Critical Thinking Group 4 - HW3

Sreejaya, Suman, Vuthy

October 10, 2016

Overview

The purpose of this project is to predict if a neighborhood will be at risk for high crime levels using binary logistic regression models. Below is a short description of the variables in the dataset.

- zn: proportion of residential land zoned for large lots (over 25000 square feet)
- indus: proportion of non-retail business acres per suburb
- chas: a dummy var. for whether the suburb borders the Charles River (1) or not (0)
- nox: nitrogen oxides concentration (parts per 10 million)
- rm: average number of rooms per dwelling
- age: proportion of owner-occupied units built prior to 1940
- dis: weighted mean of distances to five Boston employment centers
- rad: index of accessibility to radial highways
- tax: full-value property-tax rate per $10,000
- ptratio: pupil-teacher ratio by town
- black: 1000 $(B_k - 0.63)^2$ where Bk is the proportion of blacks by town
- lstat: lower status of the population (percent)
- medv: median value of owner-occupied homes in $1000s
- target: whether the crime rate is above the median crime rate (1) or not (0) (response variable)

Dataset
Crime - Training data
Crime - Evaluation Data

Data Exploration

The dataset contains 466 observations and 14 variables. The response variable is the **target** variable. Below is a glimpse of the data. A quick look indicates that chas and target might be classification variables.

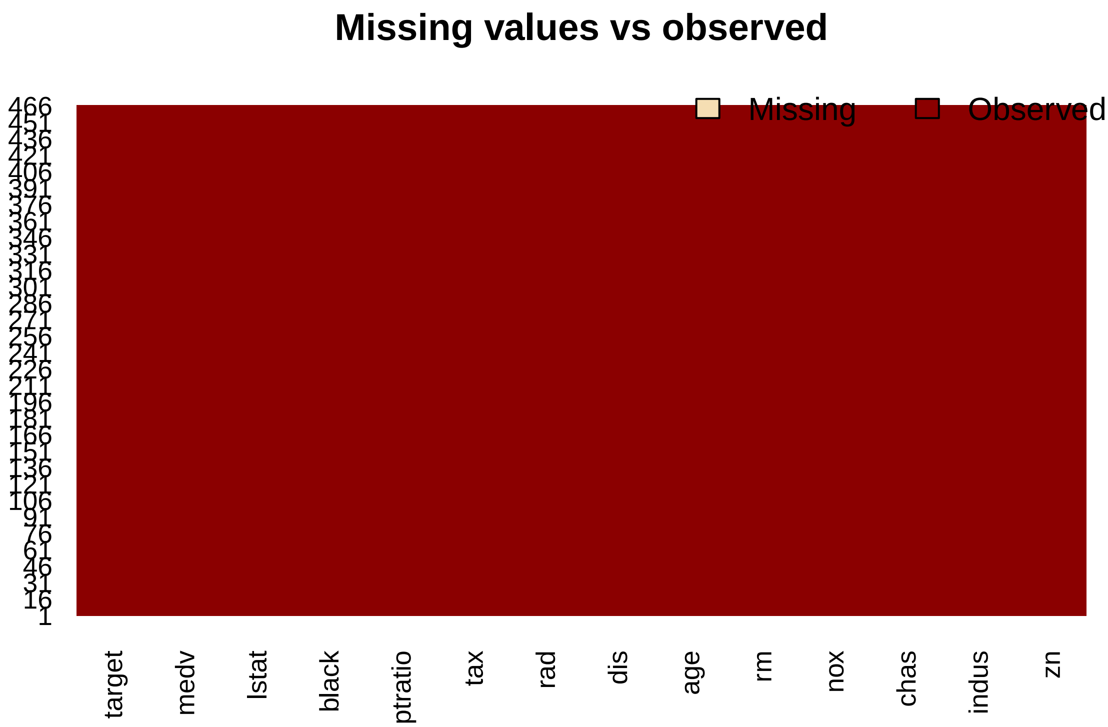
```
## Observations: 466
## Variables: 14
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 10...
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5...
## $ chas    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ nox     <dbl> 0.605, 0.871, 0.740, 0.428, 0.488, 0.520, 0.693, 0.693...
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519...
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38....
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896...
## $ rad     <int> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5,...
## $ tax     <int> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330,...
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, ...
## $ black   <dbl> 369.30, 396.90, 386.73, 374.71, 394.12, 395.58, 396.90...
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5....
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20...
## $ target  <int> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, ...
```

Taking a closer look at the data with summary statistics, we can see that two values (chas, target) should be converted to factors.

```
##        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         black            lstat
##  Min.   :187.0   Min.   :12.6   Min.   :  0.32   Min.   : 1.730
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.:375.61   1st Qu.: 7.043
##  Median :334.5   Median :18.9   Median :391.34   Median :11.350
##  Mean   :409.5   Mean   :18.4   Mean   :357.12   Mean   :12.631
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:396.24   3rd Qu.:16.930
##  Max.   :711.0   Max.   :22.0   Max.   :396.90   Max.   :37.970
##       medv          target
##  Min.   : 5.00   Min.   :0.0000
##  1st Qu.:17.02   1st Qu.:0.0000
##  Median :21.20   Median :0.0000
##  Mean   :22.59   Mean   :0.4914
##  3rd Qu.:25.00   3rd Qu.:1.0000
##  Max.   :50.00   Max.   :1.0000
```

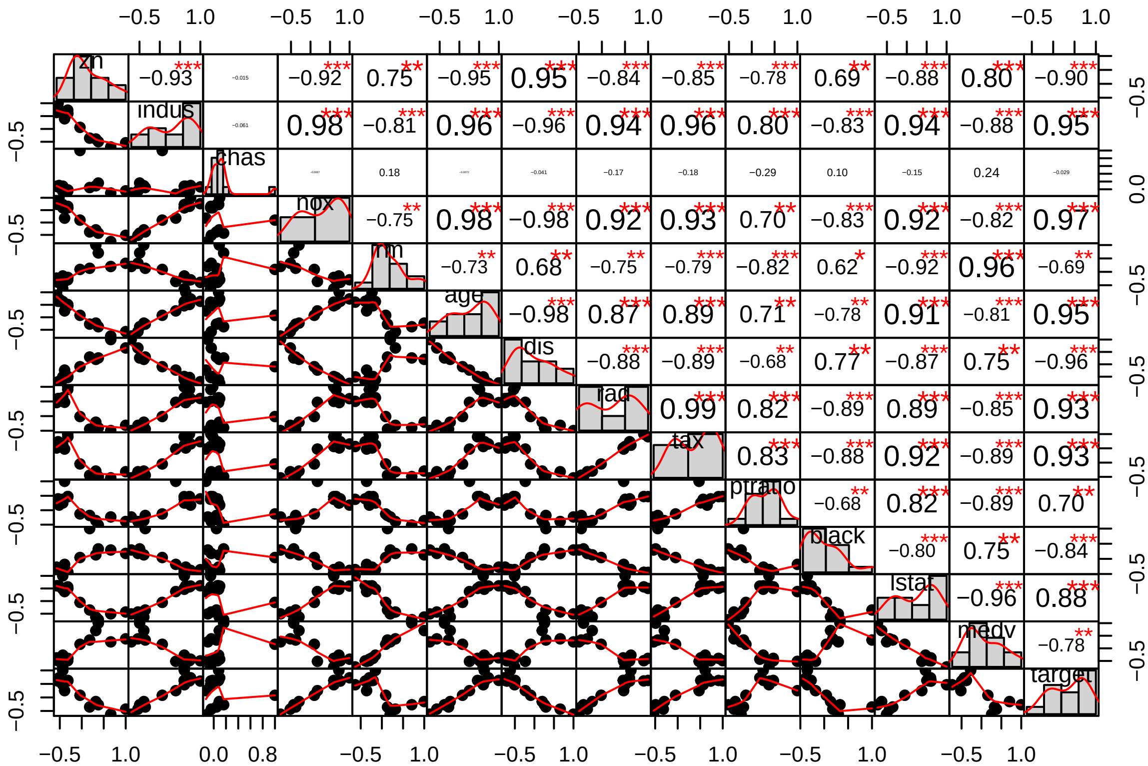
Visually assessing missing values:

The Amelia package has a plotting function missmap() that will plot the dataset and highlight missing values:



There are no missing values in the dataset. Lets plot the correlation between the variables.

Correlation:

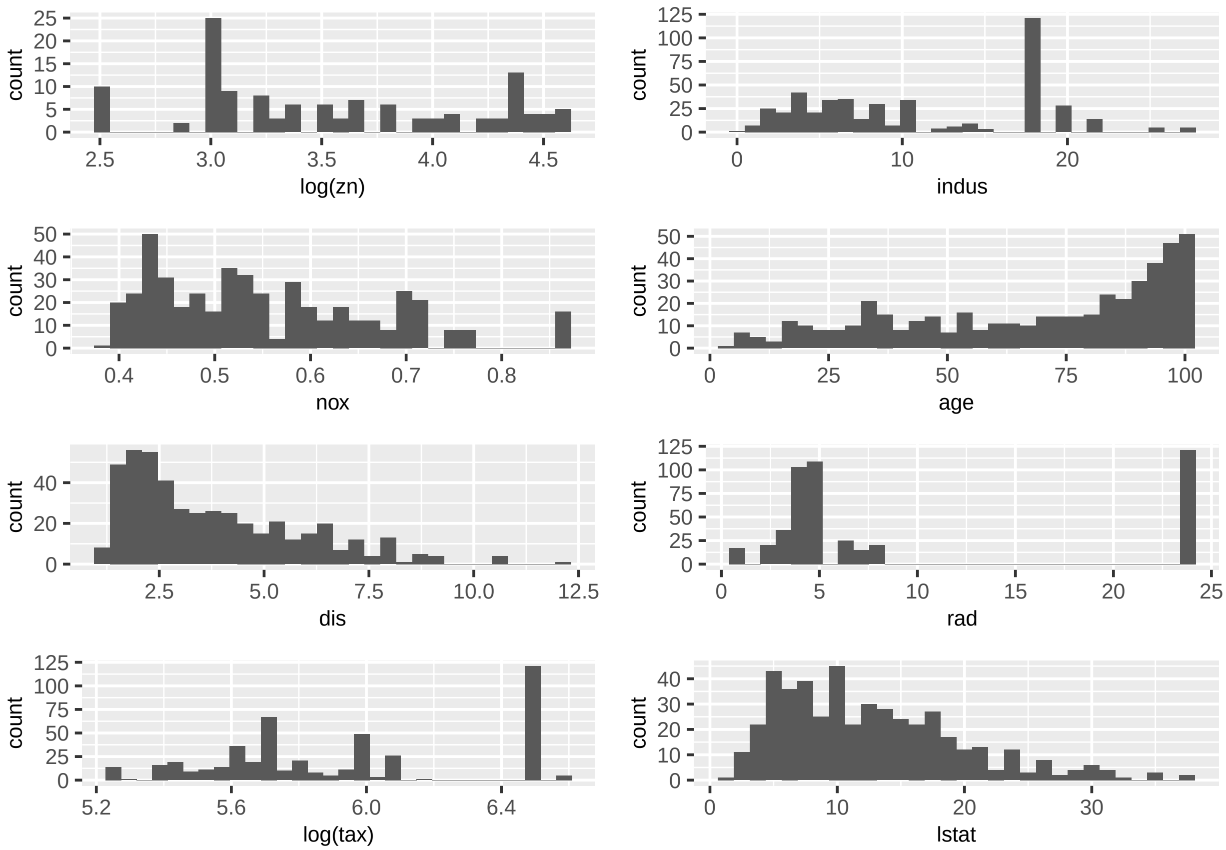


From the above correlation matrix , the **target** variable seems to have correlation with

- zn - proportion of residential land zoned for large lots
- indus - proportion of non-retail business acres per suburb
- nox - nitrogen oxides concentration
- age - proportion of owner-occupied units built prior to 1940
- dis - weighted mean of distances to five Boston employment centers
- rad - index of accessibility to radial highways
- tax - full-value property-tax rate per $10,000
- lstat - lower status of the population

Lets look at each of the predictor variable's data:

Distribution of Predictors:



From the above, it appears like majority of the neighborhoods have no residential land zoned for large lots. And the buildings with age > 100 are mentioned as 100 in the above. We could not derive a specific categorization in the other predictors.

Data Preparation

Factorize Variables:

Convert the *chas* and *target* variables into factors:

```
crime.trn$chas <- as.factor(crime.trn$chas)
crime.trn$target <- as.factor(crime.trn$target)
```

For ZN variable, 339/466 are zeros. We are going to create a new variable **zn_ind** as indicator for residential zones containing large lots (land size over 25,000 sq.ft as 1)

```
##
##   0   1
## 339 127
```

Check for Multicolinearity in the predictors:

Check for Multicolinearity among the predictor variables and remove those with excessive correlation among the explanatory variables.

	Multicolinearity score
medv	8.621044
rm	6.114789
zn_ind	5.292859
dis	4.736573
nox	4.493409
zn	3.971424
lstat	2.894468
indus	2.768836
ptratio	2.598383
age	2.582215
tax	2.179424
rad	1.898671
chas	1.276748
black	1.090804

From the above table, we do not see multi-colinearity (with VIF > 10) among the predictors.

Split the dataset into training and test:

We will randomly split our dataset into training (80%) and test (20%).

```
set.seed(999)
s = sample(1:nrow(crime.trn), 0.8 * nrow(crime.trn))
crime.train = crime.trn[s, ]
crime.test = crime.trn[-s, ]
```

Number of observations in *training* dataset is 372

Number of observations in *test* dataset is 94

Build Models

The below are the few different approaches we will try to build the models:

1. Stepwise Backward
2. Stepwise Forward
3. Manual
4. Bayesian

1. Backward elimination method

With backwards elimination, we start with full set of parameters and iteratively reduce the numbers of parameters using AIC.

```
##
## Call:
## stats::glm(formula = target ~ ., family = binomial(), data = crime.train)
##
## Deviance Residuals:
```

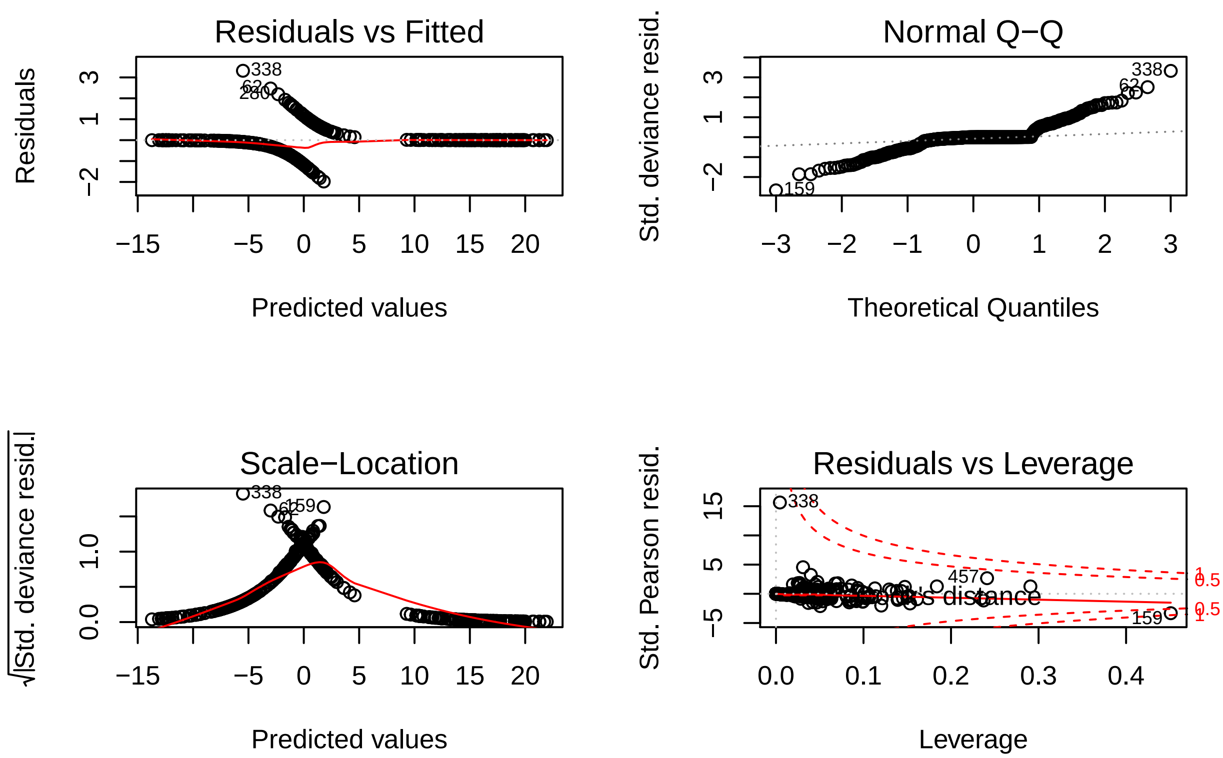
```
##     Min       1Q   Median       3Q      Max
## -2.0129  -0.1480  -0.0051   0.0029   3.4512
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -38.402903   7.657669  -5.015 5.30e-07 ***
## zn           -0.035528   0.051713  -0.687 0.492073
## indus        -0.067372   0.055720  -1.209 0.226615
## chas1         1.147487   0.801451   1.432 0.152212
## nox          47.077703   9.008344   5.226 1.73e-07 ***
## rm           -0.282639   0.820723  -0.344 0.730562
## age           0.032986   0.015177   2.173 0.029746 *
## dis           0.841631   0.276398   3.045 0.002327 **
## rad           0.617164   0.177894   3.469 0.000522 ***
## tax          -0.006143   0.003312  -1.855 0.063605 .
## ptratio       0.403943   0.153498   2.632 0.008499 **
## black        -0.011410   0.006191  -1.843 0.065338 .
## lstat         0.112049   0.064401   1.740 0.081883 .
## medv          0.190297   0.077879   2.444 0.014545 *
## zn_ind       -0.707989   1.451991  -0.488 0.625834
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 515.31  on 371  degrees of freedom
## Residual deviance: 151.82  on 357  degrees of freedom
## AIC: 181.82
##
## Number of Fisher Scoring iterations: 9
```

```
## target ~ zn + nox + age + dis + rad + tax + ptratio + black +
##     lstat + medv
```

```
##
## Call:
## stats::glm(formula = target ~ zn + nox + age + dis + rad + tax +
##     ptratio + black + lstat + medv, family = binomial(), data = crime.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9771  -0.1548  -0.0026   0.0028   3.3163
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.933559   6.996318  -4.993 5.94e-07 ***
## zn           -0.067617   0.036693  -1.843 0.065360 .
## nox          39.658956   7.198565   5.509 3.60e-08 ***
## age           0.030587   0.012501   2.447 0.014414 *
## dis           0.736719   0.246530   2.988 0.002805 **
## rad           0.695000   0.166956   4.163 3.14e-05 ***
## tax          -0.007973   0.003002  -2.656 0.007912 **
## ptratio       0.358211   0.130070   2.754 0.005887 **
## black        -0.010786   0.006052  -1.782 0.074728 .
```

```
## lstat          0.117233   0.054037   2.170 0.030045 *
## medv           0.167625   0.049876   3.361 0.000777 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 515.31  on 371  degrees of freedom
## Residual deviance: 155.24  on 361  degrees of freedom
## AIC: 177.24
##
## Number of Fisher Scoring iterations: 9
```

From the above table, the nox, rad predictors shows low p-value. A unit increase in nitrogen oxides concentration increases the log odds by 47.07, while increase in rad increases the log odds by 0.61. The next significant predictors are dis and ptratio.



In the residuals Vs Fitted graph, the red line is not flat, which indicates the linearity in residuals is not true. In the scale-location graph as well, the red line is not flat, which indicates that residual variance is not constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line.However, the Residual Vs Leverage plot has the redline not alligned with gray dotted line, this indicates that the assumption of standardized residuals centered around zero is NOT true here.

2. Forward elimination method

With forward elimination, we start with an empty candidate set of parameters and iteratively add variables using AIC.

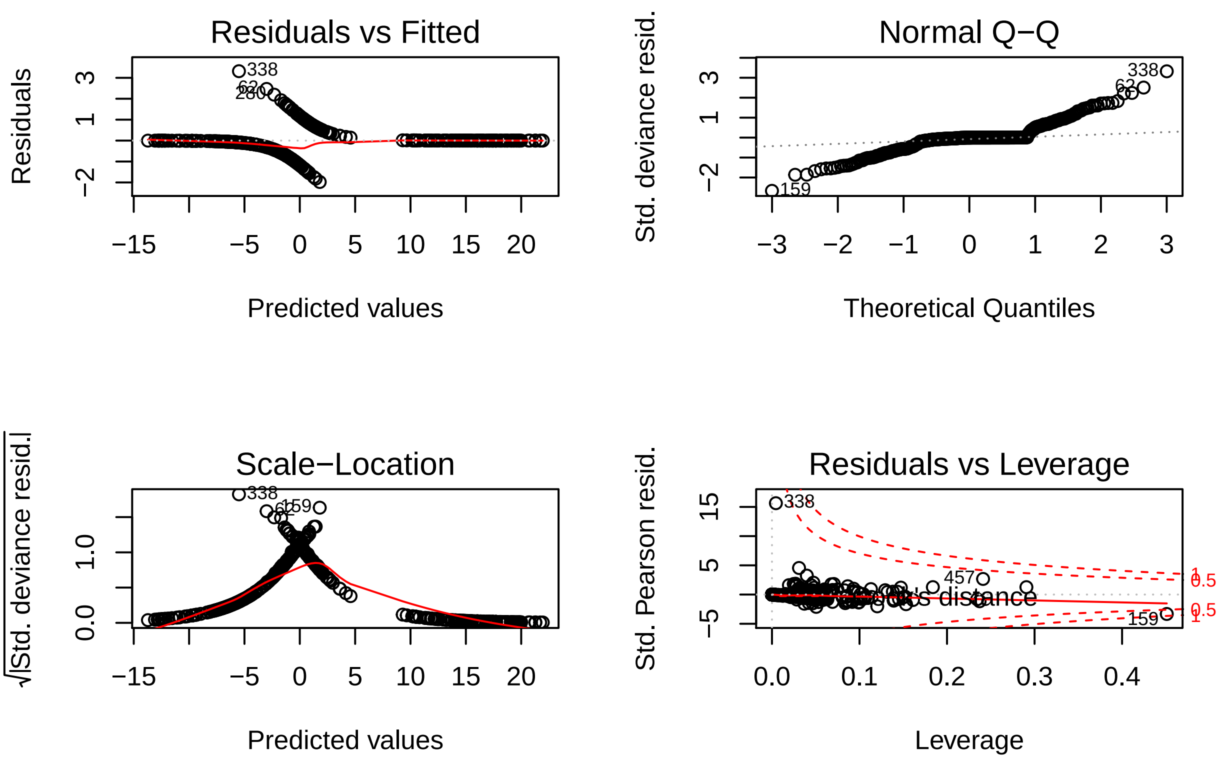
```
##
## Call:
## glm(formula = target ~ 1, family = binomial, data = crime.train,
##     trace = FALSE)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.150  -1.150  -1.150   1.205   1.205
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.06454    0.10375  -0.622    0.534
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 515.31  on 371  degrees of freedom
## Residual deviance: 515.31  on 371  degrees of freedom
## AIC: 517.31
##
## Number of Fisher Scoring iterations: 3
```

```
## target ~ nox + rad + tax + ptratio + age + black + medv + dis +
##     zn + lstat
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + age + black +
##     medv + dis + zn + lstat, family = binomial, data = crime.train,
##     trace = FALSE)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9771  -0.1548  -0.0026   0.0028   3.3163
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.933559   6.996318  -4.993 5.94e-07 ***
## nox          39.658956   7.198565   5.509 3.60e-08 ***
## rad           0.695000   0.166956   4.163 3.14e-05 ***
## tax          -0.007973   0.003002  -2.656 0.007912 **
## ptratio       0.358211   0.130070   2.754 0.005887 **
## age           0.030587   0.012501   2.447 0.014414 *
## black        -0.010786   0.006052  -1.782 0.074728 .
## medv          0.167625   0.049876   3.361 0.000777 ***
## dis           0.736719   0.246530   2.988 0.002805 **
## zn           -0.067617   0.036693  -1.843 0.065360 .
## lstat         0.117233   0.054037   2.170 0.030045 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 515.31  on 371  degrees of freedom
## Residual deviance: 155.24  on 361  degrees of freedom
## AIC: 177.24
##
## Number of Fisher Scoring iterations: 9
```

From the above table, the nox, rad, medv predictors shows low p-value. A unit increase in nitrogen oxides concentration increases the log odds by 39.65, while increase in rad increases the log odds by 0.69 and a unit increase in mdev increases the log odds by 0.16.



In the residuals Vs Fitted graph, the red line is not flat, which indicates the linearity in residuals is not true. In the scale-location graph as well, the red line is not flat, which indicates that residual variance is not constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line.However, the Residual Vs Leverage plot has the redline not alligned with gray dotted line, this indicates that the assumption of standardized residuals centered around zero is NOT true here.

From the above two models we can see that zn & age are not statistically significant. As for the statistically significant variables, rad & nox have a strong positive association of crime rate while tax has a negative coefficient, suggests as all other variables being equal as tax increases crime rate decreases.

3. Manual

Both Forward and backward elimination models produced the same model. Using the model obtained from backwards/forwards elimination, we next remove variables of low significance. We will remove Zn and age from the above models.

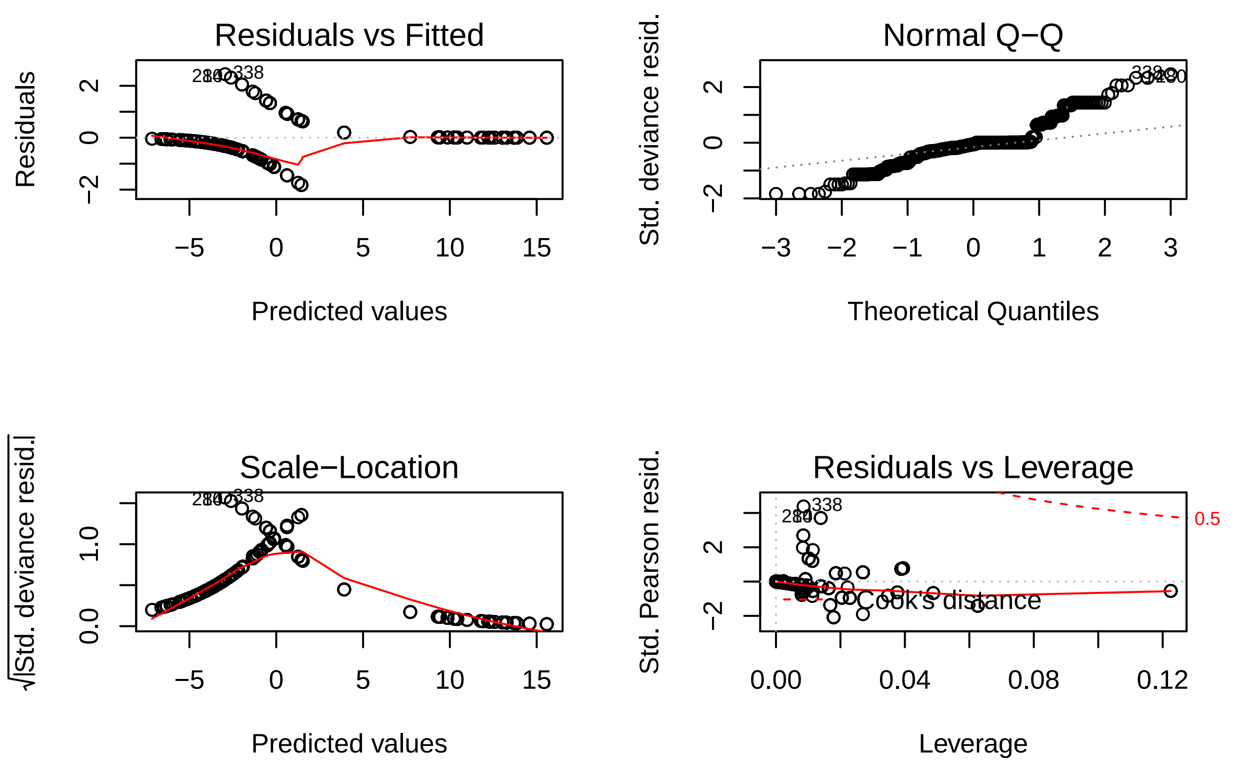
```
##
## Call:
## glm(formula = target ~ nox + rad + tax + dis, family = binomial(link = "logit"),
##     data = crime.train, trace = FALSE)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.85375  -0.31225  -0.06564   0.00749   2.51549
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -21.553260   3.652807  -5.900 3.62e-09 ***
## nox          37.708340   6.100531   6.181 6.36e-10 ***
## rad           0.562072   0.127164   4.420 9.87e-06 ***
## tax          -0.007533   0.002504  -3.009  0.00262 **
## dis           0.209893   0.161309   1.301  0.19319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 515.31  on 371  degrees of freedom
## Residual deviance: 184.63  on 367  degrees of freedom
## AIC: 194.63
##
## Number of Fisher Scoring iterations: 8
```

We will remove distance from the above model since the p value is not significant. Now the new model:

```
##
## Call:
## glm(formula = target ~ nox + rad + tax, family = binomial(link = "logit"),
##     data = crime.train, trace = FALSE)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.82233  -0.32010  -0.05947   0.00843   2.44822
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.250944   2.447132  -7.458 8.78e-14 ***
## nox          33.039818   4.740219   6.970 3.17e-12 ***
## rad           0.562869   0.127143   4.427 9.55e-06 ***
## tax          -0.007685   0.002517  -3.053  0.00227 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 515.31  on 371  degrees of freedom
## Residual deviance: 186.30  on 368  degrees of freedom
## AIC: 194.3
##
## Number of Fisher Scoring iterations: 8
```

A unit increase in index of accessibility to radial highways increses the log odds by 0.56. Also unit increase in nitrogen oxides concentration increases the log odds by 33.03, while increase in tax rate reduces the log odds by 0.008.



In the residuals Vs Fitted graph, the red line is not flat, which indicates the linearity in residuals is not true. In the scale-location graph as well, the red line is not flat, which indicates that residual variance is not constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line.However, the Residual Vs Leverage plot has the redline not alligned with gray dotted line, this indicates that the assumption of standardized residuals centered around zero is NOT true here.

4.Bayesian Approach

```
##
## Call:
## bic.glm.formula(f = target ~ ., data = crime.train, glm.family = "binomial")
##
##
##   34  models were selected
```

```
##  Best  5  models (cumulative posterior probability =  0.4857 ):
##
##            p!=0    EV         SD        model 1     model 2     model 3
## Intercept  100    -2.028e+01  5.000651  -1.825e+01  -2.354e+01  -1.719e+01
## zn          10.6  -4.531e-03  0.016035       .           .           .
## indus        2.3  -8.786e-04  0.009366       .           .           .
## chas         4.7
##     .1             4.816e-02  0.266073       .           .           .
## nox        100.0   3.271e+01  5.748499   3.304e+01   3.443e+01   2.846e+01
## rm           5.0   3.133e-02  0.182047       .           .           .
## age         18.2   3.739e-03  0.009083       .           .       2.014e-02
## dis          8.8   2.721e-02  0.106535       .           .           .
## rad        100.0   6.143e-01  0.147026   5.629e-01   6.701e-01   5.779e-01
## tax         98.9  -8.219e-03  0.002808  -7.685e-03  -8.698e-03  -8.246e-03
## ptratio     40.7   9.795e-02  0.135871       .       2.331e-01       .
## black       11.7  -1.080e-03  0.003581       .           .           .
## lstat       11.1   9.526e-03  0.032120       .           .           .
## medv         4.1   2.170e-03  0.015109       .           .           .
## zn_ind       6.4  -6.157e-02  0.295350       .           .           .
##
## nVar                                       3           4           4
## BIC                                     -1.992e+03  -1.992e+03  -1.990e+03
## post prob                                0.164       0.151       0.066
##            model 4     model 5
## Intercept  -2.264e+01  -1.633e+01
## zn              .      -4.111e-02
## indus           .           .
## chas
##     .1          .           .
## nox         3.025e+01   2.940e+01
## rm              .           .
## age         2.027e-02       .
## dis             .           .
## rad         6.906e-01   5.785e-01
## tax        -9.503e-03  -7.441e-03
## ptratio     2.321e-01       .
## black           .           .
## lstat           .           .
## medv            .           .
## zn_ind          .           .
##
## nVar          5           4
## BIC        -1.990e+03  -1.989e+03
## post prob   0.059       0.046
```

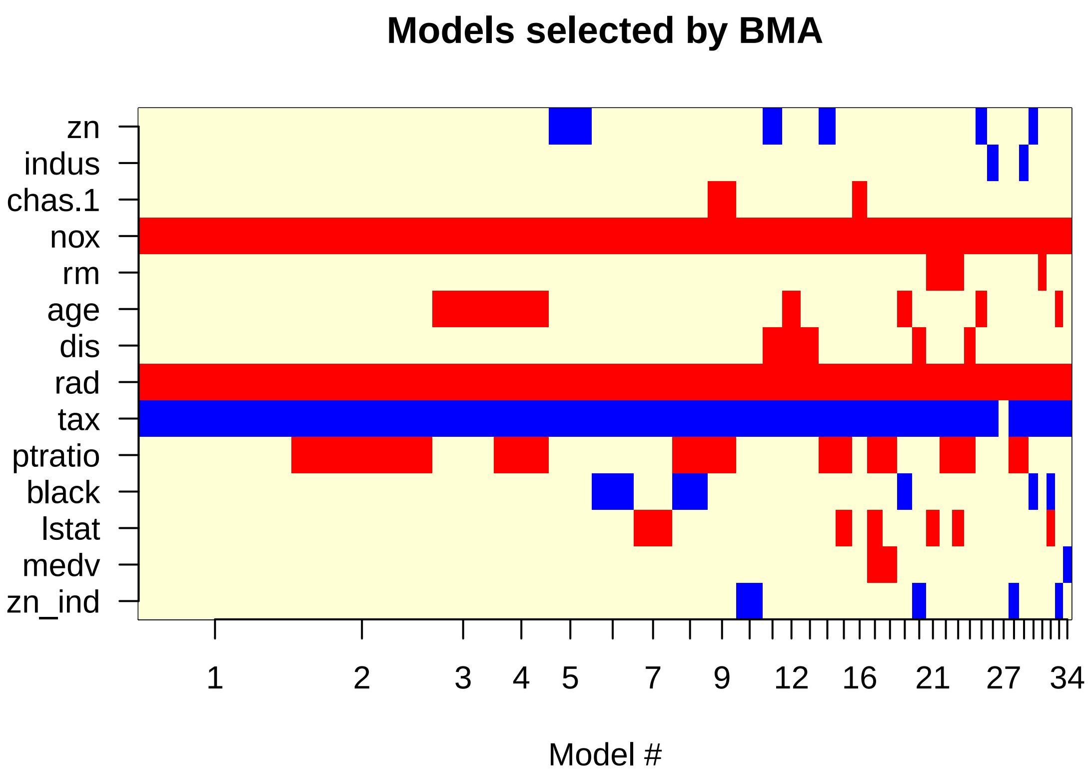
```
##  [1] 0.163696793 0.151392859 0.065592916 0.059231284 0.045806036
##  [6] 0.045023458 0.041481075 0.037812642 0.030815856 0.028442165
## [11] 0.020566195 0.020098049 0.019496949 0.017847764 0.017546644
## [16] 0.016378431 0.016324291 0.015949196 0.015794791 0.015230272
## [21] 0.014103230 0.013617958 0.012737391 0.012497881 0.012380272
## [26] 0.011982998 0.011157320 0.011011428 0.010521782 0.009747272
## [31] 0.009067229 0.009066517 0.008929459 0.008651597
```

```
##  [1] "nox,rad,tax"                    "nox,rad,tax,ptratio"
```

```
##  [3] "nox,age,rad,tax"                 "nox,age,rad,tax,ptratio"
##  [5] "zn,nox,rad,tax"                  "nox,rad,tax,black"
##  [7] "nox,rad,tax,lstat"               "nox,rad,tax,ptratio,black"
##  [9] "chas,nox,rad,tax,ptratio"        "nox,rad,tax,zn_ind"
## [11] "zn,nox,dis,rad,tax"              "nox,age,dis,rad,tax"
## [13] "nox,dis,rad,tax"                 "zn,nox,rad,tax,ptratio"
## [15] "nox,rad,tax,ptratio,lstat"       "chas,nox,rad,tax"
## [17] "nox,rad,tax,ptratio,lstat,medv" "nox,rad,tax,ptratio,medv"
## [19] "nox,age,rad,tax,black"           "nox,dis,rad,tax,zn_ind"
## [21] "nox,rm,rad,tax,lstat"            "nox,rm,rad,tax,ptratio"
## [23] "nox,rm,rad,tax,ptratio,lstat"   "nox,dis,rad,tax,ptratio"
## [25] "zn,nox,age,rad,tax"              "indus,nox,rad,tax"
## [27] "nox,rad"                         "nox,rad,tax,ptratio,zn_ind"
## [29] "indus,nox,rad,tax,ptratio"       "zn,nox,rad,tax,black"
## [31] "nox,rm,rad,tax"                  "nox,rad,tax,black,lstat"
## [33] "nox,age,rad,tax,zn_ind"          "nox,rad,tax,medv"
```

```
##  [1] "zn"      "indus"   "chas"    "nox"     "rm"      "age"     "dis"
##  [8] "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"    "zn_ind"
```

```
##      zn   indus    chas     nox      rm     age     dis     rad     tax
##    10.6     2.3     4.7   100.0     5.0    18.2     8.8   100.0    98.9
## ptratio   black   lstat    medv  zn_ind
##    40.7    11.7    11.1     4.1     6.4
```




```
##    (Intercept)             zn          indus           chas1             nox
```

```
## -2.028043e+01 -4.530699e-03 -8.786346e-04  4.815851e-02  3.270613e+01
##            rm           age           dis           rad           tax
##  3.133168e-02  3.738830e-03  2.721342e-02  6.143394e-01 -8.219396e-03
##       ptratio         black         lstat          medv        zn_ind
##  9.795350e-02 -1.079994e-03  9.526345e-03  2.170472e-03 -6.156917e-02
```

From the above resuls it appears like nitrogen oxides concentration(nox), accessibility to radial highways(rad) and property-tax rate(tax) are the 3 variables contributing across all 5 best models selected out of 35 models prepared by the baysian approach. Hence, we consider those 3 predictors for our baysian model. (target ~ nox+rad+tax)

Select Models

Majority of the models provided us with the below model formula:

target ~ nox+rad+tax

Let us try to apply the performance measurs to each of the above models and select the one with best possible accuracy.

Performance measures:

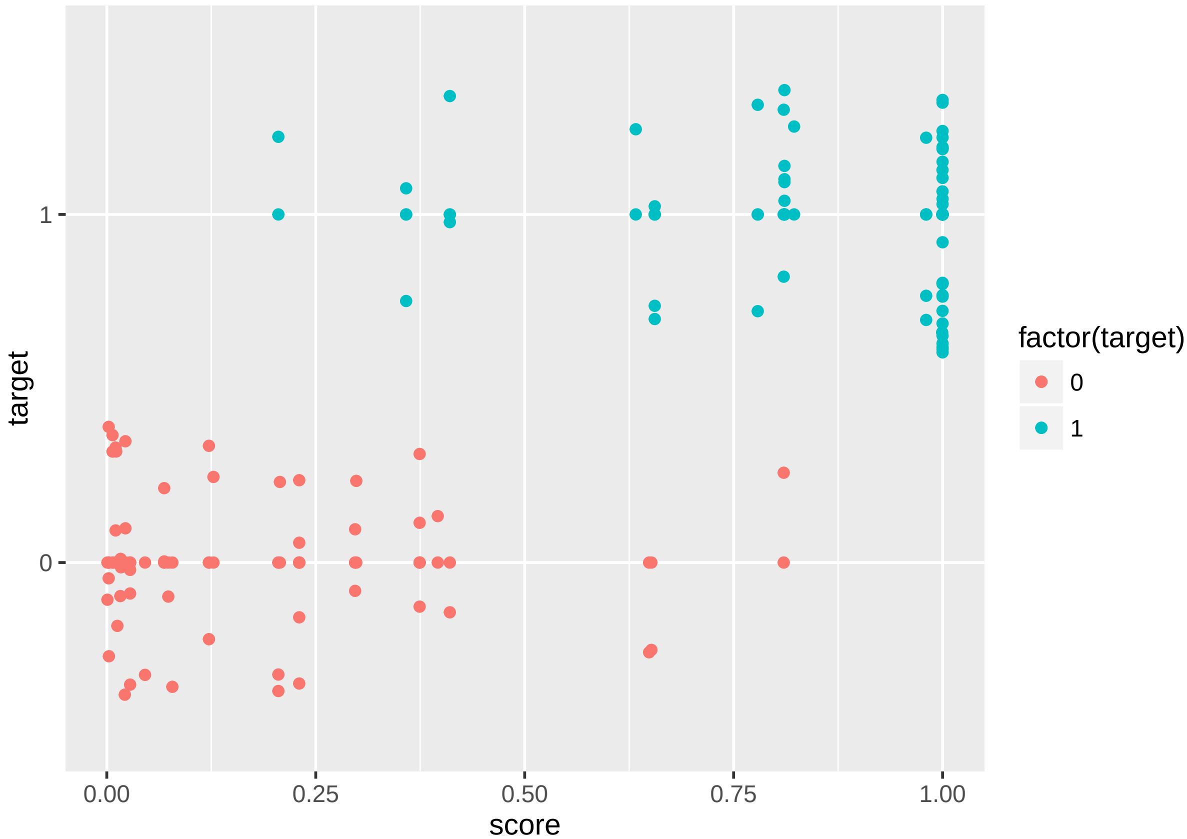
Sensitivity is basically the ability of the model to capture all positives. And Specificity is the ability of the model to capture all negatives.

$$Sensitivity = \frac{TP}{TP + FN}$$
$$Specificity = \frac{TN}{TN + FP}$$
$$Accuracy = \frac{TP + TN}{(TP + FN) + (FP + TN)}$$

For an ideal model, the predictions will be perfect - meaning the 'accuracy, sensitivity and specificity' will all be 1 where as the mis-classification error will be zero. In practical scenarios we would like to have the sensitivity and spcificity as close to 1 as possible.

Apply the performance measures on the Manual model

Score should be high when outcome is 1 and low when outcome is 0. Lets visualize how our binary response is behaving with respect to the score that we obtained.



We can see that the response 0 is bunched around low scores and response 1 is bunched around high scores. However, there is also overlap as well across some scores. We need to find a cutoff in the score so as to reach our target here.

Based on our previous homework, these are some properties of the cutoff/threshold:

All the predicted values above cutoff will be 1
All the predicted values below cutoff will be 0
Response values above cutoff(predicted 1) which are 1 in reality will be noted as TP
Response values above cutoff(predicted 1) which are 0 in reality will be noted as FP
Response values below cutoff(predicted 0) which are 1 in reality will be noted as FN
Response values below cutoff(predicted 0) which are 0 in reality will be noted as TN

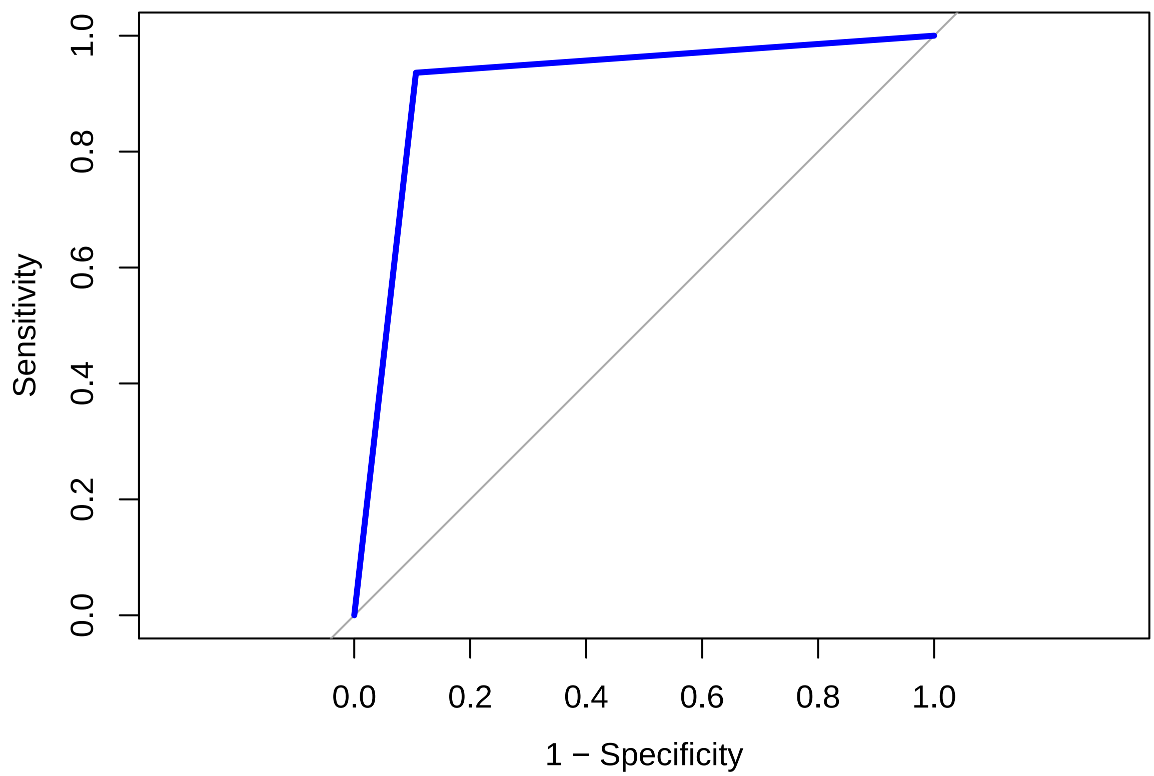
Based on our visualization, it appears like 0.50 could be a better cutoff.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 42  5
##          1  3 44
##
##                Accuracy : 0.9149
##                  95% CI : (0.8392, 0.9625)
##     No Information Rate : 0.5213
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8298
```

```
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.8980
##             Specificity : 0.9333
##          Pos Pred Value : 0.9362
##          Neg Pred Value : 0.8936
##              Prevalence : 0.5213
##          Detection Rate : 0.4681
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9156
##
##        'Positive' Class : 1
##
```

AUC for Manual model

```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.9149
## 95% CI: 0.858-0.9717 (DeLong)
```

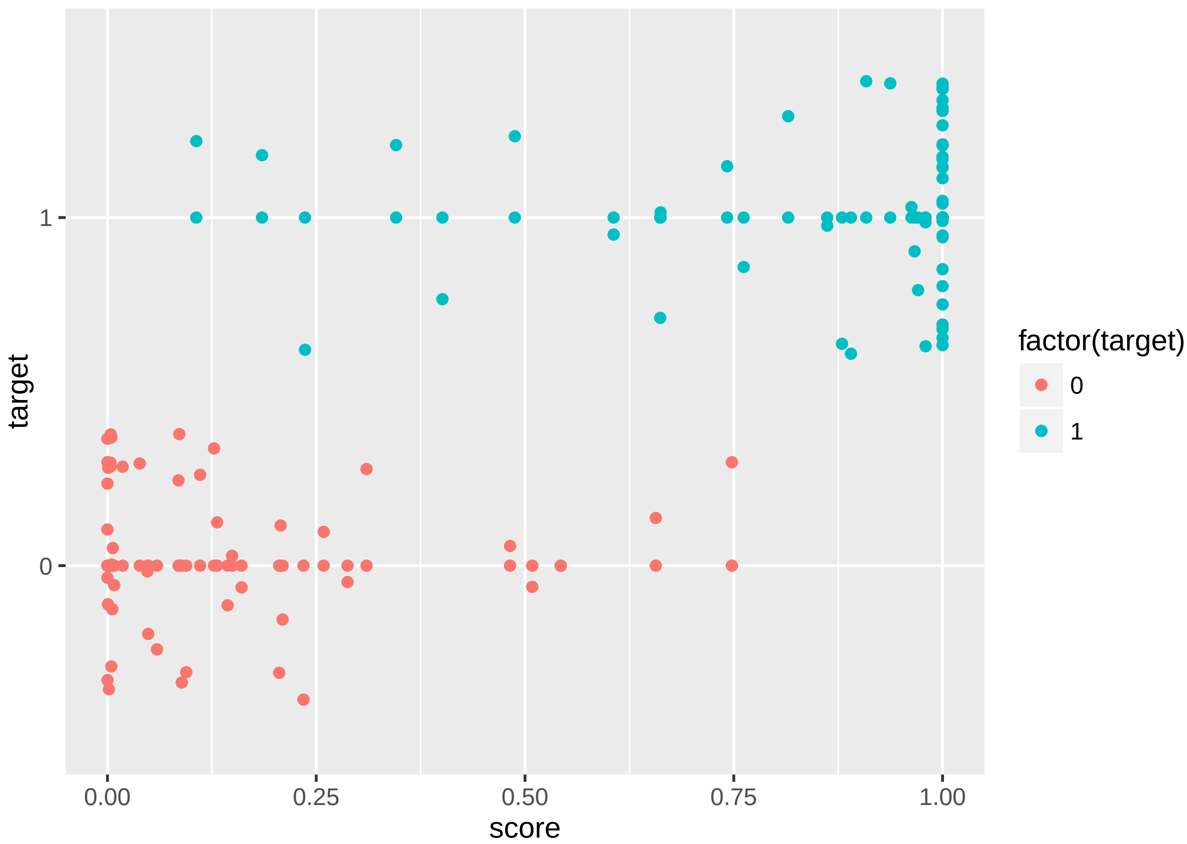



```
##
```

```
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, ci
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.9149
## 95% CI: 0.858-0.9717 (DeLong)

## [1] 0.9149
```

Apply the performance measures on the Forward Elimination model

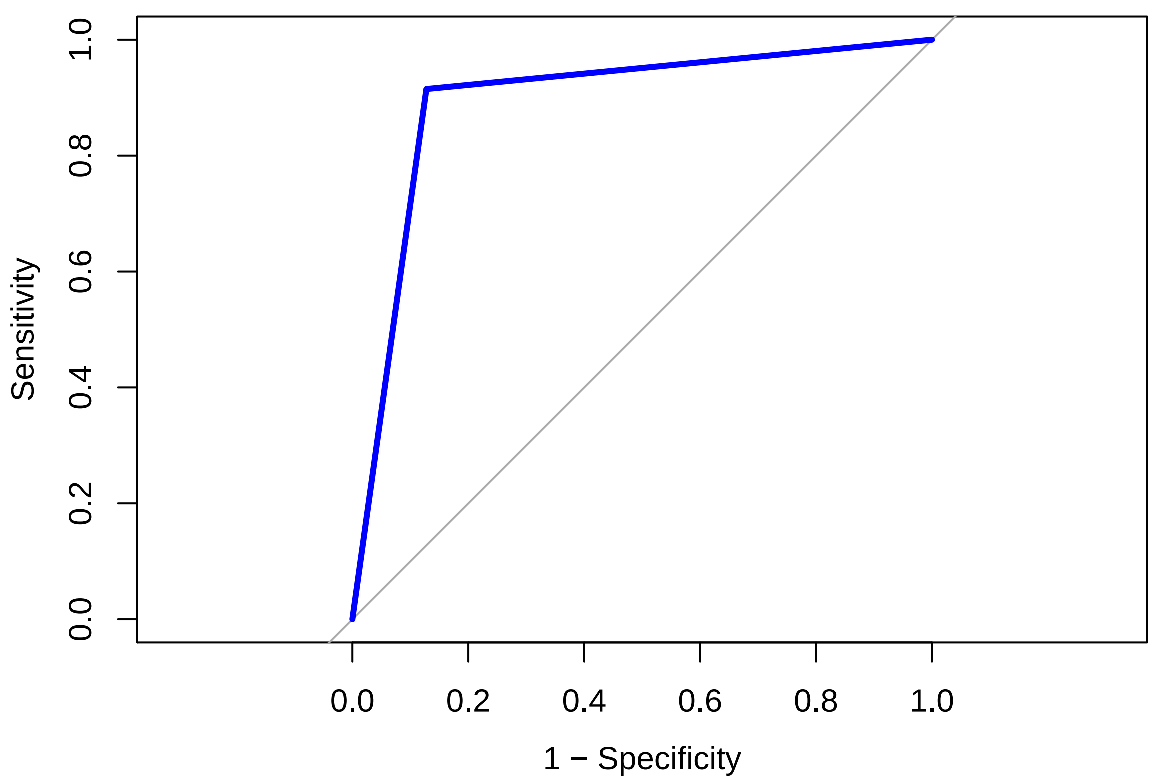



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 41  6
##          1  4 43
##
##                Accuracy : 0.8936
##                  95% CI : (0.813, 0.9478)
##     No Information Rate : 0.5213
##     P-Value [Acc > NIR] : 1.12e-14
##
##                   Kappa : 0.7872
##  Mcnemar's Test P-Value : 0.7518
##
```

```
##             Sensitivity : 0.8776
##             Specificity : 0.9111
##          Pos Pred Value : 0.9149
##          Neg Pred Value : 0.8723
##              Prevalence : 0.5213
##          Detection Rate : 0.4574
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8943
##
##        'Positive' Class : 1
##
```

AUC for Forward Elimination

```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.8936
## 95% CI: 0.8308-0.9565 (DeLong)
```

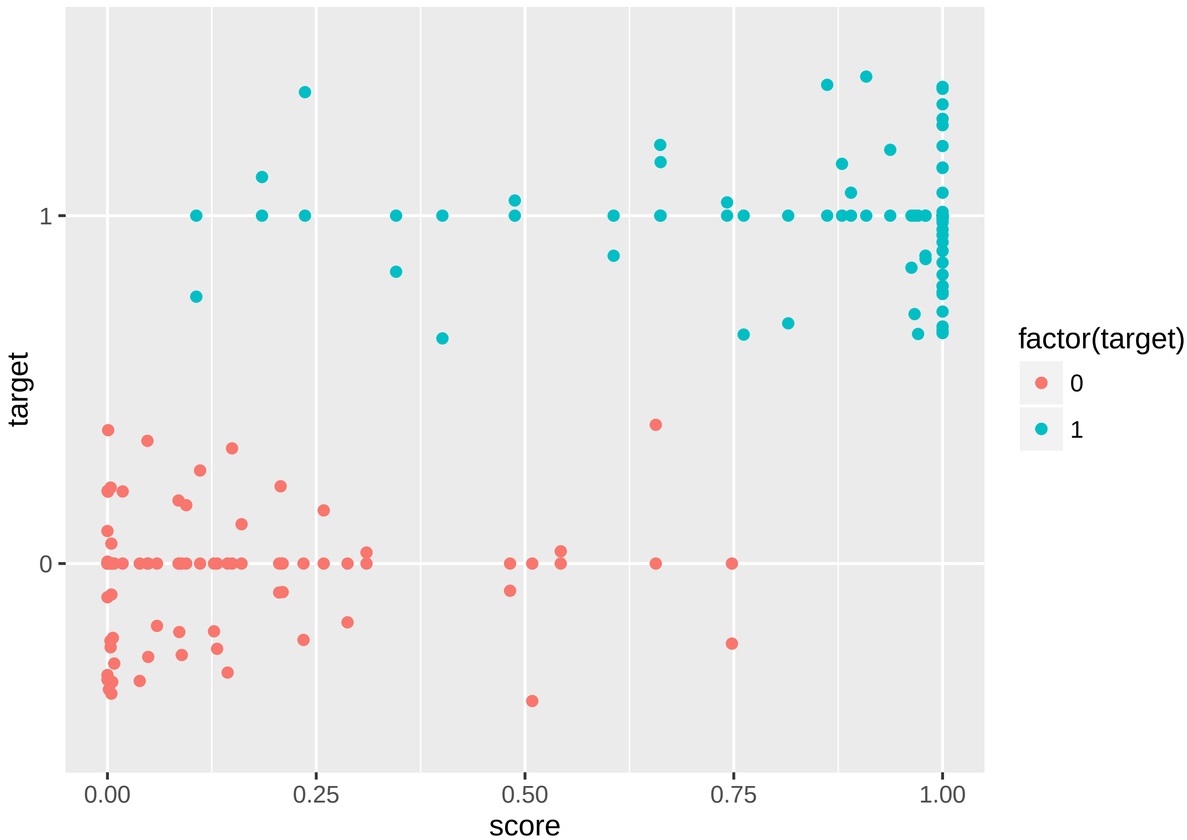



```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
```

```
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.8936
## 95% CI: 0.8308-0.9565 (DeLong)
```

```
## [1] 0.8936
```

Apply the performance measures on the Backward Elimination model

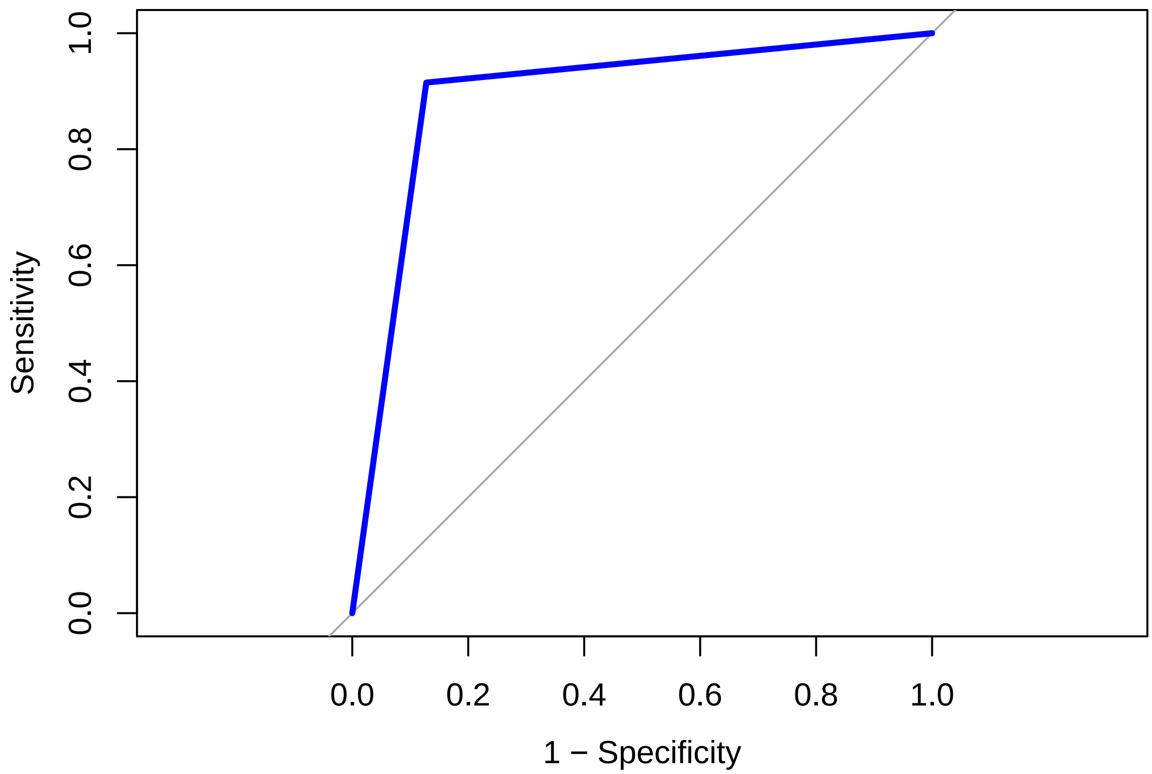



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 41  6
##          1  4 43
##
##                Accuracy : 0.8936
##                  95% CI : (0.813, 0.9478)
##     No Information Rate : 0.5213
##     P-Value [Acc > NIR] : 1.12e-14
##
##                   Kappa : 0.7872
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.8776
##             Specificity : 0.9111
```

```
##          Pos Pred Value : 0.9149
##          Neg Pred Value : 0.8723
##              Prevalence : 0.5213
##          Detection Rate : 0.4574
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8943
##
##        'Positive' Class : 1
##
```

AUC for Backward Elimination

```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.8936
## 95% CI: 0.8308-0.9565 (DeLong)
```

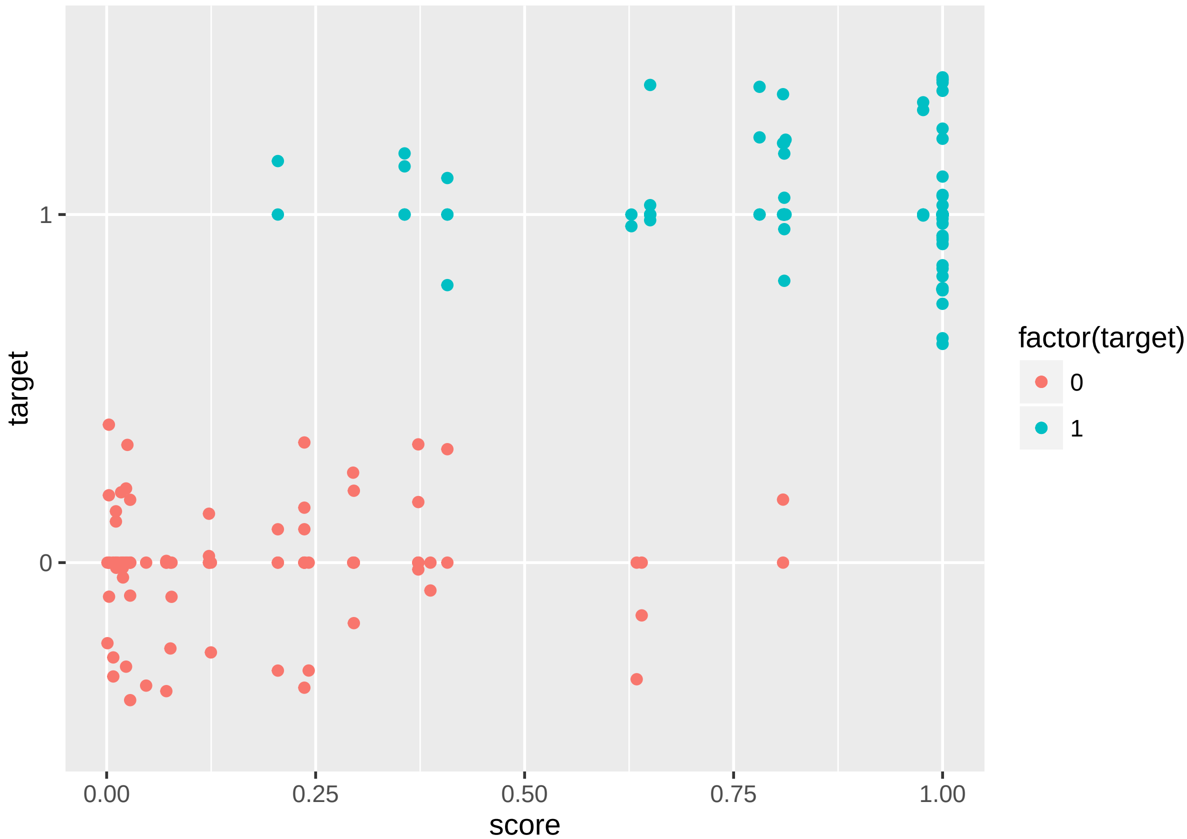



```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
```

```
## Area under the curve: 0.8936
## 95% CI: 0.8308-0.9565 (DeLong)
```

```
## [1] 0.8936
```

Apply the performance measures on the Bayesian model

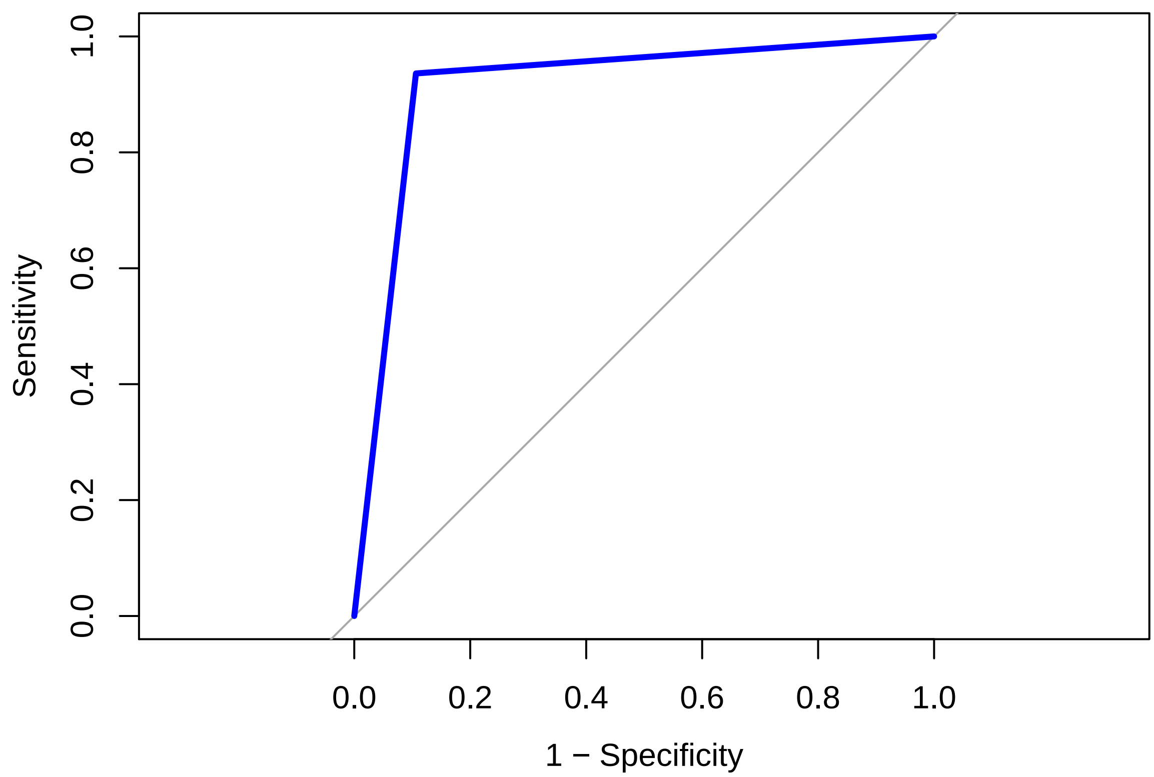



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 42  5
##          1  3 44
##
##                Accuracy : 0.9149
##                  95% CI : (0.8392, 0.9625)
##     No Information Rate : 0.5213
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8298
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.8980
##             Specificity : 0.9333
##          Pos Pred Value : 0.9362
##          Neg Pred Value : 0.8936
```

```
##               Prevalence : 0.5213
##           Detection Rate : 0.4681
##     Detection Prevalence : 0.5000
##        Balanced Accuracy : 0.9156
##
##         'Positive' Class : 1
##
```

AUC for Forward Elimination

```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.9149
## 95% CI: 0.858-0.9717 (DeLong)
```




```
##
## Call:
## roc.formula(formula = factor(predicted) ~ as.numeric(target),     data = crime.test, plot = FALSE, c
##
## Data: as.numeric(target) in 47 controls (factor(predicted) 0) < 47 cases (factor(predicted) 1).
## Area under the curve: 0.9149
## 95% CI: 0.858-0.9717 (DeLong)
```

```
## [1] 0.9149
```

Compare Results:

Method	Sn	Sp	Accuracy	AUC
Manual	0.898	0.9333	0.9149	0.9149
Forward Elimination	0.8776	0.9111	0.8936	0.8936
Backward Elimination	0.8776	0.9111	0.8936	0.8936
Bayesian Model	0.898	0.9333	0.9149	0.9149

From the above, we select Manual/Bayesian models to predict the target variable of the given dataset.

Predictions

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	predicted
0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	0
0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	380.02	10.26	18.2	0
0	8.14	0	0.538	6.495	94.4	4.4547	4	307	21.0	387.94	12.80	18.4	0
0	8.14	0	0.538	5.950	82.0	3.9900	4	307	21.0	232.60	27.71	13.2	0
0	5.96	0	0.499	5.850	41.5	3.9342	5	279	19.2	396.90	8.77	21.0	0
25	5.13	0	0.453	5.741	66.2	7.2254	8	284	19.7	395.11	13.15	18.7	0
25	5.13	0	0.453	5.966	93.4	6.8185	8	284	19.7	378.08	14.44	16.0	0
0	4.49	0	0.449	6.630	56.1	4.4377	3	247	18.5	392.30	6.53	26.6	0
0	4.49	0	0.449	6.121	56.8	3.7476	3	247	18.5	395.15	8.44	22.2	0
0	2.89	0	0.445	6.163	69.6	3.4952	2	276	18.0	391.83	11.34	21.4	0
0	25.65	0	0.581	5.856	97.0	1.9444	2	188	19.1	370.31	25.41	17.3	1
0	25.65	0	0.581	5.613	95.6	1.7572	2	188	19.1	359.29	27.26	15.7	1
0	21.89	0	0.624	5.637	94.7	1.9799	4	437	21.2	396.90	18.34	14.3	1
0	19.58	0	0.605	6.101	93.0	2.2834	5	403	14.7	240.16	9.81	25.0	1
0	19.58	0	0.605	5.880	97.3	2.3887	5	403	14.7	348.13	12.03	19.1	1
0	10.59	1	0.489	5.960	92.1	3.8771	4	277	18.6	393.25	17.27	21.7	0
0	6.20	0	0.504	6.552	21.4	3.3751	8	307	17.4	380.34	3.76	31.5	1
0	6.20	0	0.507	8.247	70.4	3.6519	8	307	17.4	378.95	3.95	48.3	1
22	5.86	0	0.431	6.957	6.8	8.9067	7	330	19.1	386.09	3.53	29.6	0
90	2.97	0	0.400	7.088	20.8	7.3073	1	285	15.3	394.72	7.85	32.2	0
80	1.76	0	0.385	6.230	31.5	9.0892	1	241	18.2	341.60	12.93	20.1	0
33	2.18	0	0.472	6.616	58.1	3.3700	7	222	18.4	393.36	8.93	28.4	0
0	9.90	0	0.544	6.122	52.8	2.6403	4	304	18.4	396.90	5.98	22.1	0
0	7.38	0	0.493	6.415	40.1	4.7211	5	287	19.6	396.90	6.12	25.0	0
0	7.38	0	0.493	6.312	28.9	5.4159	5	287	19.6	396.90	6.15	23.0	0
0	5.19	0	0.515	5.895	59.6	5.6150	5	224	20.2	394.81	10.56	18.5	0
80	2.01	0	0.435	6.635	29.7	8.3440	4	280	17.0	390.94	5.99	24.5	0
0	18.10	0	0.718	3.561	87.9	1.6132	24	666	20.2	354.70	7.12	27.5	1
0	18.10	1	0.631	7.016	97.5	1.2024	24	666	20.2	392.05	2.96	50.0	1
0	18.10	0	0.584	6.348	86.1	2.0527	24	666	20.2	83.45	17.64	14.5	1
0	18.10	0	0.740	5.935	87.9	1.8206	24	666	20.2	68.95	34.02	8.4	1
0	18.10	0	0.740	5.627	93.9	1.8172	24	666	20.2	396.90	22.88	12.8	1
0	18.10	0	0.740	5.818	92.4	1.8662	24	666	20.2	391.45	22.11	10.5	1
0	18.10	0	0.740	6.219	100.0	2.0048	24	666	20.2	395.69	16.59	18.4	1
0	18.10	0	0.740	5.854	96.6	1.8956	24	666	20.2	240.52	23.79	10.8	1

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	predicted
0	18.10	0	0.713	6.525	86.5	2.4358	24	666	20.2	50.92	18.13	14.1	1
0	18.10	0	0.713	6.376	88.4	2.5671	24	666	20.2	391.43	14.65	17.7	1
0	18.10	0	0.655	6.209	65.4	2.9634	24	666	20.2	396.90	13.22	21.4	1
0	9.69	0	0.585	5.794	70.6	2.8927	6	391	19.2	396.90	14.10	18.3	1
0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9	0

Appendix

```
library(dplyr)
library(ggplot2)
library(gridExtra)
library(e1071)
library(car)
library(recommenderlab)
library(knitr)
library(Amelia)
library(PerformanceAnalytics)
library(robustbase)
library(BMA)
library(caret)
library(pROC)

crime.trn <- read.csv("https://raw.githubusercontent.com/Nguyver/DATA621-HW/master/HW3/crime-training-da
    header = TRUE, sep = ",", stringsAsFactors = FALSE)

crime.evl <- read.csv("https://raw.githubusercontent.com/Nguyver/DATA621-HW/master/HW3/crime-evaluation-
    header = TRUE, sep = ",", stringsAsFactors = FALSE)

glimpse(crime.trn)

summary(crime.trn)

missmap(crime.trn, main = "Missing values vs observed")

cor.matrix <- cor(crime.trn[, 1:ncol(crime.trn)])
chart.Correlation(cor.matrix, histogram = TRUE, pch = 19)

g_zn <- ggplot(data = crime.trn) + geom_histogram(aes(x = log(zn))) + theme(axis.text = element_text(si
    axis.title = element_text(size = 8))
g_indus <- ggplot(data = crime.trn) + geom_histogram(aes(x = indus)) + theme(axis.text = element_text(s
    axis.title = element_text(size = 8))
g_nox <- ggplot(data = crime.trn) + geom_histogram(aes(x = nox)) + theme(axis.text = element_text(size
    axis.title = element_text(size = 8))
g_age <- ggplot(data = crime.trn) + geom_histogram(aes(x = age)) + theme(axis.text = element_text(size
    axis.title = element_text(size = 8))
g_dis <- ggplot(data = crime.trn) + geom_histogram(aes(x = dis)) + theme(axis.text = element_text(size
    axis.title = element_text(size = 8))
g_rad <- ggplot(data = crime.trn) + geom_histogram(aes(x = rad)) + theme(axis.text = element_text(size
    axis.title = element_text(size = 8))
g_tax <- ggplot(data = crime.trn) + geom_histogram(aes(x = log(tax))) + theme(axis.text = element_text(
```

```
    axis.title = element_text(size = 8))
g_lstat <- ggplot(data = crime.trn) + geom_histogram(aes(x = lstat)) + theme(axis.text = element_text(si
    axis.title = element_text(size = 8))
grid.arrange(g_zn, g_indus, g_nox, g_age, g_dis, g_rad, g_tax, g_lstat, ncol = 2)

crime.trn$chas <- as.factor(crime.trn$chas)
crime.trn$target <- as.factor(crime.trn$target)
crime.trn$zn_ind <- ifelse(crime.trn$zn > 0, 1, 0)
# crime.trn$zn <- ifelse(crime.trn$zn > 0, 1, 0)
table(crime.trn$zn_ind)

fit <- glm(target ~ ., data = crime.trn, family = binomial)
# Lets check for Multi-Collinearity - lets find vif value and drop those that has
vifFit1 <- vif(fit)
# sort by descending
vif.df <- as.data.frame(sort(vifFit1, decreasing = T))
names(vif.df) <- c("VIF")
kable(vif.df)

set.seed(999)
s = sample(1:nrow(crime.trn), 0.8 * nrow(crime.trn))
crime.train = crime.trn[s, ]
crime.test = crime.trn[-s, ]

fullmodel = stats::glm(target ~ ., family = binomial(), data = crime.train)
summary(fullmodel)

backwards.model = step(fullmodel, trace = FALSE)  #Backwards selection is the default
backwards.formula <- formula(backwards.model)
backwards.formula
summary(backwards.model)

par(mfrow = c(2, 2))
graphics::plot(backwards.model)

forwards.model = step(nothing, scope = list(lower = formula(nothing), upper = formula(fullmodel)),
    direction = "forward", trace = FALSE)
forwards.formula <- formula(forwards.model)
forwards.formula
summary(forwards.model)

par(mfrow = c(2, 2))
graphics::plot(backwards.model)


manual.model <- glm(target ~ nox + rad + tax + dis, family = binomial(link = "logit"),
    data = crime.train, trace = FALSE)
summary(manual.model)

manual.final <- glm(target ~ nox + rad + tax, family = binomial(link = "logit"),
    data = crime.train, trace = FALSE)
summary(manual.final)
```

```r
par(mfrow = c(2, 2))
graphics::plot(manual.final)

bayesian.model <- bic.glm(target ~ ., data = crime.train, glm.family = "binomial")
summary(bayesian.model)

# Posterior probability of each of 11 models (rest very small by comparison, so
# are omitted, change value of OR to see them)
bayesian.model$postprob
bayesian.model$label

# For each of 8 variables, probability they should be in the model
bayesian.model$names
bayesian.model$probne0

imageplot.bma(bayesian.model)
bayesian.model$postmean

bayesian.model.final <- bic.glm(target ~ nox + rad + tax, data = crime.train, glm.family = "binomial")

performance.results <- vector()

crime.test$score <- predict(manual.final, newdata = subset(crime.test, select = c(4,
    8, 9)), type = "response")
ggplot(crime.test, aes(y = target, x = score, color = factor(target))) + geom_point() +
    geom_jitter()

cutoff = 0.5
crime.test$predicted = as.numeric(crime.test$score > cutoff)
TP = sum(crime.test$predicted == 1 & crime.test$target == 1)
FP = sum(crime.test$predicted == 1 & crime.test$target == 0)
FN = sum(crime.test$predicted == 0 & crime.test$target == 1)
TN = sum(crime.test$predicted == 0 & crime.test$target == 0)

# lets also calculate total number of real positives and negatives in the data
P = TP + FN
N = TN + FP
total = P + N

confusionMatrix(factor(crime.test$predicted), factor(crime.test$target), positive = "1")

sensitivity <- round(sensitivity(factor(crime.test$predicted), crime.test$target,
    positive = "1"), 4)

specificity <- round(specificity(factor(crime.test$predicted), crime.test$target,
    negative = "0"), 4)

# accuracy = (TP+TN)/(P+N)
accuracy <- round(((TP + TN)/(P + N)), 4)

cnfMtx <- confusionMatrix(crime.test$predicted, crime.test$target, positive = "1")

(roc <- roc(factor(predicted) ~ as.numeric(target), data = crime.test, plot = FALSE,
```

```
    ci = TRUE))
graphics::plot(roc, legacy.axes = TRUE, col = "blue", lwd = 3)
(auc <- round(auc(factor(predicted) ~ as.numeric(target), crime.test), 4))

performance.results <- rbind(performance.results, c("Manual", sensitivity, specificity,
    accuracy, auc))

crime.test$score <- predict(forwards.model, newdata = subset(crime.test, select = c(nox,
    rad, tax, ptratio, age, black, medv, dis, zn, lstat)), type = "response")
ggplot(crime.test, aes(y = target, x = score, color = factor(target))) + geom_point() +
    geom_jitter()


cutoff = 0.5
crime.test$predicted = as.numeric(crime.test$score > cutoff)
TP = sum(crime.test$predicted == 1 & crime.test$target == 1)
FP = sum(crime.test$predicted == 1 & crime.test$target == 0)
FN = sum(crime.test$predicted == 0 & crime.test$target == 1)
TN = sum(crime.test$predicted == 0 & crime.test$target == 0)

# lets also calculate total number of real positives and negatives in the data
P = TP + FN
N = TN + FP
total = P + N

confusionMatrix(factor(crime.test$predicted), factor(crime.test$target), positive = "1")

sensitivity <- round(sensitivity(factor(crime.test$predicted), crime.test$target,
    positive = "1"), 4)

specificity <- round(specificity(factor(crime.test$predicted), crime.test$target,
    negative = "0"), 4)

# accuracy = (TP+TN)/(P+N)
accuracy <- round(((TP + TN)/(P + N)), 4)

cnfMtx <- confusionMatrix(crime.test$predicted, crime.test$target, positive = "1")

(roc <- roc(factor(predicted) ~ as.numeric(target), data = crime.test, plot = FALSE,
    ci = TRUE))
graphics::plot(roc, legacy.axes = TRUE, col = "blue", lwd = 3)
(auc <- round(auc(factor(predicted) ~ as.numeric(target), crime.test), 4))

performance.results <- rbind(performance.results, c("Forward Elimination", sensitivity,
    specificity, accuracy, auc))

crime.test$score <- predict(backwards.model, newdata = subset(crime.test, select = c(zn,
    nox, age, dis, rad, tax, ptratio, black, lstat, medv)), type = "response")
ggplot(crime.test, aes(y = target, x = score, color = factor(target))) + geom_point() +
    geom_jitter()


cutoff = 0.5
```

```r
crime.test$predicted = as.numeric(crime.test$score > cutoff)
TP = sum(crime.test$predicted == 1 & crime.test$target == 1)
FP = sum(crime.test$predicted == 1 & crime.test$target == 0)
FN = sum(crime.test$predicted == 0 & crime.test$target == 1)
TN = sum(crime.test$predicted == 0 & crime.test$target == 0)

# lets also calculate total number of real positives and negatives in the data
P = TP + FN
N = TN + FP
total = P + N

confusionMatrix(factor(crime.test$predicted), factor(crime.test$target), positive = "1")

sensitivity <- round(sensitivity(factor(crime.test$predicted), crime.test$target,
    positive = "1"), 4)

specificity <- round(specificity(factor(crime.test$predicted), crime.test$target,
    negative = "0"), 4)

# accuracy = (TP+TN)/(P+N)
accuracy <- round(((TP + TN)/(P + N)), 4)

cnfMtx <- confusionMatrix(crime.test$predicted, crime.test$target, positive = "1")

(roc <- roc(factor(predicted) ~ as.numeric(target), data = crime.test, plot = FALSE,
    ci = TRUE))
graphics::plot(roc, legacy.axes = TRUE, col = "blue", lwd = 3)
(auc <- round(auc(factor(predicted) ~ as.numeric(target), crime.test), 4))

performance.results <- rbind(performance.results, c("Backward Elimination", sensitivity,
    specificity, accuracy, auc))

crime.test$score <- predict(bayesian.model.final, newdata = subset(crime.test, select = c(nox,
    rad, tax, target)), type = "response")
ggplot(crime.test, aes(y = target, x = score, color = factor(target))) + geom_point() +
    geom_jitter()


cutoff = 0.5
crime.test$predicted = as.numeric(crime.test$score > cutoff)
TP = sum(crime.test$predicted == 1 & crime.test$target == 1)
FP = sum(crime.test$predicted == 1 & crime.test$target == 0)
FN = sum(crime.test$predicted == 0 & crime.test$target == 1)
TN = sum(crime.test$predicted == 0 & crime.test$target == 0)

# lets also calculate total number of real positives and negatives in the data
P = TP + FN
N = TN + FP
total = P + N

confusionMatrix(factor(crime.test$predicted), factor(crime.test$target), positive = "1")

sensitivity <- round(sensitivity(factor(crime.test$predicted), crime.test$target,
```

```r
    positive = "1"), 4)

specificity <- round(specificity(factor(crime.test$predicted), crime.test$target,
    negative = "0"), 4)

# accuracy = (TP+TN)/(P+N)
accuracy <- round(((TP + TN)/(P + N)), 4)

cnfMtx <- confusionMatrix(crime.test$predicted, crime.test$target, positive = "1")

(roc <- roc(factor(predicted) ~ as.numeric(target), data = crime.test, plot = FALSE,
    ci = TRUE))
graphics::plot(roc, legacy.axes = TRUE, col = "blue", lwd = 3)
(auc <- round(auc(factor(predicted) ~ as.numeric(target), crime.test), 4))
performance.results <- rbind(performance.results, c("Bayesian Model", sensitivity,
    specificity, accuracy, auc))


results <- as.data.frame(performance.results)
colnames(results) <- c("Method", "Sn", "Sp", "Accuracy", "AUC")
kable(results)

crime.evl$chas <- as.factor(crime.evl$chas)

crime.prd <- predict(manual.final, newdata = subset(crime.evl, select = c(4, 8, 9)),
    type = "response")
crime.prd <- ifelse(crime.prd > 0.5, 1, 0)

crime.evl$predicted <- crime.prd
crime.evl$predicted <- factor(crime.evl$predicted)

kable(crime.evl)
```