Annand Module 03 Lab 01

Joseph Annand

2023-11-12

Load libraries

```
library(ISLR2)
library(MASS)
```

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

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.2
```

```
library(class)
```

Question 13

```
weekly <- Weekly
View(weekly)
```

Part A

```
# Get names of columns in weekly dataset
names(weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

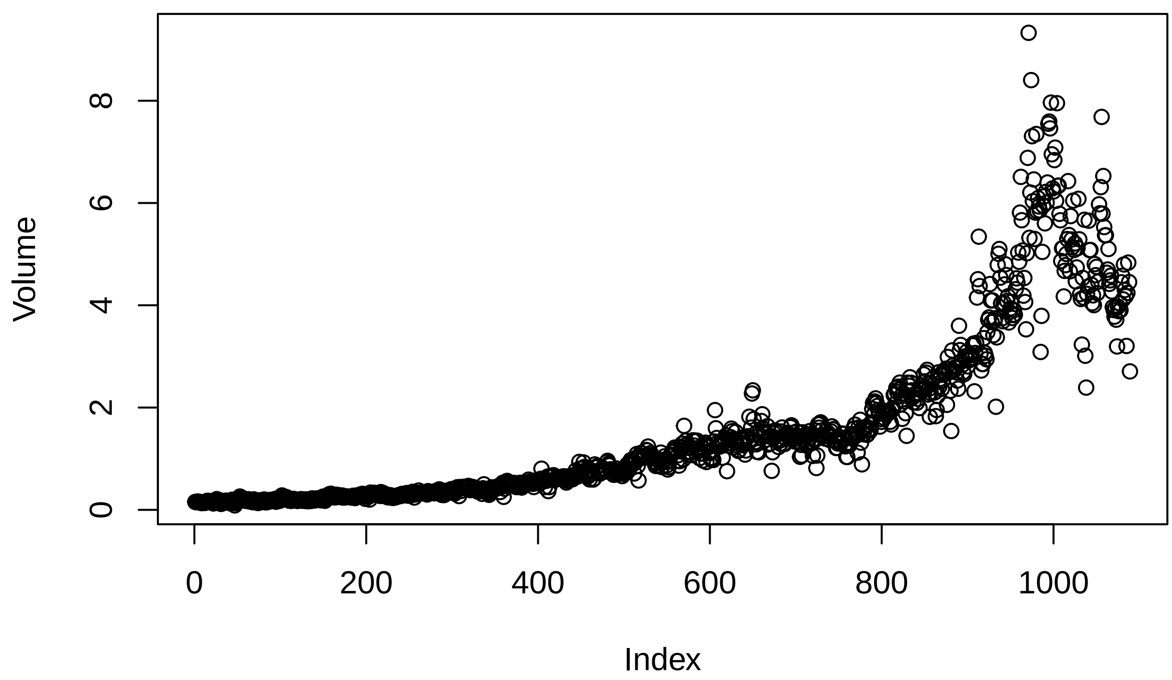
```
# get dimensions of the Weekly dataset
dim(weekly)
```

```
## [1] 1089    9
```

```r
# Produce summary statistics for each column
summary(weekly)
```

```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

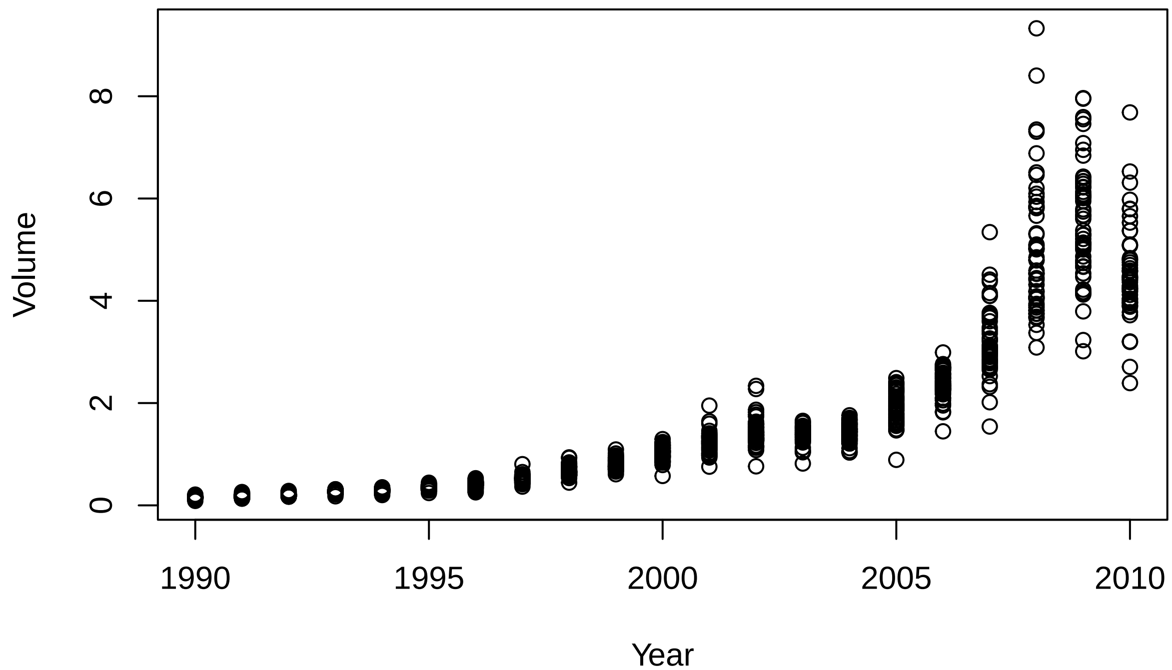
```r
# Get correlation matrix for weekly data
weekly_cor <- cor(weekly[, -9])
View(weekly_cor)

attach(weekly)
plot(Volume)
```




```r
attach(weekly)
```

```
## The following objects are masked from weekly (pos = 3):
##
##     Direction, Lag1, Lag2, Lag3, Lag4, Lag5, Today, Volume, Year
```

```r
plot(Year, Volume)
```



The only two variables that show a strong correlation are Volume and Year. All other variables are not strongly correlated to each other. The columns that represent the percentage returns for a given day all have the same Min and Max values. Distributions and means are very similar, as well.

Part B

```
# Create logistic regression with Direction as target and all other variables as
# predictors
glm.weekly <- glm(
  Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
  data = weekly, family = binomial
)

summary(glm.weekly)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
```

```
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Lag2 appears to be statistically significant because it has a p-value < 0.05.

Part C

```r
weekly.probs <- predict(glm.weekly, type = "response")
weekly.pred <- rep("Down", 1089)
weekly.pred[weekly.probs > 0.5] = "Up"

table(weekly.pred, Direction)
```

```
##            Direction
## weekly.pred Down  Up
##        Down   54  48
##        Up    430 557
```

```r
mean(weekly.pred == Direction)
```

```
## [1] 0.5610652
```

The confusion matrix tells us that our logistic regression model does well at predicting when the market will go up; however, it is majorly inaccurate in predicting when the market will go down. Overall, the model correctly predicts the movement of the market 56.1% of the time.

Part D

```r
# Create a set of test data from the weekly dataset that includes observations
# from 2009 to 2010.
train <- (Year < 2009)
weekly.2008 <- weekly[!train, ]
dim(weekly.2008)
```

```
## [1] 104   9
```

```r
Direction.2008 <- Direction[!train]

glm.train <- glm(
  Direction ~ Lag2, data = weekly, family = binomial, subset = train)
test.probs <- predict(glm.train, weekly.2008, type = "response")
test.pred <- rep("Down", 104)
test.pred[test.probs > 0.5] <- "Up"

table(test.pred, Direction.2008)
```

```
##          Direction.2008
## test.pred Down Up
##      Down    9  5
##      Up     34 56
```

```r
mean(test.pred == Direction.2008)
```

```
## [1] 0.625
```

Using only Lag2 as a predictor, the logistic model correctly predicts the direction of the market 62.5% of the time on the test data.

Part E

```r
lda.fit <- lda(Direction ~ Lag2, data = weekly, subset = train)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag2, data = weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

```r
lda.pred <- predict(lda.fit, weekly.2008)
lda.class <- lda.pred$class
table(lda.class, Direction.2008)
```

```
##          Direction.2008
## lda.class Down Up
##      Down    9  5
##      Up     34 56
```

```r
mean(lda.class == Direction.2008)
```

```
## [1] 0.625
```

Linear discriminant analysis correctly predicts the direction of the market 62.5% of the time.

Part F

```r
qda.fit <- qda(Direction ~ Lag2, data = weekly, subset = train)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag2, data = weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```r
qda.class <- predict(qda.fit, weekly.2008)$class
table(qda.class, Direction.2008)
```

```
##          Direction.2008
## qda.class Down Up
##      Down    0  0
##      Up     43 61
```

```r
mean(qda.class == Direction.2008)
```

```
## [1] 0.5865385
```

Quadratic discriminant analysis correctly predicts the direction of the market 58.7% of the time; however, it incorrectly predicts when the market goes down in every test case.

Part G

```r
# Create matrix of predictors for train and test data and vector of responses for
# training data
train.X <- as.matrix(weekly[train, ]$Lag2)
test.X <- as.matrix(weekly.2008$Lag2)
train.Direction <- Direction[train]

set.seed(1)
knn.pred <- knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, Direction.2008)
```

```
##         Direction.2008
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

```
mean(knn.pred == Direction.2008)
```

```
## [1] 0.5
```

KNN when k = 1 predicts the direction of the market correctly 50% of the time.

Part H

```
nb.fit <- naiveBayes(Direction ~ Lag2, data = weekly, subset = train)
nb.fit
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##      Down        Up
## 0.4477157 0.5522843
##
## Conditional probabilities:
##       Lag2
## Y             [,1]     [,2]
##   Down -0.03568254 2.199504
##   Up    0.26036581 2.317485
```

```
nb.class <- predict(nb.fit, weekly.2008)
table(nb.class, Direction.2008)
```

```
##         Direction.2008
## nb.class Down Up
##     Down    0  0
##     Up     43 61
```

```
mean(nb.class == Direction.2008)
```

```
## [1] 0.5865385
```

Naive Bayes correctly predicts the direction of the market 58.7% of the time; however, it incorrectly predicts when the market goes down in every test case.

Part I

```r
results <- data.frame(algorithm = c("Logistic", "LDA", "QDA", "KNN", "Naive Bayes"),
                      accuracy = c(mean(test.pred == Direction.2008),
                                   mean(lda.class == Direction.2008),
                                   mean(qda.class == Direction.2008),
                                   mean(knn.pred == Direction.2008),
                                   mean(nb.class == Direction.2008)))

results
```

```
##     algorithm  accuracy
## 1    Logistic 0.6250000
## 2         LDA 0.6250000
## 3         QDA 0.5865385
## 4         KNN 0.5000000
## 5 Naive Bayes 0.5865385
```

Logistic regression and LDA provide the best results.

Part J

```r
# Evaluate logistic regression using Lag 1 and Lag2

glm.j <- glm(
  Direction ~ Lag1 + Lag2, data = weekly, family = binomial, subset = train)
test.probs.j <- predict(glm.train, weekly.2008, type = "response")
test.pred.j <- rep("Down", 104)
test.pred.j[test.probs.j > 0.5] <- "Up"

table(test.pred.j, Direction.2008)
```

```
##            Direction.2008
## test.pred.j Down Up
##        Down    9  5
##        Up     34 56
```

```r
mean(test.pred.j == Direction.2008)
```

```
## [1] 0.625
```

The addition of Lag2 as a predictor in the model does not change the accuracy of the model.

```r
# Evaluate logistic regression with the interaction between Lag1 and Lag2

glm.int.j<- glm(
  Direction ~ Lag1 * Lag2 * Lag3, data = weekly, family = binomial, subset = train)
test.probs.int.j <- predict(glm.train, weekly.2008, type = "response")
test.pred.int.j <- rep("Down", 104)
test.pred.int.j[test.probs.j > 0.5] <- "Up"

table(test.pred.int.j, Direction.2008)
```

```
##                 Direction.2008
## test.pred.int.j Down Up
##            Down    9  5
##            Up     34 56
```

```r
mean(test.pred.int.j == Direction.2008)
```

```
## [1] 0.625
```

Modeling the interaction of Lag1, Lag2, and Lag3 does not change the accuracy of predictions.

```r
# Evaluate LDA and QDA using x^2 transformation

lda.fit.j <- lda(Direction ~ I(Lag2^2), data = weekly, subset = train)
lda.fit.j
```

```
## Call:
## lda(Direction ~ I(Lag2^2), data = weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##      I(Lag2^2)
## Down  4.828121
## Up    5.428657
##
## Coefficients of linear discriminants:
##                  LD1
## I(Lag2^2) 0.06583971
```

```r
lda.pred.j <- predict(lda.fit.j, weekly.2008)
lda.class.j <- lda.pred.j$class
table(lda.class.j, Direction.2008)
```

```
##            Direction.2008
## lda.class.j Down Up
##        Down    0  0
##        Up     43 61
```

```r
mean(lda.class.j == Direction.2008)
```

```
## [1] 0.5865385
```

```r
qda.fit.j <- qda(Direction ~ I(Lag2^2), data = weekly, subset = train)
qda.fit.j
```

```
## Call:
## qda(Direction ~ I(Lag2^2), data = weekly, subset = train)
```

```
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##      I(Lag2^2)
## Down  4.828121
## Up    5.428657
```

```r
qda.class.j <- predict(qda.fit.j, weekly.2008)$class
table(qda.class.j, Direction.2008)
```

```
##            Direction.2008
## qda.class.j Down Up
##        Down    4  6
##        Up     39 55
```

```r
mean(qda.class.j == Direction.2008)
```

```
## [1] 0.5673077
```

An x^2 transformation of Lag2 weakens the accuracy of the LDA and QDA model.

```r
# KNN using k = 5 and k = 10

set.seed(4)
knn.pred.j <- knn(train.X, test.X, train.Direction, k = 5)
table(knn.pred.j, Direction.2008)
```

```
##           Direction.2008
## knn.pred.j Down Up
##       Down   16 21
##       Up     27 40
```

```r
mean(knn.pred.j == Direction.2008)
```

```
## [1] 0.5384615
```

```r
set.seed(5)
knn.pred.j2 <- knn(train.X, test.X, train.Direction, k = 10)
table(knn.pred.j2, Direction.2008)
```

```
##            Direction.2008
## knn.pred.j2 Down Up
##        Down   17 20
##        Up     26 41
```

```r
mean(knn.pred.j2 == Direction.2008)
```

```
## [1] 0.5576923
```

Accuracy of KNN model increases when k=5 and increases more when k = 10.

Question 16

Investigate Boston dataset

```r
boston <- Boston
summary(boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

```r
cor_boston <- cor(boston)
View(cor_boston)
```

```r
# Create qualitative variable that indicates if crim value is above or below median
crim_median <- median(boston$crim)
boston$mcrim <- apply(boston, 1, FUN = function(x) if(x[1] > crim_median) "above"
                      else "below")
boston$mcrim <- as.factor(boston$mcrim)
```

Create training and test data from Boston dataset

```
# Create training and test sets

smp_size <- floor(0.75 * nrow(boston))

set.seed(2)
train.ind <- sample(seq_len(nrow(boston)), replace = F, size = smp_size)

train.bos <- boston[train.ind, ]
test.bos <- boston[-train.ind, ]

mcrim.test <- as.factor(boston$mcrim[-train.ind])
```

Logistic Regression

We first create a logistic regression model using the variables most associated with crime rate as the predictors.

```
# Logistic regression using rad, tax, lstat, indus, black, and medv

glm.boston <- glm(mcrim ~ rad + tax + lstat + indus + black + medv,
                  data = boston, family = binomial, subset = train.ind)

summary(glm.boston)
```

```
##
## Call:
## glm(formula = mcrim ~ rad + tax + lstat + indus + black + medv,
##     family = binomial, data = boston, subset = train.ind)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.320367   2.418384   1.373 0.169762
## rad         -0.557427   0.123586  -4.510 6.47e-06 ***
## tax          0.002759   0.002383   1.158 0.247019
## lstat       -0.136663   0.037384  -3.656 0.000257 ***
## indus       -0.140871   0.035411  -3.978 6.95e-05 ***
## black        0.011508   0.005201   2.213 0.026921 *
## medv        -0.093802   0.025274  -3.711 0.000206 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 525.19  on 378  degrees of freedom
## Residual deviance: 252.72  on 372  degrees of freedom
## AIC: 266.72
##
## Number of Fisher Scoring iterations: 8
```

```
blog.probs <- predict(glm.boston, test.bos, type = "response")
contrasts(boston$mcrim)
```

```
##       below
## above     0
## below     1
```

```
blog.pred <- rep("below", length(test.bos))
blog.pred[blog.probs < 0.5] <- "above"

table(blog.pred, mcrim.test)
```

```
##          mcrim.test
## blog.pred above below
##     above    53     7
##     below     8     6
```

```
(53 + 6) / (53 + 8 + 7 + 6)
```

```
## [1] 0.7972973
```

Removing the least significant variable, tax, does not improve the results of the logistic regression

```
# Logistic regression using rad, lstat, indus, black, and medv

glm.boston2 <- glm(mcrim ~ rad + lstat + indus + black + medv,
                  data = boston, family = binomial, subset = train.ind)

summary(glm.boston2)
```

```
##
## Call:
## glm(formula = mcrim ~ rad + lstat + indus + black + medv, family = binomial,
##     data = boston, subset = train.ind)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.916257   2.380202   1.645 0.099898 .
## rad         -0.507878   0.113221  -4.486 7.27e-06 ***
## lstat       -0.139782   0.037346  -3.743 0.000182 ***
## indus       -0.118121   0.028763  -4.107 4.01e-05 ***
## black        0.011432   0.005256   2.175 0.029615 *
## medv        -0.098084   0.025006  -3.922 8.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 525.19  on 378  degrees of freedom
## Residual deviance: 254.09  on 373  degrees of freedom
## AIC: 266.09
##
## Number of Fisher Scoring iterations: 8
```

```r
blog2.probs <- predict(glm.boston2, test.bos, type = "response")
blog2.pred <- rep("below", length(test.bos))
blog2.pred[blog.probs < 0.5] <- "above"

table(blog2.pred, mcrim.test)
```

```
##           mcrim.test
## blog2.pred above below
##      above    53     7
##      below     8     6
```

Linear Discriminant Analysis

LDA model has slightly better accuracy than logistic regression using the same set of predictors.

```r
# LDA using rad, lstat, indus, black, and medv

lda.boston <- lda(mcrim ~ rad + lstat + indus + black + medv,
                  data = boston, subset = train.ind)

lda_bos.pred <- predict(lda.boston, test.bos)
lda_bos.class <- lda_bos.pred$class
table(lda_bos.class, mcrim.test)
```

```
##              mcrim.test
## lda_bos.class above below
##         above    46     4
##         below    22    55
```

```r
(46 + 55) / (46 + 55 + 22 + 4)
```

```
## [1] 0.7952756
```

We create an LDA model using just lstat and medv which were variables of interest when we analyzed Boston dataset in a previous homework assignment. We see the accuracy of the model gets worse.

```r
# LDA using just lstat and medv
lda.boston2 <- lda(mcrim ~ lstat + medv,
                  data = boston, subset = train.ind)

lda_bos2.pred <- predict(lda.boston2, test.bos)
lda_bos2.class <- lda_bos2.pred$class
table(lda_bos2.class, mcrim.test)
```

```
##               mcrim.test
## lda_bos2.class above below
##          above    39    13
##          below    29    46
```

```r
(39 + 46) / (39 + 46 + 29 + 13)
```

```
## [1] 0.6692913
```

Naive Bayes

Naive Bayes returns similar results to logistic regression and LDA but is slightly worse.

```r
# Naive Bayes using rad, lstat, indus, black, and medv

nb_bos.fit <- naiveBayes(mcrim ~ rad + lstat + indus + black + medv,
                         data = boston, subset = train.ind)

nb_bos.class <- predict(nb_bos.fit, test.bos)
table(nb_bos.class, mcrim.test)
```

```
##             mcrim.test
## nb_bos.class above below
##        above    42     3
##        below    26    56
```

```r
(42 + 56) / (42 + 56 + 26 + 3)
```

```
## [1] 0.7716535
```

Naive Bayes using just lstat and medv as predictors is considerably worse at predicting the test data.

```r
# Naive Bayes using just lstat and medv

nb_bos2.fit <- naiveBayes(mcrim ~ lstat + medv,
                         data = boston, subset = train.ind)

nb_bos2.class <- predict(nb_bos2.fit, test.bos)
table(nb_bos2.class, mcrim.test)
```

```
##              mcrim.test
## nb_bos2.class above below
##         above    41    11
##         below    27    48
```

```r
(41 + 48) / (41 + 48 + 27 + 11)
```

```
## [1] 0.7007874
```

K-Nearest Neighbor

KNN yields the best results at predicting the crime rate across all classifiers tested. The best model uses rad, lstat, indus, black, and medv variables as predictors. Increasing the value of k form 1 to 3 does not affect the accuracy of the model.

```r
# KNN using rad, lstat, indus, black, and medv

bos.train <- cbind(boston$rad, boston$lstat, boston$indus, boston$black, boston$medv)[train.ind, ]
bos.test <- cbind(boston$rad, boston$lstat, boston$indus, boston$black, boston$medv)[-train.ind, ]
bos.mcrim <- boston$mcrim[train.ind]

set.seed(3)
knn_bos.pred <- knn(bos.train, bos.test, bos.mcrim, k = 1)
table(knn_bos.pred, mcrim.test)
```

```
##             mcrim.test
## knn_bos.pred above below
##        above    58    12
##        below    10    47
```

```r
(58 + 47) / (58 + 47 + 10 + 12)
```

```
## [1] 0.8267717
```

```r
# KNN using rad, lstat, indus, black, and medv k = 3

set.seed(3)
knn_bos2.pred <- knn(bos.train, bos.test, bos.mcrim, k = 3)
table(knn_bos2.pred, mcrim.test)
```

```
##              mcrim.test
## knn_bos2.pred above below
##         above    56    10
##         below    12    49
```

```r
(56 + 49) / (56 + 49 + 12 + 10)
```

```
## [1] 0.8267717
```

```r
# KNN using lstat and medv

bos2.train <- cbind(boston$lstat, boston$medv)[train.ind, ]
bos2.test <- cbind(boston$lstat, boston$medv)[-train.ind, ]

set.seed(3)
knn_bos3.pred <- knn(bos2.train, bos2.test, bos.mcrim, k = 3)
table(knn_bos3.pred, mcrim.test)
```

```
##              mcrim.test
## knn_bos3.pred above below
##         above    45    19
##         below    23    40
```

```r
(45 + 40) / (45 + 40 + 23 + 19)
```

```
## [1] 0.6692913
```