

## 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 best according to misclassification error rate?

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
T	F	-

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

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	-

- Splitting on  $a_1$ , the misclassification error rate =  $2 / 9 = 0.22$
- Splitting on  $a_2$ , the misclassification error rate =  $5 / 9 = 0.55$
- Splitting on  $a_3$ , [ So, splitting on  $a_3$  will not be straight because it is not a nominal value or categorical value. Here, the  $a_3$  has discrete values and I decided to split on condition  $a_3 < 5.0$  as +  $a_3 \geq 5.0$  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](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](http://www-stat.wharton.upenn.edu/~dmease/test_data.csv) linked here.

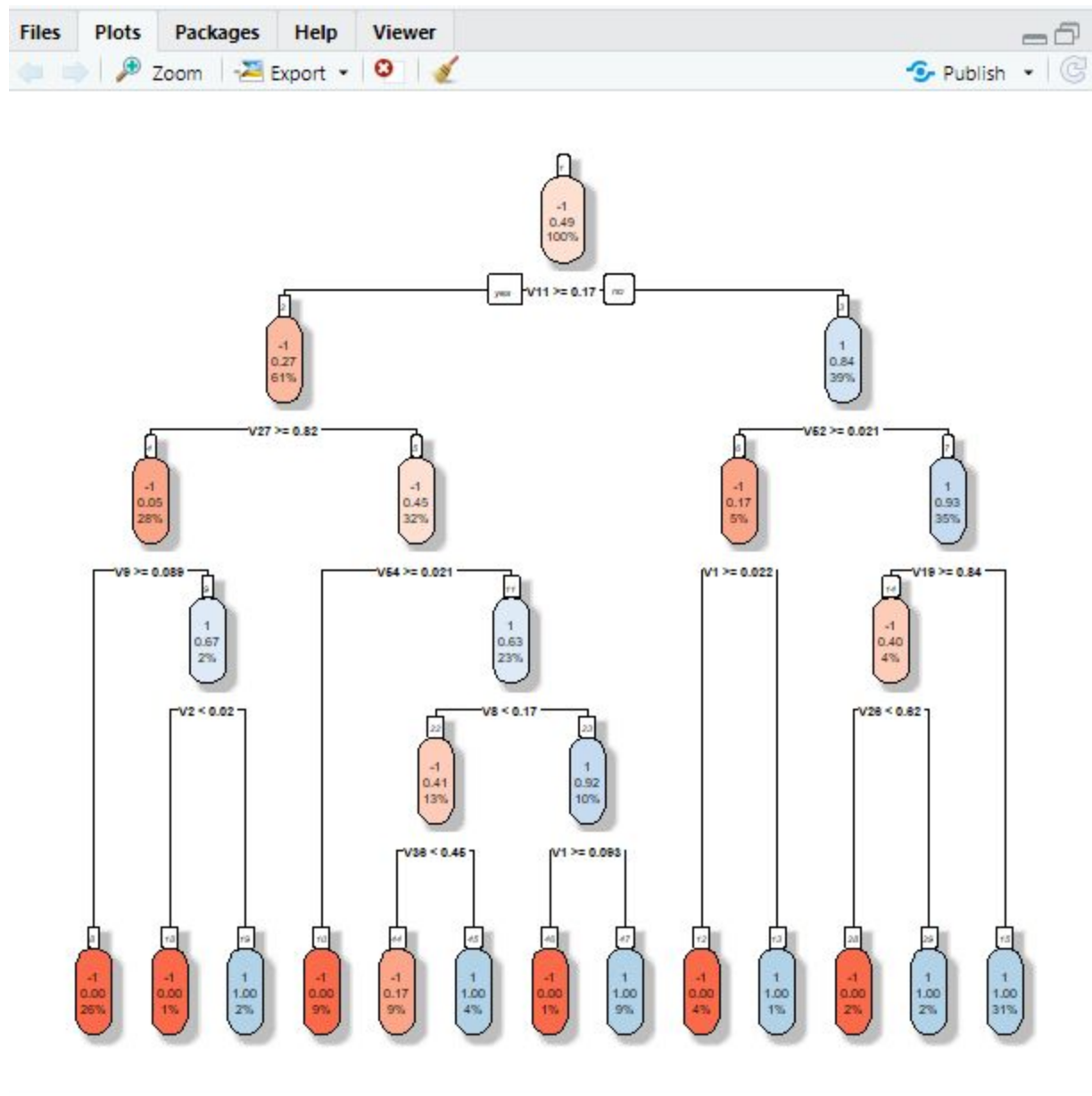
Do this manually - do not use R or any software.

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

- Age = young, Number = 2, Start = 17, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8
- Age = old, Number = 10, Start = 6, the class label is present, as we traverse from 1 -> 3 -> 7 -> 15
- Age = young, Number = 2, Start = 17, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8
- Age = old, Number = 4, Start = 15, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 8

- Age = middle, Number = 5, Start = 15, the class label is absent, as we traverse from 1 -> 2 -> 5 -> 10
- Age = young, Number = 3, Start = 13, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 9
- Age = old, Number = 5, Start = 8, the class label is present, as we traverse from 1 -> 3 -> 7 -> 15
- Age = young, Number = 7, Start = 9, the class label is absent, as we traverse from 1 -> 2 -> 4 -> 9
- 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](http://www-stat.wharton.upenn.edu/~dmease/sonar_train.csv)) and a test set ([http://www-stat.wharton.upenn.edu/~dmease/sonar\\_test.csv](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.



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.

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.

```

>
> 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
>

```

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.

```

> x_train = train[,1:60]
> y_train = as.factor(train[,61])
>
> x_test = test[,1:60]
> y_test = as.factor(test[,61])
>
> help("knn")
> model1<-knn(x_train, x_test,y_train, k = 5)
> 1 - sum(y_test == model1) / length(y_test)
[1] 0.2051282
>
> model2<-knn(x_train, x_test, y_train, k = 6)
> 1 - sum(y_test == model2) / length(y_test)
[1] 0.2948718
>

```

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.

---

```
> x_train = train[,1:60]
> y_train = as.factor(train[,61])
>
> x_test = test[,1:60]
> y_test = as.factor(test[,61])
>
> help("knn")
> model1<-knn(x_train, x_test,y_train, k = 5)
> 1 - sum(y_test == model1) / length(y_test)
[1] 0.2051282
>
> model2<-knn(x_train, x_test, y_train, k = 6)
> 1 - sum(y_test == model2) / length(y_test)
[1] 0.3205128
> |
```

---