Predictive Modeling Exercises Week 2

Amol Gote

7/18/2020

# Problem 10

names(Weekly)

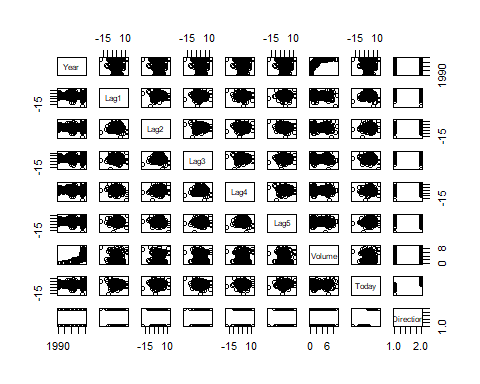
## [1] "Year" "Lag1" "Lag2" "Lag3" "Lag4" "Lag5"   
## [7] "Volume" "Today" "Direction"

# 10a - Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

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   
##   
##   
##   
##

pairs(Weekly)



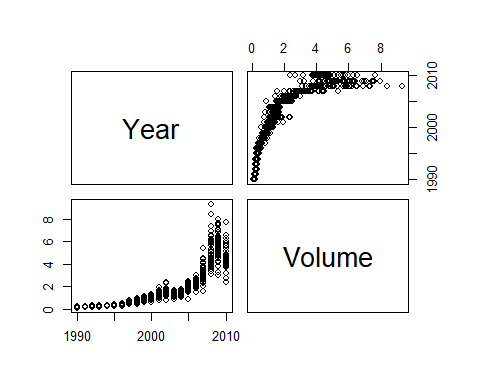
weekly\_dataset <- select (Weekly,-c(Direction))  
cor(weekly\_dataset)

## Year Lag1 Lag2 Lag3 Lag4  
## Year 1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923  
## Lag1 -0.03228927 1.000000000 -0.07485305 0.05863568 -0.071273876  
## Lag2 -0.03339001 -0.074853051 1.00000000 -0.07572091 0.058381535  
## Lag3 -0.03000649 0.058635682 -0.07572091 1.00000000 -0.075395865  
## Lag4 -0.03112792 -0.071273876 0.05838153 -0.07539587 1.000000000  
## Lag5 -0.03051910 -0.008183096 -0.07249948 0.06065717 -0.075675027  
## Volume 0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617  
## Today -0.03245989 -0.075031842 0.05916672 -0.07124364 -0.007825873  
## Lag5 Volume Today  
## Year -0.030519101 0.84194162 -0.032459894  
## Lag1 -0.008183096 -0.06495131 -0.075031842  
## Lag2 -0.072499482 -0.08551314 0.059166717  
## Lag3 0.060657175 -0.06928771 -0.071243639  
## Lag4 -0.075675027 -0.06107462 -0.007825873  
## Lag5 1.000000000 -0.05851741 0.011012698  
## Volume -0.058517414 1.00000000 -0.033077783  
## Today 0.011012698 -0.03307778 1.000000000

weekly\_dataset <- Weekly

1. Only relationship which is clearly visible is between Year and volume

pairs(Year ~ Volume, weekly\_dataset)



# 10b - Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so,which ones?

logistic\_regression <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = binomial)  
summary(logistic\_regression)

##   
## Call:  
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +   
## Volume, family = binomial, data = Weekly)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6949 -1.2565 0.9913 1.0849 1.4579   
##   
## 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

1. Lag2 appears to be statistically significant as the p-value is less than 0.05.

# 10c Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

predictions <- predict(logistic\_regression, Weekly, type="response")  
predicted\_direction <- as.factor(ifelse(predictions < 0.5, "Down", "Up"))  
confusion\_matrix <- table(predicted\_direction,   
 Weekly$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 54 48  
## Up 430 557

error\_rate <- mean(predicted\_direction != Weekly$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 43.89348 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 56.10652 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 88.84298 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 7.933884 %

# 10d Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

made by logistic regression.

weekly\_dataset\_lt\_2009 <- (Weekly$Year < 2009)  
train\_dataset <- Weekly[weekly\_dataset\_lt\_2009, ]  
test\_dataset <- Weekly[!weekly\_dataset\_lt\_2009, ]  
  
logistic\_regression <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = weekly\_dataset\_lt\_2009)  
predictions <- predict(logistic\_regression, test\_dataset, type="response")  
predicted\_direction <- as.factor(ifelse(predictions < 0.5, "Down", "Up"))  
confusion\_matrix <- table(predicted\_direction,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 9 5  
## Up 34 56

error\_rate <- mean(predicted\_direction != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 37.5 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 62.5 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 79.06977 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 8.196721 %

# 10e Repeat (d) using LDA.

lda\_model <- lda(Direction ~ Lag2, data = Weekly, subset = weekly\_dataset\_lt\_2009)  
predictions <- predict(lda\_model, test\_dataset, type="response")  
confusion\_matrix <- table(predictions$class,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 9 5  
## Up 34 56

error\_rate <- mean(predictions$class != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 37.5 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 62.5 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 79.06977 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 8.196721 %

# 10f Repeat (d) using QDA.

qda\_model <- qda(Direction ~ Lag2, data = Weekly, subset = weekly\_dataset\_lt\_2009)  
predictions <- predict(qda\_model, test\_dataset, type="response")  
confusion\_matrix <- table(predictions$class,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 0 0  
## Up 43 61

error\_rate <- mean(predictions$class != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 41.34615 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 58.65385 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 100 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 0 %

# 10g Repeat (d) using KNN with K = 1.

train\_dataset\_matrix <- as.matrix(train\_dataset$Lag2)  
test\_dataset\_matrix <- as.matrix(test\_dataset$Lag2)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$Direction, 1)  
confusion\_matrix <- table(predictions,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 21 30  
## Up 22 31

error\_rate <- mean(predictions != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 50 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 50 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 51.16279 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 49.18033 %

# 10h Which of these methods appears to provide the best results on this data?

1. Logistic regression and LDA methods have similar and lowest error rates, followed by QDA and then KNN (K=1).

#10i Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

1. Logistic Regression

logistic\_regression <- glm(Direction ~ Lag1 \* Lag2, data = Weekly, family = binomial, subset = weekly\_dataset\_lt\_2009)  
predictions <- predict(logistic\_regression, test\_dataset, type="response")  
predicted\_direction <- as.factor(ifelse(predictions < 0.5, "Down", "Up"))  
confusion\_matrix <- table(predicted\_direction,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 7 8  
## Up 36 53

error\_rate <- mean(predicted\_direction != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 42.30769 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 57.69231 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 83.72093 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 13.11475 %

1. LDA

lda\_model <- lda(Direction ~ Lag1 \* Lag2, data = Weekly, subset = weekly\_dataset\_lt\_2009)  
predictions <- predict(lda\_model, test\_dataset, type="response")  
confusion\_matrix <- table(predictions$class,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 7 8  
## Up 36 53

error\_rate <- mean(predictions$class != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 42.30769 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 57.69231 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 83.72093 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 13.11475 %

1. QDA

qda\_model <- qda(Direction ~ Lag1 \* Lag2, data = Weekly, subset = weekly\_dataset\_lt\_2009)  
predictions <- predict(qda\_model, test\_dataset, type="response")  
confusion\_matrix <- table(predictions$class,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 23 36  
## Up 20 25

error\_rate <- mean(predictions$class != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 53.84615 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 46.15385 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 46.51163 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 59.01639 %

1. KNN (n = 10)

train\_dataset\_matrix <- as.matrix(train\_dataset$Lag2)  
test\_dataset\_matrix <- as.matrix(test\_dataset$Lag2)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$Direction, 10)  
confusion\_matrix <- table(predictions,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 18 19  
## Up 25 42

error\_rate <- mean(predictions != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 42.30769 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 57.69231 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 58.13953 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 31.14754 %

1. KNN (n = 50)

train\_dataset\_matrix <- as.matrix(train\_dataset$Lag2)  
test\_dataset\_matrix <- as.matrix(test\_dataset$Lag2)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$Direction, 50)  
confusion\_matrix <- table(predictions,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 21 24  
## Up 22 37

error\_rate <- mean(predictions != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 44.23077 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 55.76923 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 51.16279 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 39.34426 %

1. KNN (n = 100)

train\_dataset\_matrix <- as.matrix(train\_dataset$Lag2)  
test\_dataset\_matrix <- as.matrix(test\_dataset$Lag2)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$Direction, 100)  
confusion\_matrix <- table(predictions,   
 test\_dataset$Direction,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status Down Up  
## Down 9 12  
## Up 34 49

error\_rate <- mean(predictions != test\_dataset$Direction)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 44.23077 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 55.76923 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 79.06977 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 19.67213 %

**Out of all the combinations Logistic Regression and LDA have the best error rate performance (low error rate).**

# Problem 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.

# 10a Boston dataset info

names(Boston)

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

1. Create datasets

Boston$crime\_rate\_above\_median <- 0  
Boston$crime\_rate\_above\_median[Boston$crim > median(Boston$crim)] <- 1  
Boston$crime\_rate\_above\_median <-factor(Boston$crime\_rate\_above\_median)  
data\_partition <- createDataPartition(y = Boston$crime\_rate\_above\_median, p = 0.75, list = FALSE)  
train\_dataset <- Boston[data\_partition,]  
test\_dataset <- Boston[-data\_partition,]

1. Logistic Regresssion

logistic\_regression <- glm(crime\_rate\_above\_median ~ indus + nox + rad + tax + lstat, data = train\_dataset, family = binomial)  
predictions <- predict(logistic\_regression, test\_dataset, type="response")  
predicted\_direction <- as.factor(ifelse(predictions > 0.5, 1, 0))  
confusion\_matrix <- table(predicted\_direction,   
 test\_dataset$crime\_rate\_above\_median,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status 0 1  
## 0 59 9  
## 1 4 54

error\_rate <- mean(predicted\_direction != test\_dataset$crime\_rate\_above\_median)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 10.31746 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 89.68254 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 6.349206 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 14.28571 %

1. LDA

lda\_model <- lda(crime\_rate\_above\_median ~ indus + nox + rad + tax + lstat, data = train\_dataset)  
predictions <- predict(lda\_model, test\_dataset, type="response")  
confusion\_matrix <- table(predictions$class,   
 test\_dataset$crime\_rate\_above\_median,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status 0 1  
## 0 61 11  
## 1 2 52

error\_rate <- mean(predictions$class != test\_dataset$crime\_rate\_above\_median)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 10.31746 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 89.68254 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 3.174603 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 17.46032 %

1. KNN (n=1)

train\_dataset\_matrix <- as.matrix(train\_dataset$indus,train\_dataset$nox,train\_dataset$rad,train\_dataset$tax,train\_dataset$lstat)  
test\_dataset\_matrix <- as.matrix(test\_dataset$indus,test\_dataset$nox,test\_dataset$rad,test\_dataset$tax,test\_dataset$lstat)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$crime\_rate\_above\_median, 1)  
confusion\_matrix <- table(predictions,   
 test\_dataset$crime\_rate\_above\_median,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status 0 1  
## 0 61 1  
## 1 2 62

error\_rate <- mean(predictions != test\_dataset$crime\_rate\_above\_median)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 2.380952 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 97.61905 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 3.174603 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 1.587302 %

1. KNN (n=10)

train\_dataset\_matrix <- as.matrix(train\_dataset$indus,train\_dataset$nox,train\_dataset$rad,train\_dataset$tax,train\_dataset$lstat)  
test\_dataset\_matrix <- as.matrix(test\_dataset$indus,test\_dataset$nox,test\_dataset$rad,test\_dataset$tax,test\_dataset$lstat)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$crime\_rate\_above\_median, 10)  
confusion\_matrix <- table(predictions,   
 test\_dataset$crime\_rate\_above\_median,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status 0 1  
## 0 45 1  
## 1 18 62

error\_rate <- mean(predictions != test\_dataset$crime\_rate\_above\_median)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 15.07937 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 84.92063 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 28.57143 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 1.587302 %

1. KNN (n=100)

train\_dataset\_matrix <- as.matrix(train\_dataset$indus,train\_dataset$nox,train\_dataset$rad,train\_dataset$tax,train\_dataset$lstat)  
test\_dataset\_matrix <- as.matrix(test\_dataset$indus,test\_dataset$nox,test\_dataset$rad,test\_dataset$tax,test\_dataset$lstat)  
predictions <- knn(train\_dataset\_matrix, test\_dataset\_matrix, train\_dataset$crime\_rate\_above\_median, 100)  
confusion\_matrix <- table(predictions,   
 test\_dataset$crime\_rate\_above\_median,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status 0 1  
## 0 53 11  
## 1 10 52

error\_rate <- mean(predictions != test\_dataset$crime\_rate\_above\_median)  
correct\_prediction <- 1 - error\_rate  
false\_positive <- confusion\_matrix[2,1] / sum(confusion\_matrix[,1])  
false\_negative <- confusion\_matrix[1,2] / sum(confusion\_matrix[,2])  
cat("\n")

cat("Error rate or training error rate is:", error\_rate \* 100, "%\n")

## Error rate or training error rate is: 16.66667 %

cat("Correct prediction rate is: ", 100 \* (1 - error\_rate), "%\n")

## Correct prediction rate is: 83.33333 %

cat("False positive Rate: ", 100 \* false\_positive, "%\n")

## False positive Rate: 15.87302 %

cat("False negative Rate: ", 100 \* false\_negative, "%\n")

## False negative Rate: 17.46032 %

**With following sub predictors indus,nox,rad,tax,lstat, we got the lowest error rate with KNN with n = 1.**