

# Data Mining Assignment #6

Jeremy Keys

11/14/2017

## Part I

### Predict Treatment Outcome

**Step 1) Examine the records where TREATMENT\_RESPONSE is non-missing.**

First, I converted the CSV file to an ARFF file, using the Arff Viewer contained within Weka. Then I loaded the results into Weka.

1) There are 72 instances total. Of all the instances, 57 (or 79%) of the instances are missing the Treatment\_Response attribute; alternatively,  $72 - 57 = 13$  instances do contain the Treatment\_Response attribute. (See Figure 1.)

2) The records which are missing a value in the Treatment\_Response attribute can be identified by Source and Class. All of the records which contain a Treatment\_Response value originated from (had a Source value of) CALGB. None of the records with Source = 'St-Jude', 'DFCI', or 'CCG' contain a non-empty value for the Treatment\_Response attribute. Furthermore, all records which have at-

Name: Treatment_Response		Type: Nominal	
Missing: 57 (79%)		Distinct: 2	Unique: 0 (0%)
No.	Label	Count	Weight
1	Failure	8	8.0
2	Success	7	7.0

Figure 1:

Weka analysis

No.		Name
1	<input checked="" type="checkbox"/>	SNUM
2	<input type="checkbox"/>	FAB_if_AML
3	<input type="checkbox"/>	pct_Blasts
4	<input type="checkbox"/>	Treatment_Response
5	<input type="checkbox"/>	PS
6	<input type="checkbox"/>	D49950

Figure 2:  
The remaining description fields after removing useless fields

tribute Class = 'AML' contain a non-empty value for the Treatment\_Response attribute.

3) It is not correct to build predictive models on the attribute Treatment\_Response if there are records containing an empty value because you are effectively swapping the Class and Treatment\_Response (T\_R) fields. That means you would have 72 instances, with only 13 of those containing a valid Class value.

## Step 2) Select only the records with non-missing TREATMENT\_RESPONSE.

First, I converted the CSV file to an ARFF file, using the Arff Viewer contained within Weka. Then I loaded the results into Weka. I then used the filter RemoveWithValues to remove all records which did not have a non-empty T\_R value. Then, I used the Preprocess screen and the remove function to remove the attributes which fit the removal criteria described in hw6.pdf.

4) The sample description fields that should remain after removing the useless fields are: FAB\_if\_AML, pct\_Blasts, Treatment\_Response, and PS. (See Figure 2.)

## Step 3) Use J4.8 to build a model and use leave-one-out cross validation to measure the error rate.

5) Figure 3 shows the produced tree and error rate. The tree correctly classified 13/15 instances, for a 13.33% error rate.

6) Figure 4 shows the leave-one-out-cross-validation ranked attributes using Information Gain as the attribute evaluator. The top 10 ranked attributes are, in descending order (from best rank to worst): SNUM: 1 +/- 0, U82759: 2.1 +/- 0.25, FAB\_if\_AML: 3.3 +/- 0.44, M91432:

```

=== Classifier model (full training set) ===

J48 pruned tree
-----

U82759 <= 805: Success (8.0/1.0)
U82759 > 805: Failure (7.0)

Number of Leaves   :     2

Size of the tree :     3

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      13           86.6667 %
Incorrectly Classified Instances     2           13.3333 %
Kappa statistic                    0.7321
Mean absolute error                  0.2488
Root mean squared error             0.3857
Relative absolute error             47.0191 %
Root relative squared error         72.7678 %
Total Number of Instances          15

```

Figure 3:  
The results of running **J-48** classifier on genes-reduced.arff

```

=== Attribute selection 15 fold cross-validation (str:

average merit      average rank  attribute
0.993 +- 0.007      1 +- 0        1 SNUM
0.73 +- 0.103       2.1 +- 0.25    24 U82759
0.061 +- 0.048      3.3 +- 0.44     2 FAB_if_AML
0 +- 0              4.5 +- 0.88    16 M91432
0 +- 0              6.2 +- 2.4     14 M62762
0 +- 0              6.3 +- 0.44    15 M81933
0 +- 0              7.3 +- 0.44    18 U12471_cds1
0 +- 0              7.5 +- 1.26    17 S50223
0 +- 0              10 +- 0       19 U32944
0 +- 0              10 +- 1.67    12 M21551_rnal
0 +- 0              10.6 +- 0.8    13 M55150
0 +- 0              12.3 +- 1     11 L47738
0 +- 0              14 +- 2       21 U50136_rnal
0 +- 0              14.2 +- 0.75    6 D49950
0 +- 0              14.3 +- 5.64    7 D62222

```

Figure 4:  
Leave-one-out-cross-validation with **Info Gain** attribute evaluation and  
Ranker search method

=== Attribute selection 15 fold cross-validation (s

average merit	average rank	attribute
0.736 +- 0.104	1 +- 0	24 U82759
0.261 +- 0.002	2.3 +- 0.44	1 SNUM
0.034 +- 0.027	3.3 +- 0.44	2 FAB_if_AML
0 +- 0	4.5 +- 0.88	16 M91432
0 +- 0	6.2 +- 2.4	14 M62762
0 +- 0	6.3 +- 0.44	15 M81933
0 +- 0	7.3 +- 0.44	18 U12471_cdsl
0 +- 0	7.5 +- 1.26	17 S50223
0 +- 0	10 +- 0	19 U32944
0 +- 0	10 +- 1.67	12 M21551_rnal
0 +- 0	10.6 +- 0.8	13 M55150
0 +- 0	12.3 +- 1	11 L47738
0 +- 0	14 +- 2	21 U50136_rnal
0.109 +- 0.218	14.1 +- 6.04	7 D63880
0 +- 0	14.2 +- 0.75	6 D49950

Figure 5:

Leave-one-out-cross-validation with **Gain Ratio** attribute evaluation and Ranker search method

4.5 +/- 0.88, M62762: 6.2 +/- 2.4, M81933: 6.3 +/- 0.44, U12471\_cdsl: 7.3 +/- 0.44, S50223: 7.5 +/- 1.26, U32944: 10 +/- 0, and M21551\_rnal: 10 +/- 1.67. Because all records have a unique SNUM, the information gain by splitting on SNUM would be equal to the total information in the root, which is more than any other attribute. (Witten et al., p. 105) The obvious problem here is that SNUM is not a good attribute for predictive purposes (does not provide any actual information/predictive value), and should not be included in the cross-validation.

7) Figure 5 shows the leave-one-out-cross-validation ranked attributes using Gain Ratio as the attribute evaluator. Using Gain Ratio improves the situation (if not outright “solves”), because Gain Ratio corrects for the bias towards attributes with many different values, by biasing the gain ratios towards attributes with fewer daughter nodes. (Witten et al., p. 105) The top attribute is U82759 (the same attribute which J-48 splits on in Figure 3). The top 10 ranked attributes are, in descending order (from best rank to worst): U82759: 1 +/- 0, SNUM: 2.3 +/- 0.44, FAB\_if\_AML: 3.3 +/- 0.44, M91432: 4.5 +/- 0.88, M62762: 6.2 +/- 2.4, M81933: 6.3 +/- 0.44, U12471\_cdsl: 7.3 +/- 0.44, S50223: 7.5 +/- 1.26, U32944: 10 +/- 0, and M21551\_rnal: 10 +/- 1.67.

```

-----
Test mode:      15-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----

D63880 <= 131: Success (4.0)
D63880 > 131
|   L47738 <= 20: Success (3.0)
|   L47738 > 20: Failure (8.0)

Number of Leaves   :     3

Size of the tree   :     5

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          7           46.6667 %
Incorrectly Classified Instances        8           53.3333 %
Kappa statistic                        -0.0714
Mean absolute error                     0.5528
Root mean squared error                 0.7237
Relative absolute error                104.4619 %
Root relative squared error            136.5235 %
Total Number of Instances              15

```

Figure 6:

The results of running J-48 classifier on genes-reduced.arff, with attribute U82759 removed

## Step 4) Remove the top attribute obtained in Q7 and re-run J4.8.

I removed the top attribute with the Remove feature in the Preprocess tab.

8) Figure 6 shows the tree produced by J-48 after removing attribute U82759. The expected error rate is 8/15 incorrectly classified instances, or 53.33%.

9) Figure 7 shows the leave-one-out-cross-validation ranked attributes using Gain Ratio as the attribute evaluator. The top 10 ranked attributes are, in descending order (from best rank to worst): SNUM: 1.3 +/- 0.44, FAB\_if\_AML: 2.3 +/- 0.44, M91432: 3.9 +/- 1.45, M62762: 4.3 +/- 0.6, M81933: 4.7 +/- 0.44, S50223: 6.5 +/- 1.26,

=== Attribute selection 15 fold cross-validation (st

average merit	average rank	attribute
0.261 +- 0.002	1.3 +- 0.44	1 SNUM
0.034 +- 0.027	2.3 +- 0.44	2 FAB_if_AML
0 +- 0	3.9 +- 1.45	16 M91432
0 +- 0	4.3 +- 0.6	14 M62762
0 +- 0	4.7 +- 0.44	15 M81933
0 +- 0	6.5 +- 1.26	17 S50223
0 +- 0	7.3 +- 0.6	12 M21551_rnal
0 +- 0	8.3 +- 0.6	18 U12471_cds1
0 +- 0	9.1 +- 0.5	19 U32944
0 +- 0	9.9 +- 1.36	13 M55150
0 +- 0	11.3 +- 0.44	11 L47738
0 +- 0	12.3 +- 0.44	21 U50136_rnal
0.109 +- 0.218	13.1 +- 6.04	7 D63880
0 +- 0	13.9 +- 1.59	6 D49950
0 +- 0	14.3 +- 0.44	3 pct_Blasts
0 +- 0	15.3 +- 0.44	5 PS
0 +- 0	16.5 +- 1.26	10 L13278
0 +- 0	18.1 +- 0.25	8 J03473
0 +- 0	19.1 +- 0.25	9 J05243
0 +- 0	20.1 +- 0.25	20 U35451
0 +- 0	21.1 +- 0.25	42 U22125

Figure 7: Leave-one-out-cross-validation, with attribute U82759 removed, with **Gain Ratio** attribute evaluation and Ranker search method

**M21551\_rnal: 7.3 +/- 0.6, U12471\_cdsl: 8.3 +/- 0.6, U32944: 9.1 +/- 0.5, and M55150: 9.9 +/- 1.36.**

**10) When the highest ranked attribute by Gain Ratio (U82759) was removed from the dataset, the produced tree jumped from an expected 13.33% error rate (Figure 3) to a 53.33% error rate (Figure 6). This result shows that there is a distinct correlation between the rankings of attributes by gain ratio and the quality of the produced model (as measured by expected error rates).**