Data Mining Assignment 4

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

2) Repeat in Class Exercise #38 using the misclassification error rate instead of information gain to determine the best split. Which of these splits considered is the

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

best according to misclassification error rate?

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?

i. As we can see from the above table, if we split on A, the misclassification error would be: 3/10=0.3. Because in rows 4, 9 and 10 we can see that three records of A are misclassified and 10 is the total number of records

ii. If we split on B, there are misclassifications in row 1 and 9 with respect to B, so the rate would be 0.2.

iii. Since the misclassification rate is low when we split the data set on B, we need to induct our decision tree based on B split.

3) Repeat In Class Exercise #39 using the misclassification error rate instead of information gain to determine the best split. Which of these splits considered is the best according to misclassification error rate?

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	−

i. Splitting on a1, the misclassification error rate =2/9=0.22

ii. Splitting on a2, the misclassification error rate=5/9=0.55

iii. Splitting on a3, [So, splitting on a3 will not be straight because it is not a nominal value or categorical value. Here, the a3 has discrete values and I decided to split on condition a3 < 5.0 as + a3 >= 5.0 as -], the misclassification error rate would be=3/9=0.33

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.

i. Age = Middle, Number = 5 and Start = 10, the class label is present, as we traverse from 1 -> 2 -> 5 -> 11

ii. Age = young, Number = 2, Start = 17, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8

iii. Age = old, Number = 10, Start = 6, the class label is present, as we traverse from 1 -> 3 -> 7 -> 15

iv. Age = young, Number = 2, Start = 17, the class label is absent, as we

traverse from 1 -> 2 -> 4 -> 8

v. Age = old, Number = 4, Start = 15, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8

vi. Age = middle, Number = 5, Start = 15, the class label is absent, as we traverse from 1 -> 2 -> 5 -> 10

vii. Age = young, Number = 3, Start = 13, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 9

viii. Age = old, Number = 5, Start = 8, the class label is present, as we traverse from 1 -> 3 -> 7 -> 15

ix. Age = young, Number = 7, Start = 9, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 9

x. Age = middle, Number = 3, Start = 13, the class label is absent, as we traverse from 1 -> 2 -> 5 -> 10

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.

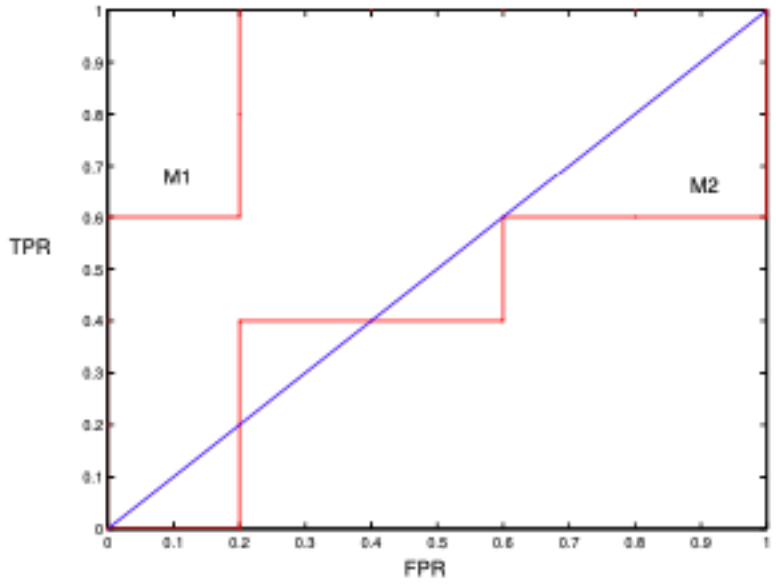
```
> setwd("D:\\Courses\\2nd Year\\4.Data Science\\-Sp-Data_Science_20190125\\Intro to Data Mining\\DM Assignment4\\")
>
> #5
>
> test<-read.csv("sonar_test.csv", header=FALSE)
> train<-read.csv("sonar_train.csv", header=FALSE)
>
> y<-as.factor(train[,61])
> x<-train[,1:60]
> y_test<-as.factor(test[,61])
> x_test<-test[,1:60]
> library(rpart)
> fit<- rpart(y~.,x,control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5))
> error = 1-sum(y_test==predict(fit,x_test, type="class"))/length(y_test)
> cat("Misclassification Error:",error)
Misclassification Error: 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.

Instance	True Class	$P(+\|A,\ldots,Z,M_1)$	$P(+\|A,\ldots,Z,M_2)$
1	+	0.73	0.61
2	+	0.69	0.03
3	−	0.44	0.68
4	−	0.55	0.31
5	+	0.67	0.45
6	+	0.47	0.09
7	−	0.08	0.38
8	−	0.15	0.05
9	+	0.45	0.01
10	−	0.35	0.04

(a) Plot the ROC curve for both M1 and M2. (You should plot them on the same graph.) Which model do you think is better? Explain your reasons.

Answer: The ROC curve for M1 and M2 are shown below.
M1 is better, since its area under the ROC curve is larger than the area under ROC curve for M2.



ROC curve

7) Compute the misclassification error on the training data for the Random Forest classifier from In Class Exercise #47. Show your R code for doing this.

```
> fit<-randomForest(x,y)
> error_rate = 1-sum(y==predict(fit,x))/length(y)
> cat("Misclassification Error rate:",error_rate)
Misclassification Error rate: 0
> |
```

8) This question deals with In Class Exercise #42.

a) Repeat In Class Exercise #42 for the k-nearest neighbor classifier for k=5 and k=6.

```
> #8
>
> #8a
> library(class)
> fit_train<-knn(x,x,y,k=5)
> train_error = 1-sum(y==fit_train)/length(y)
> cat("Train Error rate:",train_error)
Train Error rate: 0.2076923>
>
> fit_test<-knn(x,x_test,y,k=5)
> test_error= 1-sum(y_test==fit_test)/length(y_test)
> cat("\n Test Error rate:",test_error)

 Test Error rate: 0.2307692
> |
```

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.

In the help for the knn function it states “ties broken at random”. For each row of the test set, the k nearest (in Euclidean distance) training set vectors are found, and the classification is decided by majority vote, with ties broken at random. If there are ties for the kth nearest vector, all candidates are included in the vote. For odd k, there will never be ties, while for even k, there are frequently ties.