Data Mining Assignment 4

1) Read Chapter 4 (all sections) and Chapter 5 (Sections 5.2, 5.5, 5.6 and 5.7).

2) Consider the following data set for a binary class problem.

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

Calculate the misclassification error rate when splitting on A and B to determine the best split. Which of these splits considered is the best according to misclassification error rate?

Splitting on A:

	A = T	**A = F**
+	**4**	**0**
-	**3**	**3**

Misclassification error of A:

Rate = 1 – [(TP + TN) / Total records]

= 1 – [(4 + 3) / 10]

= 3/10 = **0.3**

Splitting on B:

	B = T	**B = F**
+	**3**	**1**
-	**1**	**5**

Misclassification error of B:

Rate = 1 – [(TP + TN) / Total records]

= 1 – [(3 + 5) / 10]

= 2/10 = **0.2**

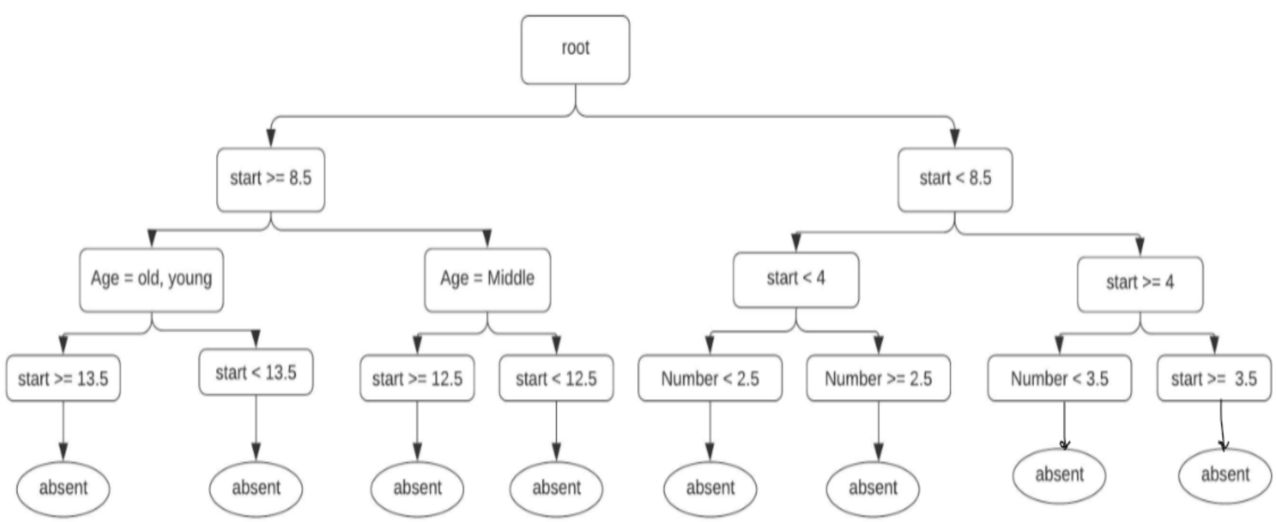
As the misclassification error of B is lower, hence the splitting the table on B should be considered as the best split.

3) Consider the training examples shown below for a binary classification problem.

Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

For a3, which is a continuous attribute compute misclassification error rate for every possible split to determine the best split. Which of these splits considered is the best according to misclassification error rate?

4) The file http://www-stat.wharton.upenn.edu/~dmease/rpart_text_example.txt gives an example of text output for a tree fit using the rpart() function in R from the library rpart. Use this tree to predict the class labels for the 10 observations in the test data http://www-stat.wharton.upenn.edu/~dmease/test_data.csv linked here. Do this manually - do not use R or any software.



Age	Number	Start	
Middle	5	10	present
Young	2	17	absent
Old	10	6	present
Old	4	15	absent
Middle	5	15	absent
Young	3	13	absent
Old	5	8	present
Young	7	8	absent
Middle	3	13	absent

5) I split the popular sonar data set into a training set (http://www-stat.wharton.upenn.edu/~dmease/sonar_train.csv) and a test set (http://www-stat.wharton.upenn.edu/~dmease/sonar_test.csv). Use R to compute the misclassification error rate on the test set when training on the training set for a tree of depth 5 using all the default values except control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5). Remember that the 61st column is the response and the other 60 columns are the predictors.

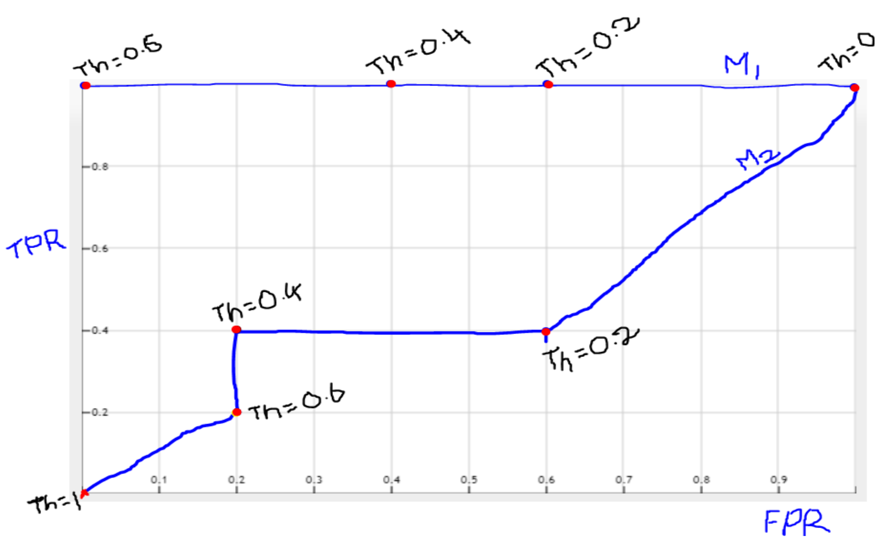
```
> sonar_fit <- rpart(y ~ ., x, control = rpart.control(minsplit=0,minbucket=0,cp=-1, ma
xcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5))
> plot(sonar_fit)
> text(sonar_fit)
> print(sonar_fit)
n= 130

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 130 64 -1 (0.50769231 0.49230769)
   2) V11>=0.17095 79 21 -1 (0.73417722 0.26582278)
     4) V27>=0.8191 37  2 -1 (0.94594595 0.05405405)
       8) V9>=0.0889 34  0 -1 (1.00000000 0.00000000) *
       9) V9< 0.0889 3  1 1 (0.33333333 0.66666667)
        18) V2< 0.0195 1  0 -1 (1.00000000 0.00000000) *
        19) V2>=0.0195 2  0 1 (0.00000000 1.00000000) *
     5) V27< 0.8191 42 19 -1 (0.54761905 0.45238095)
      10) V54>=0.02075 12  0 -1 (1.00000000 0.00000000) *
      11) V54< 0.02075 30 11 1 (0.36666667 0.63333333)
        22) V8< 0.17045 17  7 -1 (0.58823529 0.41176471)
          44) V36< 0.4491 12  2 -1 (0.83333333 0.16666667) *
          45) V36>=0.4491 5  0 1 (0.00000000 1.00000000) *
        23) V8>=0.17045 13  1 1 (0.07692308 0.92307692)
          46) V1>=0.0925 1  0 -1 (1.00000000 0.00000000) *
          47) V1< 0.0925 12  0 1 (0.00000000 1.00000000) *
   3) V11< 0.17095 51  8 1 (0.15686275 0.84313725)
     6) V52>=0.0209 6  1 -1 (0.83333333 0.16666667)
      12) V1>=0.02225 5  0 -1 (1.00000000 0.00000000) *
      13) V1< 0.02225 1  0 1 (0.00000000 1.00000000) *
     7) V52< 0.0209 45  3 1 (0.06666667 0.93333333)
      14) V19>=0.8351 5  2 -1 (0.60000000 0.40000000)
        28) V26< 0.6153 3  0 -1 (1.00000000 0.00000000) *
        29) V26>=0.6153 2  0 1 (0.00000000 1.00000000) *
      15) V19< 0.8351 40  0 1 (0.00000000 1.00000000) *
```

```
> predictions <- as.numeric(predict(sonar_fit, sonar_test, type = "class"))
> predictions <- replace(predictions, predictions == 1, -1)
> predictions <- replace(predictions, predictions == 2, 1)
> predictions
 [1] -1 -1 -1  1 -1 -1  1  1 -1 -1 -1  1  1  1 -1 -1 -1 -1  1  1 -1 -1  1 -1  1  1 -1
[28] -1 -1 -1 -1  1 -1  1  1  1  1  1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1 -1 -1 -1  1  1
[55] -1  1  1 -1  1  1  1 -1 -1  1 -1 -1  1 -1  1 -1  1  1 -1 -1  1  1  1  1
> actual_values <- sonar_test[, 61]
> actual_values
 [1] -1 -1 -1  1  1  1  1  1 -1 -1 -1  1 -1  1 -1 -1 -1  1  1  1 -1 -1  1 -1 -1  1 -1
[28] -1  1  1  1  1  1  1  1  1 -1  1 -1 -1 -1 -1  1 -1  1 -1  1  1  1 -1 -1 -1  1 -1
[55] -1 -1 -1 -1 -1 -1  1 -1 -1 -1 -1 -1  1 -1 -1 -1 -1  1 -1 -1  1  1 -1  1
```

```
> xtab <- table(predictions, actual_values)
```

```
> results <- confusionMatrix(xtab)
> results
Confusion Matrix and Statistics

           actual_values
predictions -1  1
         -1 33  8
         1  12 25

               Accuracy : 0.7436
                 95% CI : (0.6321, 0.83
    No Information Rate : 0.5769
    P-Value [Acc > NIR] : 0.001668

                  Kappa : 0.4831

 Mcnemar's Test P-Value : 0.502335

            Sensitivity : 0.7333
            Specificity : 0.7576
         Pos Pred Value : 0.8049
         Neg Pred Value : 0.6757
             Prevalence : 0.5769
         Detection Rate : 0.4231
   Detection Prevalence : 0.5256
      Balanced Accuracy : 0.7455

       'Positive' Class : -1

> mean(predictions != actual_values)
[1] 0.2564103
```

6) Do Chapter 5 textbook problem #17 (parts a and c only) on pages 322-323. Note that there is a typo in part c - it should read "Repeat the analysis for part (b)". We will do part b in class.



M1

Cut-off threshold = 0.5

TP = 1, 2, 5

TN = 3, 7, 8, 10

FN = 6.9

Precision (p) = TP / (TP + FP) ➔ 3 / (3+1) = 0.75

Recall (r) = TP / (TP + FN) ➔ 3 / (3+2) = 0.6

The harmonic mean between recall and precision is F1.

F1 = 2 / (1/r + 1/p) ➔ 2 / (1/0.6 + 1/0.75) = 0.9/1.35 = 0.667

M2

Cut-off threshold = 0.5

TP = 1

TN = 4, 7, 8, 10

FN = 2, 5, 6, 9

FP = 3

Precision (p) = TP / (TP + FP) ➔ 1 / (1+1) = 0.5

Recall (r) = TP / (TP + FN) ➔ 1 / (1+4) = 0.2

The harmonic mean between recall and precision is F1.

F1 = 2 / (1/r + 1/p) ➔ 2 / (1/0.2 + 1/0.5) = 0.2/0.7 = 0.2857

By comparing M1, M2; F1 is larger for M1 which indicates a better model.

7) Compute the misclassification error on the training data for the Random Forest classifier to the last column of the sonar training data. Show your R code for doing this.

```
> # Load the party package. It will automatically load other
> # required packages.
> library(party)
> library(randomForest)
>
> # Create the forest.
> output_fit <- randomForest(X.1 ~ .,
+                            data = sonar_training_data)
>
> # View the forest results.
> print(output_fit)

Call:
 randomForest(formula = X.1 ~ ., data = sonar_training_data)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 7

        OOB estimate of  error rate: 17.83%
Confusion matrix:
   -1  1 class.error
-1 56  9   0.1384615
1  14 50   0.2187500
```

Misclassification error rate = 1 – ((True Positives + True Negatives) / Total observations)

= 1 – ((56 + 50) / (56 + 9 + 14 + 50))

= 1 – (106 / 129)

= 1 – 0.8217054264

= **0.1782945736**

8) This question deals with sonar data

a) Use knn() for the k-nearest neighbor classifier for k=5 and k=6 to the last column of the sonar training data. Compute the misclassification error on the training data and also on the test data.

```
> setwd("E:\\MSIT-II\\Data Science Specialization\\Intro to ML Repo\\Data Mining\\DM As
signment4")
> sonar_train <- read.csv("sonar_train.csv")
> sonar_test <- read.csv("sonar_test.csv")
> x_train <- sonar_train[, 1:60]
> y_train <- sonar_train[, 61]
> x_test <- sonar_test[, 1:60]
> y_test <- sonar_test[, 61]
> names(sonar_train)
 [1] "X0.0258" "X0.0433" "X0.0547" "X0.0681" "X0.0784" "X0.125"  "X0.1296" "X0.1729"
 [9] "X0.2794" "X0.2954" "X0.2506" "X0.2601" "X0.2249" "X0.2115" "X0.127"  "X0.1193"
[17] "X0.1794" "X0.2185" "X0.1646" "X0.074"  "X0.0625" "X0.2381" "X0.4824" "X0.6372"
[25] "X0.7531" "X0.8959" "X0.9941" "X0.9957" "X0.9328" "X0.9344" "X0.8854" "X0.769"
[33] "X0.6865" "X0.639"  "X0.6378" "X0.6629" "X0.5983" "X0.4565" "X0.3129" "X0.4158"
[41] "X0.4325" "X0.4031" "X0.4201" "X0.4557" "X0.3955" "X0.2966" "X0.2095" "X0.1558"
[49] "X0.0884" "X0.0265" "X0.0121" "X0.0091" "X0.0062" "X0.0019" "X0.0045" "X0.0079"
[57] "X0.0031" "X0.0063" "X0.0048" "X0.005"  "X.1"
```

```
> knn_fit <- knn(x_train, x_train, k = 5, cl = sonar_train$X.1)
> knn_fit
  [1] -1 -1 1  -1 1  -1 -1 -1 -1 -1 1  -1 -1 -1 1  -1 1  1  1  1  -1 -1 -1 1  1  -1
 [27] 1  1  1  -1 1  -1 -1 -1 -1 -1 -1 -1 -1 -1 1  -1 1  -1 -1 1  -1 1  -1 -1 -1 1
 [53] -1 1  -1 -1 1  1  1  1  1  -1 -1 -1 -1 1  1  1  1  -1 -1 -1 -1 1  1  -1 -1 -1
 [79] 1  1  -1 1  -1 -1 1  -1 -1 -1 -1 1  -1 -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 -1 -1
[105] 1  -1 -1 -1 1  1  -1 1  -1 -1 1  1  -1 -1 -1 -1 -1 -1 1  -1 1  1  -1 1  -1
Levels: -1 1
> library(gmodels)
> CrossTable(x = sonar_train$X.1, y = knn_fit, prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  129


                | knn_fit
sonar_train$X.1 |        -1 |         1 | Row Total |
----------------|-----------|-----------|-----------|
             -1 |        59 |         6 |        65 |
                |     0.908 |     0.092 |     0.504 |
                |     0.738 |     0.122 |           |
                |     0.457 |     0.047 |           |
----------------|-----------|-----------|-----------|
              1 |        21 |        43 |        64 |
                |     0.328 |     0.672 |     0.496 |
                |     0.263 |     0.878 |           |
                |     0.163 |     0.333 |           |
----------------|-----------|-----------|-----------|
   Column Total |        80 |        49 |       129 |
                |     0.620 |     0.380 |           |
----------------|-----------|-----------|-----------|


> summary(knn_fit)
-1  1
80 49
> mis_error_train = 1-sum(knn_fit == y_train)/length(y_train)
> mis_error_train
[1] 0.2093023
```

```
> knn_fit <- knn(x_train, x_train, k = 6, cl = sonar_train$X.1)
> knn_fit
  [1] -1 -1 1  -1 1  -1 -1 -1 -1 -1 1  -1 -1 -1 -1 -1 1  1  1  1  -1 -1 -1 1  1  -1
 [27] 1  1  1  -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 1  -1 1  -1 -1 1  -1 1  -1 -1 -1 1
 [53] 1  1  -1 -1 1  1  1  1  1  -1 -1 -1 -1 1  1  1  1  1  -1 -1 -1 1  1  -1 -1 -1
 [79] 1  1  -1 1  -1 -1 1  -1 -1 -1 1  1  -1 -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 -1 -1
[105] -1 -1 -1 -1 1  1  -1 1  -1 -1 1  1  -1 -1 -1 -1 -1 1  1  -1 1  1  -1 1  -1
Levels: -1 1
> library(gmodels)
> CrossTable(x = sonar_train$X.1, y = knn_fit, prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  129


                 | knn_fit
sonar_train$X.1 |        -1 |         1 | Row Total |
----------------|-----------|-----------|-----------|
             -1 |        57 |         8 |        65 |
                |     0.877 |     0.123 |     0.504 |
                |     0.750 |     0.151 |           |
                |     0.442 |     0.062 |           |
----------------|-----------|-----------|-----------|
              1 |        19 |        45 |        64 |
                |     0.297 |     0.703 |     0.496 |
                |     0.250 |     0.849 |           |
                |     0.147 |     0.349 |           |
----------------|-----------|-----------|-----------|
   Column Total |        76 |        53 |       129 |
                |     0.589 |     0.411 |           |
----------------|-----------|-----------|-----------|


> mis_error_train_k6 = 1-sum(knn_fit == y_train)/length(y_train)
> mis_error_train_k6
[1] 0.2093023
```

```
> knn_fit_train_k5 <- knn(x_train, x_train, k = 5, cl = sonar_train$X.1)
> knn_fit_train_k5
  [1] -1 -1 1  -1 1  -1 -1 -1 -1 -1 1  -1 -1 -1 1  -1 1  1  1  1  -1 -1 -1 1  1  -1
 [27] 1  1  1  -1 1  -1 -1 -1 -1 -1 -1 -1 -1 -1 1  -1 1  -1 -1 1  -1 1  -1 -1 -1 1
 [53] -1 1  -1 -1 1  1  1  1  1  -1 -1 -1 -1 1  1  1  1  -1 -1 -1 -1 1  1  -1 -1 -1
 [79] 1  1  -1 1  -1 -1 1  -1 -1 -1 -1 1  -1 -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 -1 -1
[105] 1  -1 -1 -1 1  1  -1 1  -1 -1 1  1  -1 -1 -1 -1 -1 -1 1  -1 1  1  -1 1  -1
Levels: -1 1
> library(gmodels)
> CrossTable(x = sonar_train$X.1, y = knn_fit_train_k5, prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  129


                | knn_fit_train_k5
sonar_train$X.1 |        -1 |         1 | Row Total |
----------------|-----------|-----------|-----------|
             -1 |        59 |         6 |        65 |
                |     0.908 |     0.092 |     0.504 |
                |     0.738 |     0.122 |           |
                |     0.457 |     0.047 |           |
----------------|-----------|-----------|-----------|
              1 |        21 |        43 |        64 |
                |     0.328 |     0.672 |     0.496 |
                |     0.263 |     0.878 |           |
                |     0.163 |     0.333 |           |
----------------|-----------|-----------|-----------|
   Column Total |        80 |        49 |       129 |
                |     0.620 |     0.380 |           |
----------------|-----------|-----------|-----------|


> misclassify_error_train_k5 = 1 - (sum(knn_fit_train_k5 == y_train) / length(y_train))
> misclassify_error_train_k5
[1] 0.2093023
```

```
> knn_fit_train_k6 <- knn(x_train, x_train, k = 6, cl = sonar_train$X.1)
> knn_fit_train_k6
  [1] -1 -1 1  -1 1  -1 -1 -1 -1 -1 1  -1 -1 -1 1  -1 1  1  1  1  -1 -1 -1 -1 1  -1
 [27] 1  1  1  -1 1  -1 -1 -1 -1 1  -1 -1 -1 -1 1  -1 1  -1 -1 1  -1 1  1  1  -1 1
 [53] -1 1  -1 -1 1  1  1  1  1  -1 -1 -1 -1 1  1  1  1  1  -1 -1 -1 1  1  -1 -1 -1
 [79] 1  1  -1 1  -1 -1 1  -1 -1 -1 -1 1  -1 -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 -1 -1
[105] 1  -1 -1 -1 1  -1 -1 -1 -1 -1 1  1  -1 -1 -1 -1 -1 1  1  -1 1  1  1  1  -1
Levels: -1 1
> library(gmodels)
> CrossTable(x = sonar_train$X.1, y = knn_fit_train_k6, prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  129


                | knn_fit_train_k6 
sonar_train$X.1 |        -1 |         1 | Row Total | 
----------------|-----------|-----------|-----------|
             -1 |        58 |         7 |        65 | 
                |     0.892 |     0.108 |     0.504 | 
                |     0.753 |     0.135 |           | 
                |     0.450 |     0.054 |           | 
----------------|-----------|-----------|-----------|
              1 |        19 |        45 |        64 | 
                |     0.297 |     0.703 |     0.496 | 
                |     0.247 |     0.865 |           | 
                |     0.147 |     0.349 |           | 
----------------|-----------|-----------|-----------|
   Column Total |        77 |        52 |       129 | 
                |     0.597 |     0.403 |           | 
----------------|-----------|-----------|-----------|


> misclassify_error_train_k6 = 1 - (sum(knn_fit_train_k6 == y_train) / length(y_train))
> misclassify_error_train_k6
[1] 0.2015504
```

```
> knn_fit_test_k5 <- knn(x_test, x_test, k = 5, cl = sonar_test$X.1)
> knn_fit_test_k5
 [1] -1 -1 1  -1 1  -1 1  -1 -1 -1 1  -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 -1 -1 1  -1
[27] -1 1  1  1  -1 1  -1 -1 1  -1 -1 1  -1 1  -1 -1 -1 1  -1 1  1  1  -1 -1 -1 1
[53] 1  -1 -1 -1 -1 -1 -1 1  -1 -1 -1 -1 -1 1  -1 -1 -1 -1 -1 -1 -1 1  1  -1 -1
Levels: -1 1
> library(gmodels)
> CrossTable(x = sonar_test$X.1, y = knn_fit_test_k5, prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  77


               | knn_fit_test_k5
sonar_test$X.1 |        -1 |         1 | Row Total |
---------------|-----------|-----------|-----------|
            -1 |        40 |         4 |        44 |
               |     0.909 |     0.091 |     0.571 |
               |     0.769 |     0.160 |           |
               |     0.519 |     0.052 |           |
---------------|-----------|-----------|-----------|
             1 |        12 |        21 |        33 |
               |     0.364 |     0.636 |     0.429 |
               |     0.231 |     0.840 |           |
               |     0.156 |     0.273 |           |
---------------|-----------|-----------|-----------|
  Column Total |        52 |        25 |        77 |
               |     0.675 |     0.325 |           |
---------------|-----------|-----------|-----------|


> misclassify_error_test_k5 = 1 - (sum(knn_fit_test_k5 == y_test) / length(y_test))
> misclassify_error_test_k5
[1] 0.2077922
```

```
> knn_fit_test_k6 <- knn(x_test, x_test, k = 6, cl = sonar_test$X.1)
> knn_fit_test_k6
 [1] -1 -1 1  -1 1  -1 1  -1 -1 -1 1  -1 1  -1 -1 -1 -1 1  -1 -1 -1 -1 -1 -1 1  -1
[27] 1  -1 1  1  -1 1  -1 1  -1 1  -1 1  1  1  -1 -1 -1 1  -1 1  1  1  -1 -1 -1 -1
[53] 1  -1 -1 -1 -1 -1 -1 1  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1  -1 -1 -1
Levels: -1 1
> library(gmodels)
> CrossTable(x = sonar_test$X.1, y = knn_fit_test_k6, prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  77


               | knn_fit_test_k6
 sonar_test$X.1 |        -1 |         1 | Row Total |
---------------|-----------|-----------|-----------|
            -1 |        38 |         6 |        44 |
               |     0.864 |     0.136 |     0.571 |
               |     0.704 |     0.261 |           |
               |     0.494 |     0.078 |           |
---------------|-----------|-----------|-----------|
             1 |        16 |        17 |        33 |
               |     0.485 |     0.515 |     0.429 |
               |     0.296 |     0.739 |           |
               |     0.208 |     0.221 |           |
---------------|-----------|-----------|-----------|
  Column Total |        54 |        23 |        77 |
               |     0.701 |     0.299 |           |
---------------|-----------|-----------|-----------|


> misclassify_error_test_k6 = 1 - (sum(knn_fit_test_k6 == y_test) / length(y_test))
> misclassify_error_test_k6
[1] 0.2857143
```

b) Repeat part a using the exact same R code a few times. Explain why both the training errors and the test errors often change for k=6 but not for k=5. Hint: Read the help on the knn function if you do not know.

If there exists any tie in the values of Kth nearest vector then all the candidates are included in vote.

For the value, k = 6, there are ties for 6th nearest vector.

Hence there are changes in the misclassification error rate as all the candidates will be included in the vote.