

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

$$\text{Error}(A=T) = 1 - \max(4/7, 3/7) = 3/7 = 0.43$$

$$\text{Error}(A=F) = 1 - \max(0/3, 3/3) = 0$$

$$\text{weighted average error} = 7/10 * 0.43 + 3/10 * 0 = 0.30$$

splitting on B

$$\text{Error}(B=T) = 1 - \max(3/4, 1/4) = 1/4 = 0.25$$

$$\text{Error}(B=F) = 1 - \max(1/6, 5/6) = 1/6 = 0.16$$

$$\text{weighted average error} = 4/10 * 0.25 + 6/10 * 0.16 = 0.20$$

According to misclassification error rate, splitting on B is the best split, because the weighted average error for B is less than A.

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

For a_3 , 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?

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	-

a_3	Class Label	Split point	weighted Error
1.0	+	2.0	0.33
3.0	-	3.5	0.42
4.0	+	4.5	0.33
5.0	-	5.5	0.44
5.0	+		0.44
6.0	+	6.5	0.44
7.0	-	7.5	0.44
7.0	+		0.44

split point 2.0 or 4.5 is the best split according to weighted error.

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	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

Predictions on the above test data are:

- i. Age = middle, Number = 5, Start = 10
Path : 1 → 2 → 5 → 11 → Present
- ii. Age = young, Number = 2, Start = 17
Path : 1 → 2 → 4 → 8 → Absent
- iii. Age = old, Number = 10, Start = 6
Path : 1 → 3 → 7 → 15 → Present
- iv. Age = young, Number = 2, Start = 17
Path : 1 → 2 → 4 → 8 → Absent
- v. Age = old, Number = 4, Start = 15
Path : 1 → 2 → 4 → 8 → Absent
- vi. Age = middle, Number = 5, Start = 15
Path : 1 → 2 → 5 → 10 → Absent
- vii. Age = young, Number = 3, Start = 13
Path : 1 → 2 → 4 → 9 → Absent
- viii. Age = old, Number = 5, Start = 8
Path : 1 → 3 → 7 → 15 → Present
- ix. Age = young, Number = 7, Start = 9
Path : 1 → 2 → 4 → 9 → Absent
- x. Age = middle, Number = 3, Start = 13
Path : 1 → 2 → 5 → 10 → 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.

The screenshot shows the RStudio interface with the following components:

- Console:** Displays R code and its output. The code reads 'sonar_test.csv' and 'sonar_train.csv', creates factors for the response variable (column 61), and fits an rpart model with specific control parameters. The final output is a misclassification error of 0.2564103.
- Environment:** Lists the objects in the global environment: 'fit' (a list of 14), 'testdata' (78 obs. of 61 variables), 'traindata' (130 obs. of 61 variables), 'x' (130 obs. of 60 variables), and 'x_test' (78 obs. of 60 variables).
- Files:** Shows the file explorer with 'sonar_test.csv' and 'sonar_train.csv' loaded.

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.

You are asked to evaluate the performance of two classification models, M_1 and M_2 . The test set you have chosen contains 26 binary attributes, labeled as A through Z.

Table 4.5 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.

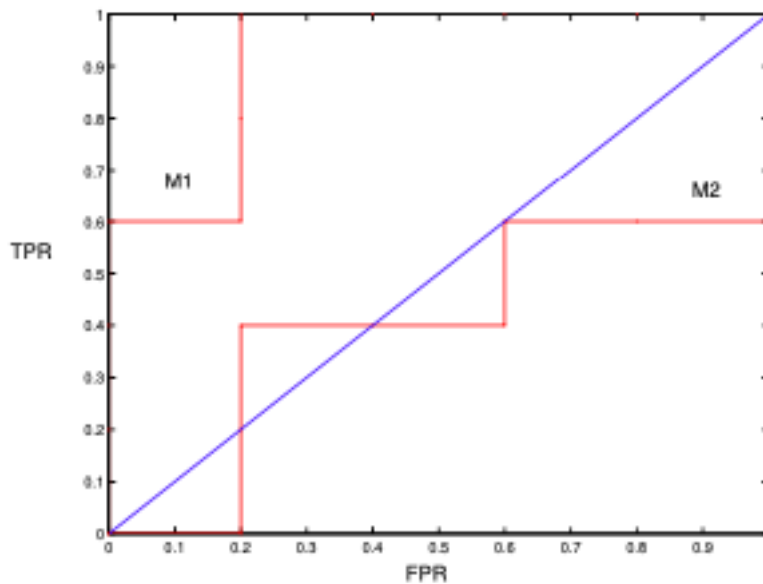
Table 4.5. Posterior probabilities for Exercise 16.

Instance	True Class	$P(+ A, \dots, Z, M_1)$	$P(+ A, \dots, Z, M_2)$
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 M_1 and M_2 . (You should plot them on the same graph.) Which model do you think is better? Explain your reasons.

Answer: The ROC curve for M_1 and M_2 are shown below.

M_1 is better, since its area under the ROC curve is larger than the area under ROC curve for M_2 .



ROC curve

- c) Repeat the analysis for part (b) using the same cutoff threshold on model M_2 . Compare the F-measure results for both models. Which model is better? Are the results consistent with what you expect from the ROC curve?

When $t = 0.5$, the confusion matrix for M1 is shown below.

	predicted	+	-
Actual	+	3	2
	-	1	4

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}) = 3/(3+1) = 3/4 = 75\%.$$

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) = 3/(3+2) = 3/5 = 60\%.$$

$$\text{F-measure} = 2rp/r+p = (2 \times .75 \times .6)/(.75 + .6) = 0.667.$$

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

	Predicted	+	-
Actual	+	1	4
	-	1	4

$$\text{Precision} = 1/1+1 = 1/2 = 50\%.$$

$$\text{Recall} = 1/1+4 = 1/5 = 20\%.$$

$$\text{F-measure} = 2rp/r+p = (2 \times .5 \times .2)/(.5 + .2) = 0.2857.$$

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

7) 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.

The screenshot shows the RStudio interface. The Console pane displays the output of the R session, including the installation of the randomForest package and the execution of the following R code:

```

> library(randomForest)
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
> f=randomForest(x,y)
> e = 1-sum(y==predict(f,x))/length(y)
> cat("misclassification error is :",e)
misclassification error is : 0
>

```

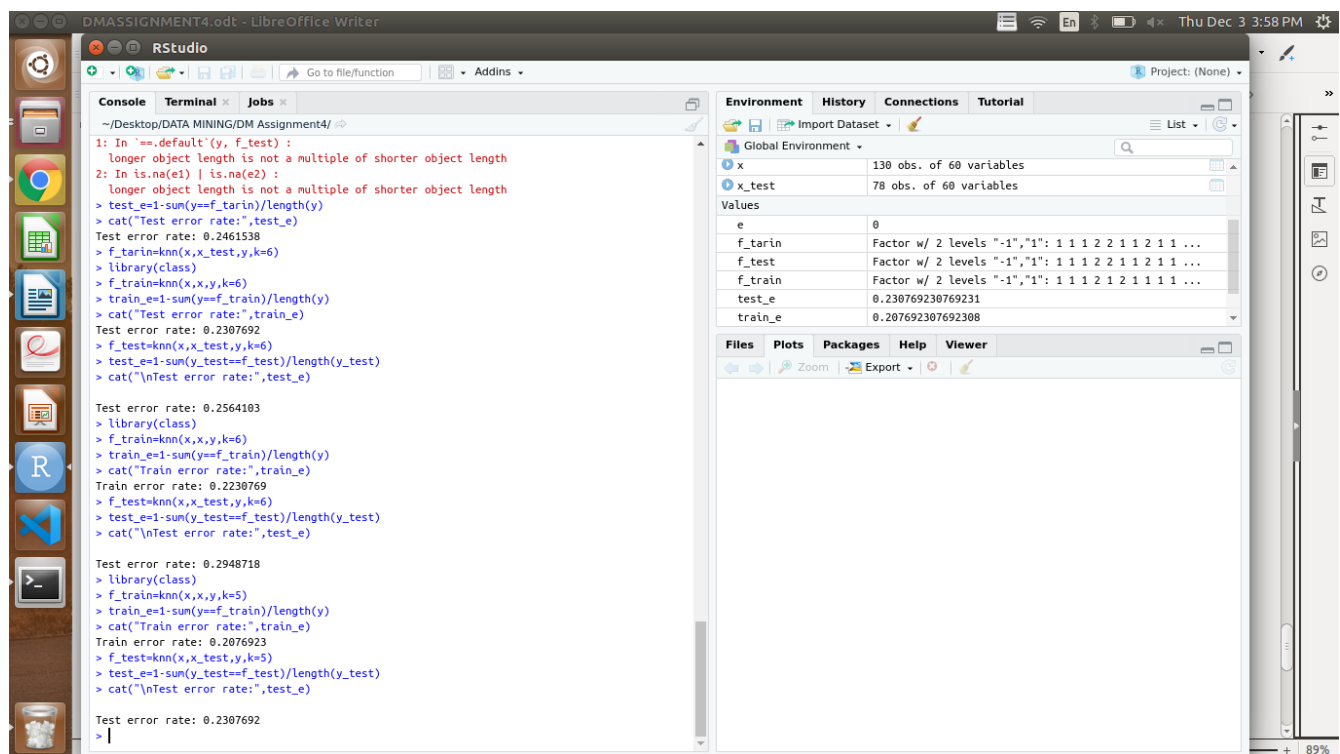
The Environment pane on the right shows the objects in the global environment:

Object	Type	Size
f	randomForest	List of 18
fit	randomForest	List of 14
testdata	data.frame	78 obs. of 61 variables
traindata	data.frame	130 obs. of 61 variables
x	matrix	130 obs. of 60 variables
x_test	matrix	78 obs. of 60 variables

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.

tt



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 k th 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.