad_2 +
##     rad_3 + rad_4 + rad_5 + rad_6
##
##         Df Deviance    AIC
## <none>       123.70 151.70
## - age    1   126.82 152.82
## - rm     1   128.16 154.16
## - dis    1   128.76 154.76
## - medv   1   135.26 161.26
## - zn     1   141.21 167.21
## - rad_4  1   144.35 170.35
## - indus  1   145.93 171.93
## - rad_2  1   162.33 188.33
## - rad_3  1   166.25 192.25
## - rad_1  1   168.22 194.22
## - rad_6  1   194.71 220.71
## - rad_5  1   202.57 228.57
## - nox    1   213.94 239.94
```

```
summary(lmod.back)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + rm + age + dis + medv +
##     rad_1 + rad_2 + rad_3 + rad_4 + rad_5 + rad_6, family = binomial(link = "logit"),
##     data = dftrain_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2642  -0.0406   0.0000   0.0417   4.6501
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -27.04848    6.14058  -4.405 1.06e-05 ***
## zn            -0.18143    0.05583  -3.249  0.00116 **
## indus         -0.28923    0.07065  -4.094 4.25e-05 ***
## nox           69.44207   12.13357   5.723 1.05e-08 ***
## rm            -1.69944    0.82175  -2.068  0.03863 *
## age            0.02380    0.01364   1.745  0.08092 .
## dis            0.57815    0.26534   2.179  0.02934 *
## medv           0.24437    0.07962   3.069  0.00215 **
## rad_1        -24.31086 1917.60218  -0.013  0.98988
## rad_2        -22.64510 2049.05400  -0.011  0.99118
## rad_3         -9.10961    2.15021  -4.237 2.27e-05 ***
## rad_4         -4.43125    1.42259  -3.115  0.00184 **
## rad_5         -7.36393    1.50464  -4.894 9.87e-07 ***
## rad_6        -10.00046    2.03065  -4.925 8.45e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 123.70  on 452  degrees of freedom
## AIC: 151.7
##
## Number of Fisher Scoring iterations: 18
```

Logit Minimal Model with forward elimination

```r
# Minimal model for forward elimination
lmod.min <- glm(target ~ 1, family = binomial(), data = dftrain_clean)
summary(lmod.min)
```

```
##
## Call:
## glm(formula = target ~ 1, family = binomial(), data = dftrain_clean)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.163  -1.163  -1.163   1.192   1.192
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.03434    0.09266  -0.371    0.711
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 645.88  on 465  degrees of freedom
## AIC: 647.88
##
## Number of Fisher Scoring iterations: 3
```

```r
# Forward elimination
lmod.fwd <- step(lmod.min, data = dftrain_clean, direction = "forward",
    scope = formula(logit))
```

```
## Start:  AIC=647.88
## target ~ 1
##
##           Df Deviance    AIC
## + nox      1   292.01 296.01
## + dis      1   409.50 413.50
## + age      1   424.75 428.75
## + tax      1   442.38 446.38
## + indus    1   453.23 457.23
## + zn       1   518.46 522.46
## + lstat    1   528.01 532.01
## + rad_3    1   603.67 607.67
## + medv     1   609.62 613.62
## + ptratio  1   615.64 619.64
## + rad_2    1   617.96 621.96
```

```
## + rad_1     1   622.27 626.27
## + rad_6     1   624.91 628.91
## + rm        1   634.82 638.82
## + rad_7     1   636.98 640.98
## + rad_8     1   637.41 641.41
## + rad_5     1   640.49 644.49
## + chas      1   642.86 646.86
## + rad_4     1   643.69 647.69
## <none>          645.88 647.88
##
## Step:  AIC=296.01
## target ~ nox
##
##           Df Deviance    AIC
## + rad_8    1   254.85 260.85
## + rad_6    1   268.66 274.66
## + rad_2    1   274.39 280.39
## + rad_1    1   278.70 284.70
## + rad_5    1   279.56 285.56
## + rad_4    1   284.30 290.30
## + rm       1   284.63 290.63
## + medv     1   285.86 291.86
## + indus    1   288.11 294.11
## + zn       1   288.29 294.29
## + tax      1   288.40 294.40
## + chas     1   288.47 294.47
## + rad_3    1   289.72 295.72
## <none>         292.01 296.01
## + ptratio  1   290.13 296.13
## + rad_7    1   290.53 296.53
## + age      1   290.62 296.62
## + dis      1   290.91 296.91
## + lstat    1   291.93 297.93
##
## Step:  AIC=260.85
## target ~ nox + rad_8
##
##           Df Deviance    AIC
## + rad_6    1   234.80 242.80
## + rad_4    1   235.47 243.47
## + rad_2    1   239.16 247.16
## + rad_1    1   243.22 251.22
## + rad_5    1   249.11 257.11
## + ptratio  1   250.38 258.38
## + tax      1   250.41 258.41
## + rad_7    1   251.21 259.21
## + dis      1   252.33 260.33
## + zn       1   252.34 260.33
## + indus    1   252.78 260.78
## <none>         254.85 260.85
## + lstat    1   254.24 262.24
## + medv     1   254.32 262.32
## + rad_3    1   254.38 262.38
## + rm       1   254.45 262.45
```

```
## + chas     1   254.46 262.46
## + age      1   254.51 262.51
##
## Step:  AIC=242.8
## target ~ nox + rad_8 + rad_6
##
##           Df Deviance    AIC
## + rad_2    1   215.90 225.90
## + rad_1    1   220.72 230.72
## + rad_4    1   221.90 231.90
## + rad_5    1   223.12 233.12
## + indus    1   227.51 237.51
## + tax      1   229.81 239.81
## + ptratio  1   231.20 241.20
## + rad_7    1   231.22 241.22
## + zn       1   231.72 241.72
## <none>         234.80 242.80
## + dis      1   233.69 243.69
## + lstat    1   233.81 243.81
## + rad_3    1   234.18 244.18
## + medv     1   234.48 244.48
## + chas     1   234.69 244.69
## + rm       1   234.79 244.79
## + age      1   234.79 244.79
##
## Step:  AIC=225.9
## target ~ nox + rad_8 + rad_6 + rad_2
##
##           Df Deviance    AIC
## + rad_5    1   198.67 210.67
## + rad_1    1   199.77 211.77
## + rad_4    1   206.67 218.67
## + rad_7    1   212.18 224.18
## + ptratio  1   212.31 224.31
## + zn       1   212.52 224.52
## <none>         215.90 225.90
## + lstat    1   214.53 226.53
## + indus    1   215.00 227.00
## + tax      1   215.18 227.18
## + rad_3    1   215.22 227.22
## + medv     1   215.48 227.48
## + dis      1   215.62 227.62
## + chas     1   215.88 227.88
## + age      1   215.89 227.89
## + rm       1   215.90 227.90
##
## Step:  AIC=210.67
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5
##
##           Df Deviance    AIC
## + rad_1    1   175.59 189.59
## + indus    1   195.32 209.32
## + rad_3    1   195.99 209.99
## + medv     1   196.14 210.14
```

```
## + rad_7     1   196.56 210.56
## <none>          198.67 210.67
## + zn        1   196.69 210.69
## + rad_4     1   197.98 211.98
## + rm        1   198.14 212.14
## + dis       1   198.27 212.27
## + chas      1   198.35 212.35
## + lstat     1   198.41 212.41
## + tax       1   198.62 212.62
## + ptratio  1   198.65 212.65
## + age       1   198.66 212.66
##
## Step:  AIC=189.59
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1
##
##           Df Deviance    AIC
## + indus     1   167.39 183.39
## + rad_3     1   172.00 188.00
## + medv      1   173.55 189.55
## <none>          175.59 189.59
## + rad_7     1   173.63 189.63
## + tax       1   173.91 189.91
## + rm        1   174.20 190.20
## + zn        1   174.46 190.46
## + rad_4     1   174.99 190.99
## + dis       1   175.05 191.05
## + lstat     1   175.34 191.34
## + chas      1   175.46 191.46
## + ptratio  1   175.51 191.51
## + age       1   175.56 191.56
##
## Step:  AIC=183.39
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus
##
##           Df Deviance    AIC
## + rad_3     1   159.81 177.81
## <none>          167.39 183.39
## + rad_7     1   165.68 183.68
## + rad_4     1   166.18 184.18
## + zn        1   166.19 184.19
## + chas      1   166.35 184.35
## + medv      1   166.43 184.43
## + dis       1   166.69 184.69
## + rm        1   166.72 184.72
## + tax       1   166.76 184.76
## + age       1   167.19 185.19
## + lstat     1   167.29 185.29
## + ptratio  1   167.30 185.30
##
## Step:  AIC=177.81
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3
##
##           Df Deviance    AIC
```

```
## + rad_4    1   148.84 168.84
## + medv     1   154.38 174.38
## + zn       1   156.86 176.86
## + rm       1   157.42 177.42
## <none>         159.81 177.81
## + chas     1   158.65 178.65
## + lstat    1   159.01 179.01
## + rad_7    1   159.43 179.43
## + ptratio  1   159.59 179.59
## + tax      1   159.75 179.75
## + dis      1   159.81 179.81
## + age      1   159.81 179.81
##
## Step:  AIC=168.85
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3 + rad_4
##
##           Df Deviance    AIC
## + zn       1   137.67 159.67
## + rad_7    1   143.52 165.52
## + tax      1   145.42 167.42
## + chas     1   145.82 167.82
## + medv     1   145.85 167.85
## <none>         148.84 168.84
## + rm       1   148.51 170.51
## + ptratio  1   148.56 170.56
## + dis      1   148.73 170.73
## + lstat    1   148.82 170.82
## + age      1   148.84 170.84
##
## Step:  AIC=159.67
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3 + rad_4 + zn
##
##           Df Deviance    AIC
## + tax      1   129.89 153.89
## + medv     1   131.67 155.67
## + ptratio  1   134.68 158.68
## + chas     1   135.07 159.07
## <none>         137.67 159.67
## + dis      1   136.16 160.16
## + rm       1   136.33 160.33
## + rad_7    1   137.13 161.13
## + lstat    1   137.66 161.66
## + age      1   137.67 161.67
##
## Step:  AIC=153.89
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3 + rad_4 + zn + tax
##
##           Df Deviance    AIC
## + medv     1   126.39 152.39
## + rad_7    1   127.23 153.23
## <none>         129.89 153.89
```

```
## + ptratio  1   128.52 154.52
## + rm       1   129.06 155.06
## + dis      1   129.07 155.07
## + chas     1   129.67 155.67
## + lstat    1   129.74 155.74
## + age      1   129.88 155.88
##
## Step:  AIC=152.39
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3 + rad_4 + zn + tax + medv
##
##           Df Deviance    AIC
## + lstat    1   123.59 151.59
## + rad_7    1   124.34 152.34
## <none>         126.39 152.39
## + dis      1   124.50 152.50
## + rm       1   125.08 153.08
## + age      1   126.03 154.03
## + ptratio  1   126.30 154.30
## + chas     1   126.33 154.33
##
## Step:  AIC=151.59
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3 + rad_4 + zn + tax + medv + lstat
##
##           Df Deviance    AIC
## + dis      1   120.57 150.57
## <none>         123.59 151.59
## + rad_7    1   122.05 152.05
## + rm       1   123.19 153.19
## + age      1   123.53 153.53
## + chas     1   123.56 153.56
## + ptratio  1   123.56 153.56
##
## Step:  AIC=150.57
## target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 + rad_1 + indus +
##     rad_3 + rad_4 + zn + tax + medv + lstat + dis
##
##           Df Deviance    AIC
## <none>         120.57 150.57
## + rad_7    1   119.15 151.15
## + rm       1   119.78 151.78
## + age      1   120.03 152.03
## + ptratio  1   120.36 152.36
## + chas     1   120.55 152.55
```

```
summary(lmod.fwd)
```

```
##
## Call:
## glm(formula = target ~ nox + rad_8 + rad_6 + rad_2 + rad_5 +
##     rad_1 + indus + rad_3 + rad_4 + zn + tax + medv + lstat +
##     dis, family = binomial(), data = dftrain_clean)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7310  -0.0446   0.0000   0.0249   4.4357
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.269e+01  6.572e+00  -4.973 6.58e-07 ***
## nox          7.284e+01  1.205e+01   6.047 1.47e-09 ***
## rad_8       -2.257e+00  2.054e+00  -1.099 0.271886
## rad_6       -1.144e+01  2.151e+00  -5.318 1.05e-07 ***
## rad_2       -2.634e+01  1.870e+03  -0.014 0.988765
## rad_5       -8.768e+00  1.685e+00  -5.203 1.96e-07 ***
## rad_1       -2.665e+01  1.874e+03  -0.014 0.988657
## indus       -1.909e-01  9.350e-02  -2.042 0.041133 *
## rad_3       -9.875e+00  2.141e+00  -4.613 3.96e-06 ***
## rad_4       -6.221e+00  1.661e+00  -3.746 0.000180 ***
## zn          -2.021e-01  5.393e-02  -3.748 0.000178 ***
## tax         -8.233e-03  4.286e-03  -1.921 0.054745 .
## medv         1.300e-01  4.997e-02   2.601 0.009293 **
## lstat        1.176e-01  5.911e-02   1.989 0.046732 *
## dis          4.204e-01  2.439e-01   1.724 0.084742 .
## ---
## 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: 120.57  on 451  degrees of freedom
## AIC: 150.57
##
## Number of Fisher Scoring iterations: 18
```

Select Models:

Here we are evaluating the performance our models to decide which model should we use

```
table(true = dftrain_clean_dummy$target, pred = round(fitted(olsreg)))
```

Linear Model Accuracy

```
##     pred
## true   0   1
##    0 225  12
##    1  19 210
```

Accuracy: $\frac{210+225}{466} = 93\%$

Classification Error Rate: $\frac{12+19}{466} = 7\%$

Precision: $\frac{210}{210+12} = 95\%$

Sensitivity: $\frac{210}{210+19} = 92\%$

Specificity: $\frac{225}{225+12} = 95\%$

F1 Score: $\frac{2*.95*.92}{.95+.92} = 93\%$

```
table(true = dftrain_clean$target, pred = round(fitted(logit)))
```

Logit Model Prediction Accuracy

```
##               pred
## true             0   1
##   below_median 233   4
##   above_median  10 219
```

Accuracy: $\frac{219+233}{466} = 97\%$

Classification Error Rate: $\frac{4+10}{466} = 3\%$

Precision: $\frac{219}{219+4} = 98\%$

Sensitivity: $\frac{219}{219+10} = 96\%$

Specificity: $\frac{233}{233+4} = 98\%$

F1 Score: $\frac{2*.98*.96}{.98+.96} = 97\%$

```
table(true = dftrain_clean$target, pred = round(fitted(lmod.fwd)))
```

Logit Model with Forward Elimination Prediction Accuracy

```
##               pred
## true             0   1
##   below_median 234   3
##   above_median   9 220
```

Accuracy: $\frac{220+234}{466} = 97\%$

Classification Error Rate: $\frac{5+12}{466} = 3\%$

Precision: $\frac{220}{220+3} = 99\%$

Sensitivity: $\frac{220}{220+9} = 96\%$

Specificity: $\frac{234}{234+3} = 98\%$

F1 Score: $\frac{2*.99*.96}{.99+.96} = 97\%$

```
table(true = dftrain_clean$target, pred = round(fitted(lmod.back)))
```

Logit Model with Backward Elimination Prediction Accuracy

```
##                 pred
## true               0   1
##   below_median 232   5
##   above_median  12 217
```

Accuracy: $\frac{217+232}{466} = 96\%$

Classification Error Rate: $\frac{5+12}{466} = 4\%$

Precision: $\frac{217}{217+5} = 98\%$

Sensitivity: $\frac{217}{217+12} = 95\%$

Specificity: $\frac{232}{232+5} = 98\%$

F1 Score: $\frac{2*.98*.95}{.98+.95} = 96\%$

Model AUCs

- Linear Model

```
pred = round(fitted(olsreg))
pROC::auc(dftrain_clean_dummy$target, pred)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Area under the curve: 0.9332
```

- Logit Model

```
pred = round(fitted(logit))
pROC::auc(dftrain_clean$target, pred)
```

```
## Setting levels: control = below_median, case = above_median

## Setting direction: controls < cases

## Area under the curve: 0.9697
```

- Logit Model with Forward Elimination

```r
pred = round(fitted(lmod.fwd))
pROC::auc(dftrain_clean$target, pred)
```

```
## Setting levels: control = below_median, case = above_median

## Setting direction: controls < cases

## Area under the curve: 0.974
```

- Logit Model with Backward Elimination

```r
pred = round(fitted(lmod.back))
pROC::auc(dftrain_clean$target, pred)
```

```
## Setting levels: control = below_median, case = above_median

## Setting direction: controls < cases

## Area under the curve: 0.9633
```

Findings All of our Logistic Models had a better AUC than our Linear Model with Logit with Forward Elimination having the best (0.974)

Now lets make predictions with the evaluation df.

- Linear Model

```r
prediction <- broom::augment(olsreg, newdata = dfeval_clean)
prediction$.fitted
```

```
##  [1]  0.044471432  0.549248991  0.590432182  0.549527352  0.100429012
##  [6]  0.602991759  0.668344513  0.036611979  0.004762499 -0.072870822
## [11]  0.249882596  0.236346164  0.697204745  0.529904005  0.501715088
## [16]  0.470213603  0.694800279  0.993663602  0.016477435 -0.148806180
## [21] -0.251508242  0.272093476  0.511840154  0.103025106  0.054423115
## [26]  0.174900012  0.209310715  1.142008184  1.096606893  0.794790626
## [31]  1.129376086  1.148942218  1.125367437  1.197800888  1.145515096
## [36]  1.072375460  1.096093208  0.927138131  0.234910336  0.338470879
```

- Logit Model

```r
prediction <- broom::augment(logit, newdata = dfeval_clean)
prediction$.fitted
```

```
##  [1] -23.1837846   1.8909140   1.7995517   1.8557650  -3.2553526  -2.0334222
##  [7]  -2.5229875  -7.1842002  -7.7530724 -26.2911541 -19.5120234 -19.5664789
## [13]   3.5376684   1.2190996   0.4821279  -0.7066933   3.2735770   6.1682811
## [19]  -2.2482983 -41.5077624 -40.2884118  -1.1204161   0.8969585  -3.5445734
## [25]  -3.6931577  -0.7112376 -15.5439099  34.5405444  28.6931706  19.9909294
## [31]  30.8838151  31.5823360  30.7894623  31.8744218  31.0194002  28.7850963
## [37]  29.6277567  26.3487159  -1.2933127 -19.4359816
```

- Logit Model with Forward Elimination

```
prediction <- broom::augment(lmod.fwd, newdata = dfeval_clean)
prediction$.fitted
```

```
##  [1] -21.1324127   1.6471631   1.9687105   2.8502070  -3.1267587  -3.3028850
##  [7]  -3.6732098  -6.6556256  -7.2930650 -23.8504338 -17.0961819 -17.1653572
## [13]   3.6153389   0.9210658   0.4594639  -1.1097315   4.0131436   6.5538691
## [19]  -1.5678079 -43.1228507 -41.8272191  -1.0616325   1.0108975  -3.3616476
## [25]  -3.3259980  -0.7694337 -18.6851352  15.7643086  11.6898199   5.7354155
## [31]  18.1336636  17.3945880  17.0257378  17.4618888  17.2745735  15.2985175
## [37]  15.4125285  11.6671505  -1.3291711 -17.4399249
```

- Logit Model with Backward Elimination

```
prediction <- broom::augment(lmod.back, newdata = dfeval_clean)
prediction$.fitted
```

```
##  [1] -18.5749014   2.2043350   1.8065924   0.8982560  -3.0322514  -1.0444374
##  [7]  -1.6745047  -7.1433557  -7.7359050 -21.1945465 -19.0579804 -19.1775642
## [13]   2.8342223   1.2115299   0.3085187  -0.9777271   5.1807736   7.9402710
## [19]  -2.0835523 -40.2270283 -39.3183668  -1.8615699   1.2132010  -3.4206755
## [25]  -3.5992473  -0.9833513 -16.1256587  21.2691328  14.8455059   4.2619536
## [31]  14.2147561  15.9542636  15.0602397  16.5703311  15.1893301  13.0524380
## [37]  14.3065313  11.1485972  -1.2497896 -17.6153601
```

References: