

Homework 3 Solutions

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

2) Splitting on A gives a misclassification error rate of $3/10=30\%$ while splitting on B gives a misclassification error rate of $2/10=20\%$ so B is preferred according to the misclassification error rate.

3) Splitting on either 2.0 or 4.5 is the best in terms of misclassification error. Either one will give you 33% misclassification error which is the best you can do by splitting on a_3 .

Split Point	Misclassification Error
2.0	$3/9=33\%$
3.5	$4/9=44\%$
4.5	$3/9=33\%$
5.5	$4/9=44\%$
6.5	$4/9=44\%$
7.5	$4/9=44\%$

4) Here are the correct predictions:

Age	Number	Start	Prediction
middle	5	10	present
young	2	17	absent
old	10	6	present
young	2	17	absent
old	4	15	absent
middle	5	15	absent
young	3	13	absent
old	5	8	present
young	7	9	absent
middle	3	13	absent

5)

```
install.packages("rpart")
library(rpart)

train<-read.csv("sonar_train.csv",header=FALSE)
y<-as.factor(train[,61])
x<-train[,1:60]

test<-read.csv("sonar_test.csv",header=FALSE)
y_test<-as.factor(test[,61])
x_test<-test[,1:60]

fit<-rpart(y~.,x,
           control=rpart.control(minsplit=0,minbucket=0,cp=-1,
                                 maxcompete=0, maxsurrogate=0, usesurrogate=0,
                                 xval=0,maxdepth=5))

1-sum(y_test==predict(fit,x_test,type="class"))/length(y_test)
```

The test error is 25.6% using the code above.

6)

a)

The ROC curve for $M1$ and $M2$ are shown in the Figure 5.5.

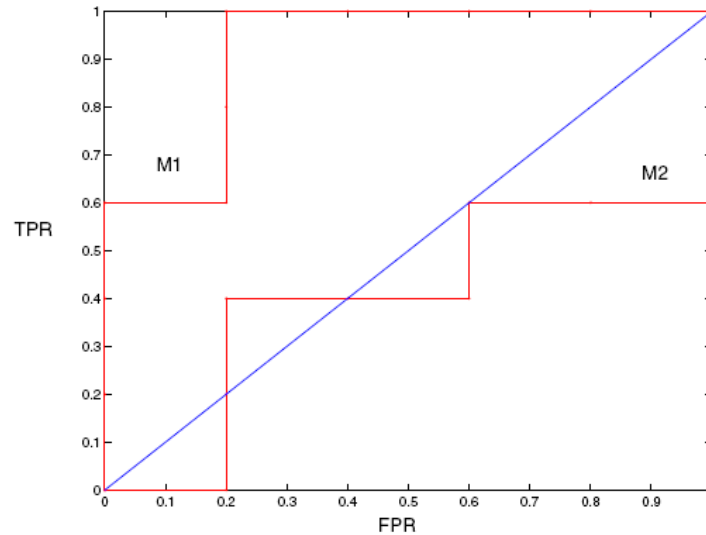


Figure 5.5. ROC curve.

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

c)

When $t = 0.5$, the confusion matrix for $M2$ is shown below.

		+	-
Actual	+	1	4
	-	1	4

Precision = $1/2 = 50\%$.

Recall = $1/5 = 20\%$.

F-measure = $(2 \times .5 \times .2) / (.5 + .2) = 0.2857$.

Based on F-measure, $M1$ is still better than $M2$. This result is consistent with the ROC plot.

7) The training error is zero found by the following R code:

```
install.packages("randomForest")
library(randomForest)
train<-read.csv("sonar_train.csv",header=FALSE)
y<-as.factor(train[,61])
x<-train[,1:60]
fit<-randomForest(x,y)
1-sum(y==predict(fit,x))/length(y)
```

8) a)

```
install.packages("class")
library(class)
train<-read.csv("sonar_train.csv",header=FALSE)
y<-as.factor(train[,61])
x<-train[,1:60]
test<-read.csv("sonar_test.csv",header=FALSE)
y_test<-as.factor(test[,61])
x_test<-test[,1:60]
```

```
fit<-knn(x,x,y,k=5)
1-sum(y==fit)/length(y)
```

```
fit_test<-knn(x,x_test,y,k=5)
1-sum(y_test==fit_test)/length(y_test)
```

For k=5 the training error is 20.8% and the test error is 23.1% using the above code.

```
fit<-knn(x,x,y,k=6)
1-sum(y==fit)/length(y)
```

```
fit_test<-knn(x,x_test,y,k=6)
1-sum(y_test==fit_test)/length(y_test)
```

For k=6 you will get many different answers. See part b for an explanation as to why.

b) In the help for the knn function it states “ties broken at random”. For odd k, there will never be ties, while for even k, there are frequently ties. How the ties are broken will randomly alter the training error and test error.