DATA621 - HW#3

Mia Chen, Wei Zhou

4/12/2020

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).

Your objective is to 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). 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 radial highways (predictor variable)
- tax: full-value property-tax rate per $10,000 (predictor variable)
- ptratio: pupil-teacher ratio by town (predictor variable)
- black: 1000(Bk - 0.63)2 where Bk 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)

Deliverables:

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
- Assigned prediction (probabilities, classifications) for the evaluation data set. Use 0.5 threshold.
- Include your R statistical programming code in an Appendix.

1. DATA EXPLORATION

Describe the size and the variables in the crime training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job.

Data acquisition

```
train <- read.csv("https://raw.githubusercontent.com/miachen410/DATA621/master/HW%233/crime-training-da
eval <- read.csv("https://raw.githubusercontent.com/miachen410/DATA621/master/HW%233/crime-evaluation-da
```

Data structure

There are 466 observations and 13 variables in the training dataset.

```
str(train)
```

```
## 'data.frame':    466 obs. of  13 variables:
##  $ zn     : num  0 0 0 30 0 0 0 0 0 80 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : num  96.2 100 100 7.8 92.2 71.3 100 100 38.1 19.1 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ rad    : int  5 5 24 6 3 5 24 24 5 1 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

Just to make the data easier on the eyes, we convert the 1s in `chas` to "Y", if the neighborhood borders Charles River, and 0s to "N", if not.

```
train$chas[train$chas == 1] <- "Y"
train$chas[train$chas == 0] <- "N"
```

We also convert the 1s in `target` to "Above", if the crime rate is above the median, and 0s to "Below", if it is below the median.

```
train$target[train$target == 1] <- "Above"
train$target[train$target == 0] <- "Below"
```

Since variables `chas` and `target` are categorical, we are going to change their class from integer to factor:

```
train$chas <- as.factor(train$chas)
train$target <- as.factor(train$target)
```

Let's look at the data structure again:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(train)
```

```
## Observations: 466
## Variables: 13
## $ 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    <fct> N, Y, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, ...
## $ 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, ...
## $ 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  <fct> Above, Above, Above, Below, Below, Below, Above, Above...
```

Summary Statistics

Looking at the `target` variable, we see 237 observations are below the median crime rate and 229 are above the median crime rate, thus we have roughly the same number of at risk and not-at-risk neighborhoods in our training data set.

```
summary(train)
```

```
##        zn              indus        chas         nox
##  Min.   :  0.00   Min.   : 0.460   N:433   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   Y: 33   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690           Median :0.5380
##  Mean   : 11.58   Mean   :11.105           Mean   :0.5543
##  3rd Qu.: 16.25   3rd Qu.:18.100           3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740           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
##  Above:229
##  Below:237
##
##
##
##
```

Missing values

How many rows of data have NA values? 0 rows, thus there are no missing values in the dataset.

```
nrow(train[is.na(train),])
```

```
## [1] 0
```

Visulization of the data set

Let's first look at the density plots of the numerical variables to view their shapes and distributions:

```
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

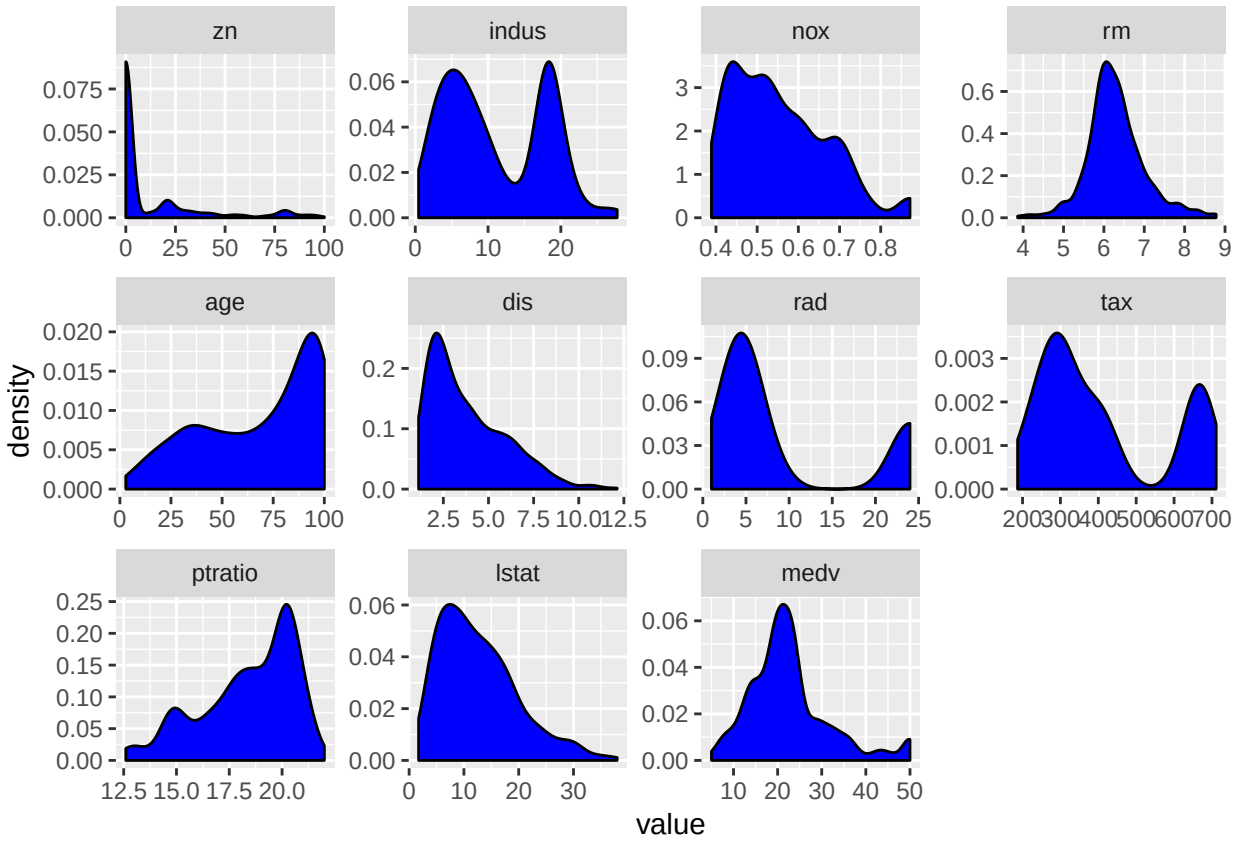
```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
library(ggplot2)

datasub = melt(train)
```

```
## Using chas, target as id variables
```

```
ggplot(datasub, aes(x = value)) +
    geom_density(fill = "blue") +
    facet_wrap(~variable, scales = 'free')
```



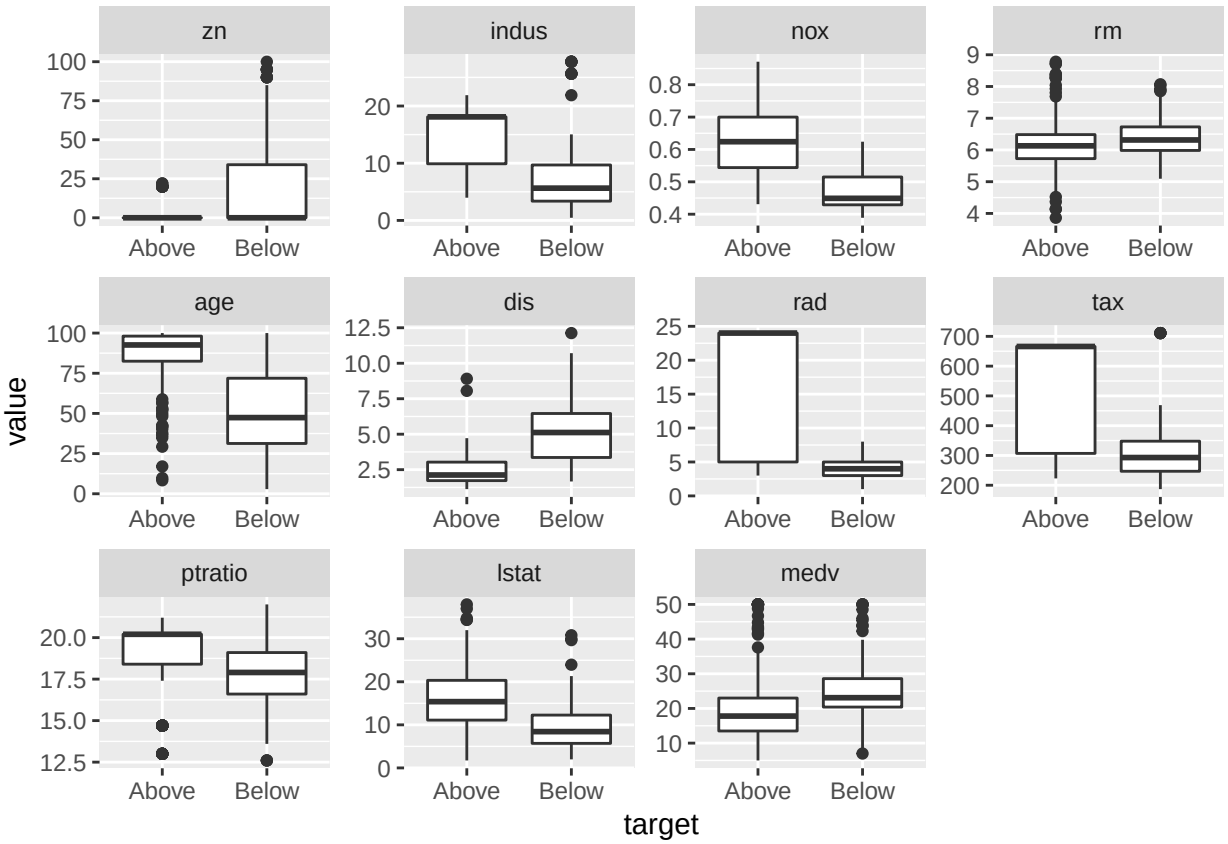
For categorial variable `chas`, we can look at a confusion matrix table to make sure that we have enough observations for all levels:

```
xtabs(~ target + chas, data=train)
```

```
##        chas
## target    N   Y
##   Above 208  21
##   Below 225  12
```

Then we will look at the boxplots of the numerical variables in relationship to `target` variable:

```
ggplot(datasub, aes(x = target, y = value)) +
    geom_boxplot() +
    facet_wrap(~variable, scales = 'free')
```



2. DATA PREPARATION

Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this.

Outlier Imputation

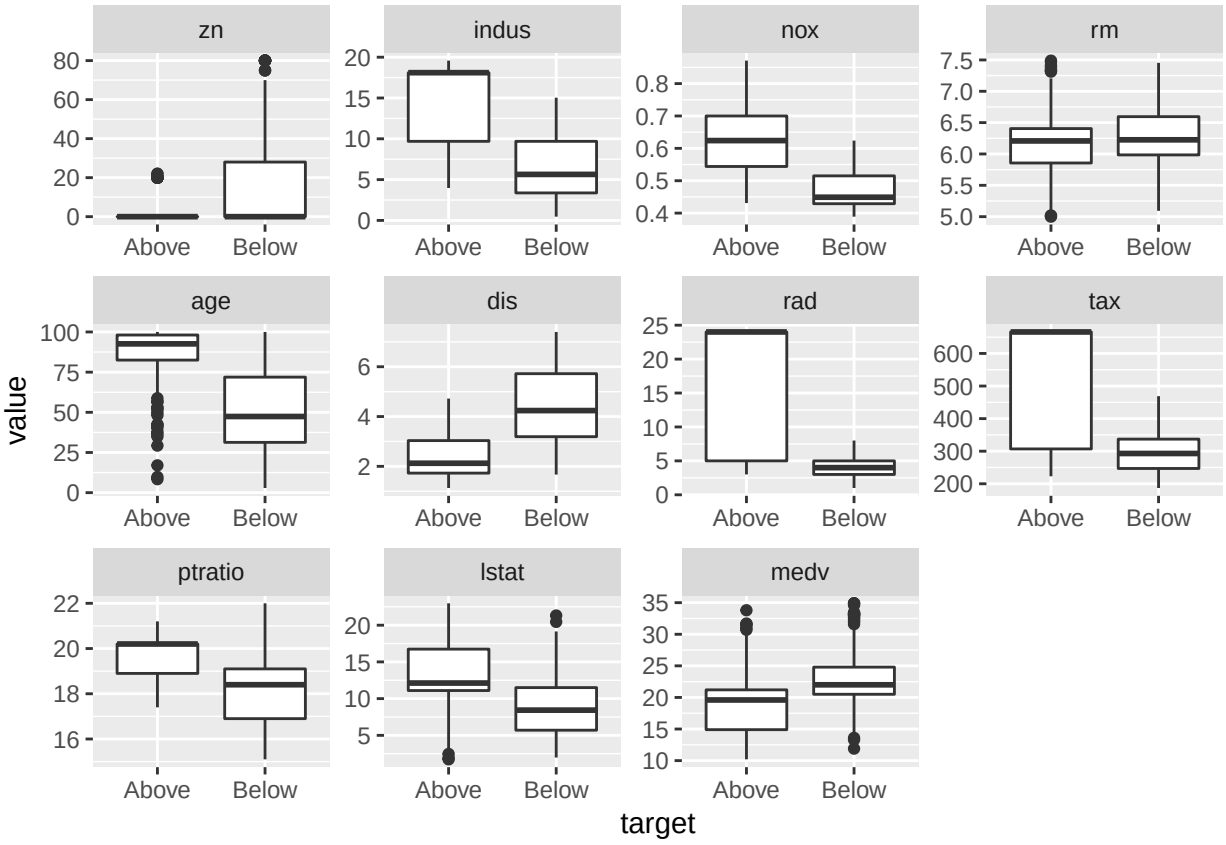
From the boxplots above, we can see there are many outliers so we are going to fix them by replacing with medians.

```
train_clean <- train %>% mutate(
  zn = ifelse(zn > 80, median(zn), zn),
  indus = ifelse(indus > 20, median(indus), indus),
  rm = ifelse(rm > 7.5 | rm < 5, median(rm), rm),
  dis = ifelse(dis > 7.5, median(dis), dis),
  tax = ifelse(tax >= 700, median(tax), tax),
  ptratio = ifelse(ptratio < 15, median(ptratio), ptratio),
  lstat = ifelse(lstat > 23, median(lstat), lstat),
  medv = ifelse(medv > 35 | medv < 10, median(medv), medv)
)
```

Let's look at the boxplots again after the outliers being imputed with median.

```
ggplot(melt(train_clean), aes(x = target, y = value)) +
    geom_boxplot() +
    facet_wrap(~variable, scales = 'free')
```

```
## Using chas, target as id variables
```



3. BUILD MODELS

Using the training data, build at least three different binary logistic regression models, using different variables (or the same variables with different transformations). You may select the variables manually, use an approach such as Forward or Stepwise, use a different approach, or use a combination of techniques. Describe the techniques you used. If you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done.

Be sure to explain how you can make inferences from the model, as well as discuss other relevant model output. Discuss the coefficients in the models, do they make sense? Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

We first create a full model by including all the variables:

```
fullMod <- glm(target ~ ., data = train_clean, family = binomial)
summary(fullMod)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.09313  -0.00050   0.00421   0.20666   1.76615
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  29.249903   5.699944   5.132 2.87e-07 ***
## zn            0.032271   0.024839   1.299 0.193863
## indus        -0.105101   0.081827  -1.284 0.198995
## chasY        -1.025767   0.748821  -1.370 0.170736
## nox         -25.754221   5.147415  -5.003 5.63e-07 ***
## rm           -0.233295   0.572700  -0.407 0.683744
## age          -0.021246   0.011243  -1.890 0.058793 .
## dis          -0.238212   0.210127  -1.134 0.256940
## rad          -0.887875   0.179730  -4.940 7.81e-07 ***
## tax           0.008718   0.004243   2.055 0.039902 *
## ptratio      -0.579561   0.158432  -3.658 0.000254 ***
## lstat         0.090336   0.053658   1.684 0.092265 .
## medv          0.056972   0.067714   0.841 0.400144
## ---
## 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: 204.02  on 453  degrees of freedom
## AIC: 230.02
##
## Number of Fisher Scoring iterations: 9
```

Model 1 - P-values Selection

From the full model, we select the variables that have small p-values:

target ~ nox + age + rad + tax + ptratio + lstat

```
logMod1 <- glm(target ~ nox + age + rad + tax + ptratio + lstat,
               data = train_clean,
               family = binomial)
summary(logMod1)
```

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax + ptratio + lstat,
##     family = binomial, data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.84428  -0.00233   0.01712   0.25100   1.96130
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  25.838414   3.495863   7.391 1.46e-13 ***
## nox         -24.449057   4.606338  -5.308 1.11e-07 ***
## age          -0.023855   0.009854  -2.421 0.015478 *
## rad          -0.748167   0.140552  -5.323 1.02e-07 ***
## tax           0.006514   0.003249   2.005 0.044955 *
```

```
## ptratio      -0.514274   0.133053  -3.865 0.000111 ***
## lstat         0.068845   0.043030   1.600 0.109616
## ---
## 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: 215.34  on 459  degrees of freedom
## AIC: 229.34
##
## Number of Fisher Scoring iterations: 8
```

Model 2 - Backward Selection

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
logMod2 <- fullMod %>% stepAIC(direction = "backward", trace = FALSE)
summary(logMod2)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + rad + tax + ptratio +
##     lstat, family = binomial, data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.02587  -0.00048   0.00337   0.21683   1.91930
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  26.640390   3.913742   6.807 9.97e-12 ***
## zn            0.036103   0.024110   1.497  0.13429
## indus        -0.130796   0.076707  -1.705  0.08817 .
## nox         -21.785987   4.375610  -4.979 6.39e-07 ***
## age          -0.022291   0.010005  -2.228  0.02589 *
## rad          -0.923183   0.173337  -5.326 1.00e-07 ***
## tax           0.010952   0.003878   2.824  0.00475 **
## ptratio      -0.615936   0.153166  -4.021 5.79e-05 ***
## lstat         0.072792   0.045645   1.595  0.11077
## ---
## 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: 208.05  on 457  degrees of freedom
```

```
## AIC: 226.05
##
## Number of Fisher Scoring iterations: 9
```

Model 3 - Forward Selection

```
# Create an empty model with no variables
emptyMod <- glm(target ~ 1, data = train_clean, family = binomial)

logMod3 <- emptyMod %>%
  stepAIC(direction = "forward",
          scope = ~ zn + indus + chas + nox + rm + age + dis
                    + rad + tax + ptratio + lstat + medv,
          trace = FALSE)
summary(logMod3)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + ptratio + chas + tax + indus +
##     age + lstat + zn, family = binomial, data = train_clean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.04415  -0.00066   0.00376   0.22440   1.90815
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  27.462810   4.041221   6.796 1.08e-11 ***
## nox         -22.348501   4.437636  -5.036 4.75e-07 ***
## rad          -0.881928   0.174844  -5.044 4.56e-07 ***
## ptratio      -0.650968   0.159068  -4.092 4.27e-05 ***
## chasY        -0.918586   0.741449  -1.239   0.2154
## tax           0.009738   0.004016   2.425   0.0153 *
## indus        -0.105532   0.078689  -1.341   0.1799
## age          -0.020504   0.010055  -2.039   0.0414 *
## lstat         0.079005   0.046289   1.707   0.0879 .
## zn            0.034177   0.024054   1.421   0.1554
## ---
## 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: 206.53  on 456  degrees of freedom
## AIC: 226.53
##
## Number of Fisher Scoring iterations: 9
```

4. SELECT MODELS

Decide on the criteria for selecting the best binary logistic regression model. Will you select models with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your

models.

For the binary logistic regression model, will you use a metric such as log likelihood, AIC, ROC curve, etc.? Using the training data set, evaluate the binary logistic regression model based on (a) accuracy, (b) classification error rate, (c) precision, (d) sensitivity, (e) specificity, (f) F1 score, (g) AUC, and (h) confusion matrix. Make predictions using the evaluation data set.

```
formula(logMod1) # Model 1 formula
```

```
## target ~ nox + age + rad + tax + ptratio + lstat
```

```
formula(logMod2) # Model 2 formula
```

```
## target ~ zn + indus + nox + age + rad + tax + ptratio + lstat
```

```
formula(logMod3) # Model 3 formula
```

```
## target ~ nox + rad + ptratio + chas + tax + indus + age + lstat +
##     zn
```