ISLR Chapter 4, Ex 13
7/20/15
Tatiana Romanchishina

13. Using the Boston data set, fit classification models in order to predict whether a given suburb has a crime rate above or below the median. Explore logistic regression, LDA, and KNN models using various subsets of the predictors. Describe your findings.

First I added a new binary variable that indicates whether the crime rate is above or below the median. Then I standardized the other predictors, so that their means are 0 and standard deviation is 1. Then I assigned 100 observations to a test set and the remaining 406 to the training set. (see the R script)

Now I am ready to fit different models.

- **Logistic Regression**

First I try all the predictors:

```
============================================
                   Dependent variable:
                   ---------------------------
                             crimbin
-------------------------------------------
zn                           -1.328*
                                      (0.719)

indus                        0.208
                                      (0.354)

chas                         0.040
                                      (0.191)

nox                          4.344***
                                      (0.887)

rm                           -0.102
                                      (0.585)

age                          0.425
                                      (0.390)

dis                          1.374**
                                      (0.542)
```

```
rad                        6.542***
                           (1.636)

tax                        -0.971**
                           (0.450)

ptratio               0.021
                           (0.319)

black                -0.454
                           (0.513)

lstat                 0.517
                           (0.435)

medv                   1.291*
                           (0.692)

Constant            3.020***
                           (0.817)

-------------------------------------------
Observations               406
Log Likelihood             -83.782
Akaike Inf. Crit.          195.564
===========================================
Note:              *p<0.1; **p<0.05; ***p<0.01
```

It looks like nox, dis, rad, and tax have a significant relationship with the crime rate.

Now I am going to try to make predictions about my test set but I will try several thresholds:
(lr.pred are the predictions of the model with the specified threshold)

- ➔ probability of crime rate being above median = 50%:

  ```
              testCrim
  lr.pred       0   1
       0       73  21
       1        5   1
  ```

  with the error rate = 0.15.

  Even though it looks like the error rate is low, it only predicts well if a suburb has a crime rate below median, but it does not predict correctly otherwise.

➔ probability of crime rate being above median = 75%:

```
            testCrim
lr.pred      0    1
   0        76   22
   1         2    0
```

with the error rate = 0.24.

The error rate is even higher than before and now it cannot predict the suburbs with crime rate above median at all.

➔ probability of crime rate being above median = 25%

```
             testCrim
lr.pred        0    1
    0         66    3
    1         12   19
```

with the error rate = 0.15.

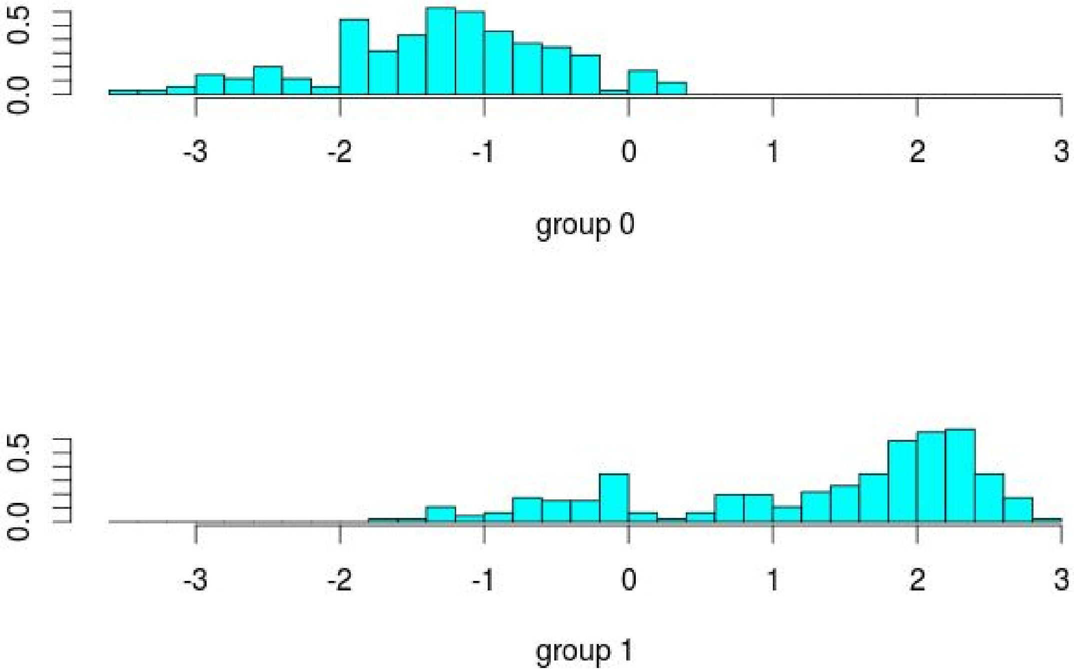
The error rate is as low as in the first try, but the predictions are looking better, because now it predicts most of both classes.

- **LDA**

1) All predictors:

```
Call: lda(crimbin ~ ., data = trainBoston)
Prior probabilities of groups:
        0           1
0.4310345   0.5689655
Group means:
    … (omitted)

Coefficients of linear discriminants:
                LD1
zn      -0.28143254
indus   0.30673534
chas    -0.03874451
nox     0.70535833
rm      0.10665269
age     0.25878572
dis     0.16752656
rad     0.94919860
tax     -0.19917231
ptratio -0.26897926
black   -0.03284464
lstat   0.12984063
medv    0.32316451
```



Now I can try to predict the test class using this model:

```
 lda.class    0   1
       0     76  22
       1      2   0
```

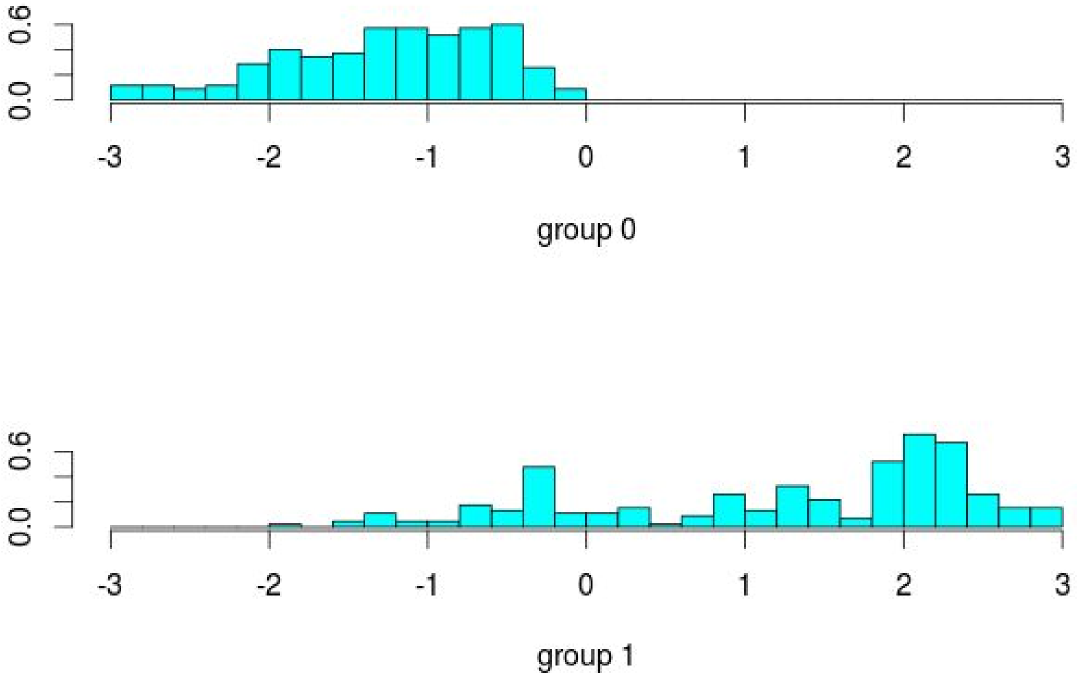
with the error rate = 0.24. Though it is a low error rate, it does not predict the suburbs with a crime rate above the median.

2) Some of the predictors: nox, dis, rad, tax, ptratio, medv

```
Call: lda(crimbin ~ nox + dis + rad + tax + ptratio + medv, data = trainBoston)
Prior probabilities of groups:
        0         1
0.4310345 0.5689655
Group means:
        nox        dis          rad          tax        ptratio         medv
0 -0.6515262  0.5396006 -0.6084553 -0.5520955 -0.2803591  0.3142349
1  0.8050526 -0.6929524  0.7388406  0.7232653  0.1655096 -0.2275223

Coefficients of linear discriminants:
              LD1
nox     0.89150541
dis     -0.27457496
```

```
rad       0.83731211
tax       -0.06205007
ptratio -0.14532932
medv   0.22763105
```



Now I can try to predict whether the crime rate is above the median or below:

```
                 testCrim
lda.class2      0     1
         0           78   22
         1             0     0
```

with the error rate = 0.22. Again the model does not predict correctly the suburbs that have a crime rate above the median, but it predicted correctly the suburbs that have a crime rate below the median.

- **KNN**
    - **All predictors**

Amazingly KNN model with all predictors and k=1 predicted correctly almost all instances of both classes:

```
           testCrim
knn.pred    0    1
       0   77    0
       1    1   22
```
with the accuracy = 98%.

Just to confirm whether this is random or not, I am going to use another test and train sets, k=1. The results are less exciting, but still high accuracy:

```
              testCrim2
knn.pred2      0    1
        0     51   27
        1      4   19
```
with the accuracy = 69%.

Now I will try using the same sets for training and testing, but try using different values of k:

- ➔ k = 3

```
             testCrim2
knn.pred2     0    1
         0   53   17
         1    2   29
```
with the accuracy = 81%

- ➔ k = 10

```
            testCrim2
knn.pred2    0    1
         0  49   11
         1   6   35
```
with the accuracy = 83%

- ➔ k = 20

```
            testCrim2
knn.pred2   0    1
         0 47    1
         1   8  45
```
with the accuracy = 91%

It appears that increasing the value of k increases the accuracy of the model. However, the first result must have been due to “lucky” sampling. Though as one can see using a different set of training/testing data did not make the result much worse.

The accuracy of the KNN model appears to be much higher than the accuracy of the previous two models.