Assignment 4 Solutions

1) 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?

Ans: If splitting is done on A, the misclassification error rate will be 3/10 as in the rows 4,9,10 we can see the three records are misclassified and the total number of records is 10 with respect to A. If splitting is done on B, the misclassification error rate will be 2/10 as the rows 1 and 9 are misclassified with respect to B.

2) 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?

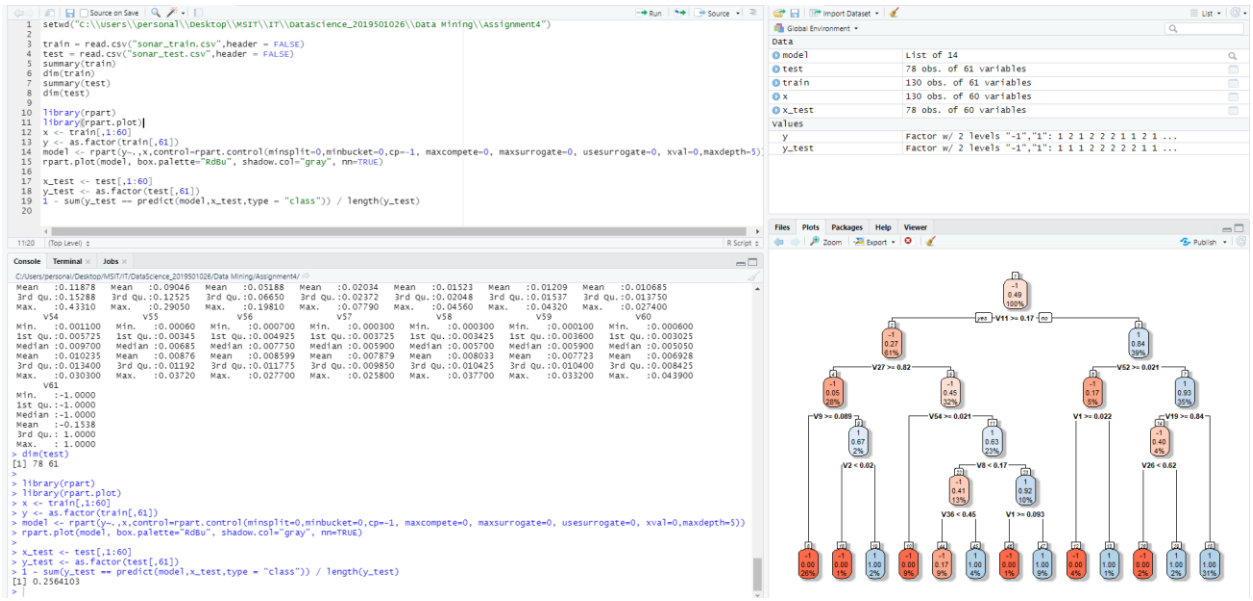
Ans: Misclassification error rate while splitting on a1 is 2/9 and while splitting on a2 is 5/9. But while splitting on a3, as a3 is not a categorical value, splitting will not be straight. Here a3 has discrete values and we can split based on one condition i.e., a3 < 5.0 as + and a3 >= 5.0 as -. Then the misclassification error rate will be 3/9.

3) 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.

Ans:

1. **Age = Middle, Number = 5 and Start = 10, the class label is present, as we traverse from 1 -> 2 -> 5 -> 11**
2. **Age = young, Number = 2, Start = 17, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8**
3. **Age = old, Number = 10, Start = 6, the class label is present, as we traverse from 1 -> 3 -> 7 -> 15**
4. **Age = young, Number = 2, Start = 17, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8**
5. **Age = old, Number = 4, Start = 15, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8**
6. **Age = middle, Number = 5, Start = 15, the class label is absent, as we traverse from 1 -> 2 -> 5 -> 10**
7. **Age = young, Number = 3, Start = 13, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 9**
8. **Age = old, Number = 5, Start = 8, the class label is present, as we traverse from 1 -> 3 -> 7 -> 15**
9. **Age = young, Number = 7, Start = 9, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 9**
10. **Age = middle, Number = 3, Start = 13, the class label is absent, as we traverse from 1 -> 2 -> 5 -> 10**

4) 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("C:\\Users\\personal\\Desktop\\MSIT\\IT\\DataScience_2019501026\\Data Mining\\Assignment4")

train = read.csv("sonar_train.csv",header = FALSE)
test = read.csv("sonar_test.csv",header = FALSE)
summary(train)
dim(train)
summary(test)
dim(test)

library(rpart)
library(rpart.plot)
x <- train[,1:60]
y <- as.factor(train[,61])
model <- rpart(y~.,x,control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5))
rpart.plot(model, box.palette="RdBu", shadow.col="gray", nn=TRUE)

x_test <- test[,1:60]
y_test <- as.factor(test[,61])
1 - sum(y_test == predict(model,x_test,type = "class")) / length(y_test)
```

```
      V1              V2              V3              V4              V5              V6              V7              V8              V9              V10             V11             V12             V13             V14             V15             V16             V17             V18             V19             V20             V21             V22             V23             V24             V25             V26             V27             V28             V29             V30             V31             V32             V33             V34             V35             V36             V37             V38             V39             V40             V41             V42             V43             V44             V45             V46             V47             V48             V49             V50             V51             V52             V53             V54             V55             V56             V57             V58             V59             V60
 Mean   :0.11878   Mean   :0.09046   Mean   :0.05188   Mean   :0.02034   Mean   :0.01523   Mean   :0.01209   Mean   :0.010685
 3rd Qu.:0.15288   3rd Qu.:0.12525   3rd Qu.:0.06650   3rd Qu.:0.02372   3rd Qu.:0.02048   3rd Qu.:0.01537   3rd Qu.:0.013750
 Max.   :0.43310   Max.   :0.29050   Max.   :0.19810   Max.   :0.07790   Max.   :0.04560   Max.   :0.04320   Max.   :0.027400
      V54             V55             V56             V57             V58             V59             V60
 Min.   :0.001100   Min.   :0.00060   Min.   :0.000700   Min.   :0.000300   Min.   :0.000300   Min.   :0.000100   Min.   :0.000600
 1st Qu.:0.005725   1st Qu.:0.00345   1st Qu.:0.004925   1st Qu.:0.003725   1st Qu.:0.003425   1st Qu.:0.003600   1st Qu.:0.003025
 Median :0.009700   Median :0.00685   Median :0.007750   Median :0.005900   Median :0.005700   Median :0.005900   Median :0.005050
 Mean   :0.010235   Mean   :0.00876   Mean   :0.008599   Mean   :0.007879   Mean   :0.008033   Mean   :0.007723   Mean   :0.006928
 3rd Qu.:0.013400   3rd Qu.:0.01192   3rd Qu.:0.011775   3rd Qu.:0.009850   3rd Qu.:0.010425   3rd Qu.:0.010400   3rd Qu.:0.008425
 Max.   :0.030300   Max.   :0.03720   Max.   :0.027700   Max.   :0.025800   Max.   :0.037700   Max.   :0.033200   Max.   :0.043900
      V61
 Min.   :-1.0000
 1st Qu.:-1.0000
 Median :-1.0000
 Mean   :-0.1538
 3rd Qu.: 1.0000
 Max.   : 1.0000
> dim(test)
[1] 78 61
>
> library(rpart)
> library(rpart.plot)
> x <- train[,1:60]
> y <- as.factor(train[,61])
> model <- rpart(y~.,x,control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5))
> rpart.plot(model, box.palette="RdBu", shadow.col="gray", nn=TRUE)
>
> x_test <- test[,1:60]
> y_test <- as.factor(test[,61])
> 1 - sum(y_test == predict(model,x_test,type = "class")) / length(y_test)
[1] 0.2564103
>
```

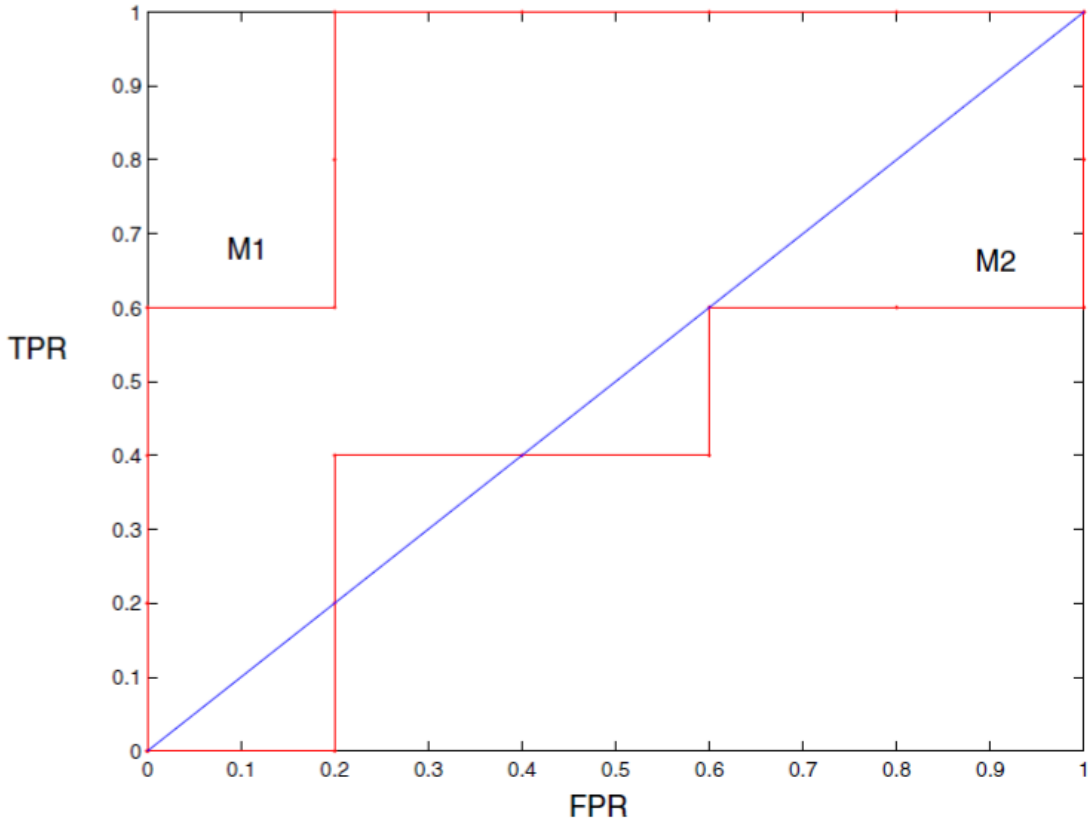
5) Do Chapter 5 textbook problem #17 (parts a and c only) on pages 322-323. Note that there is a typo in part c - it should read "Repeat the analysis for part (b)". We will do part b in class.

You are asked to evaluate the performance of two classification models, M1 and M2. The test set you have chosen contains 26 binary attributes, labeled as A through Z.
Table 5.14 shows the posterior probabilities obtained by applying the models to the test set. (Only the posterior probabilities for the positive class are shown). As this is a two-class problem, P(-) = 1- P(+) and P(-|A, . . . , Z) = 1 - P(+|A, . . . , Z). Assume that we are mostly interested in detecting instances from the positive class.

Table 5.14. Posterior probabilities for Exercise 17.

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.



M1 is better, since its area under the ROC curve is larger than the area under ROC curve for M2.

(c) Repeat the analysis for part (c) using the same cutoff threshold on model M2. Compare the F-measure results for both models. Which model is better? Are the results consistent with what you expect from the ROC curve?

For model M2 when t = 0.5: Precision = 1/2 = 50%. Recall = 1/5 = 20%.

F-measure = (2 × .5 × .2) / (.5 + .2) = 0.2857.

6) 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.

```
setwd("C:\\Users\\personal\\Desktop\\MSIT\\IT\\DataScience_2019501026\\Data Mining\\Assignment4")
install.packages("randomForest")
library("randomForest")

train <- read.csv("sonar_test.csv", header = FALSE)
test <- read.csv("sonar_test.csv", header = FALSE)

x_train = train[,1:60]
y_train = as.factor(train[,61])

x_test = test[,1:60]
y_test = as.factor(test[,61])

model<-randomForest(x_train, y_train)
1 - sum(y_train == predict(model, x_train)) / length(y_train)
```

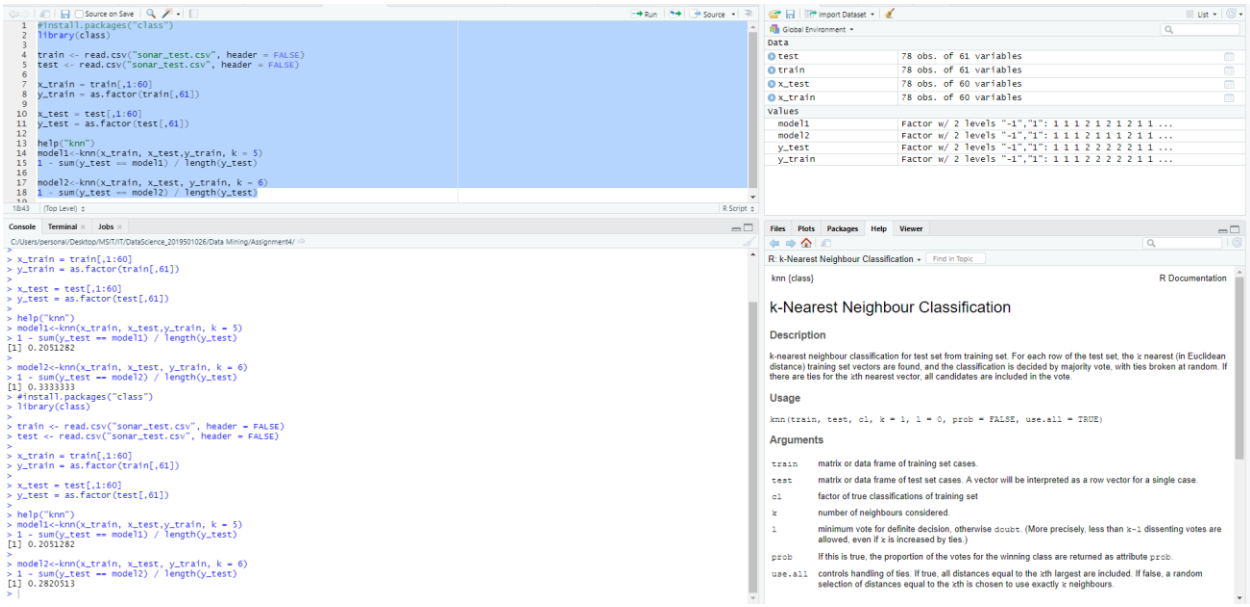
```
C:/Users/personal/Desktop/MSIT/IT/DataScience_2019501026/Data Mining/Assignment4/
> setwd("C:\\Users\\personal\\Desktop\\MSIT\\IT\\DataScience_2019501026\\Data Mining\\Assignment4")
> install.packages("randomForest")
Error in install.packages : Updating loaded packages
> library("randomForest")
>
> train <- read.csv("sonar_test.csv", header = FALSE)
> test <- read.csv("sonar_test.csv", header = FALSE)
>
> x_train = train[,1:60]
> y_train = as.factor(train[,61])
>
> x_test = test[,1:60]
> y_test = as.factor(test[,61])
>
> model<-randomForest(x_train, y_train)
> 1 - sum(y_train == predict(model, x_train)) / length(y_train)
[1] 0
```

The misclassification error was 0 on the training data using Random Forest

7) 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.

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.



We can choose any value for k either even or odd but it is better to choose an odd value for k when there are two classes because if k value is even there might be a risk of a tie in the decision of which class to choose. So, the misclassification error rate often changes for k=6.