Michael Fitzgerald
February 10, 2016
Homework 1
CAP 6673
Dr. Khoshgoftaar

IA Engineering the input: Preparing the datasets

fit.arff	test.arff	fitclass.arff	test.arff
`@relation Fit`	`@relation Test`	`@relation FitClass`	`@relation TestClass`
`@attribute NUMUORS real`	`@attribute NUMUORS real`	`@attribute NUMUORS real`	`@attribute NUMUORS real`
`@attribute NUMUANDS real`	`@attribute NUMUANDS real`	`@attribute NUMUANDS real`	`@attribute NUMUANDS real`
`@attribute TOTOTORS real`	`@attribute TOTOTORS real`	`@attribute TOTOTORS real`	`@attribute TOTOTORS real`
`@attribute TOTOPANDS real`	`@attribute TOTOPANDS real`	`@attribute TOTOPANDS real`	`@attribute TOTOPANDS real`
`@attribute VG real`	`@attribute VG real`	`@attribute VG real`	`@attribute VG real`
`@attribute NLOGIC real`	`@attribute NLOGIC real`	`@attribute NLOGIC real`	`@attribute NLOGIC real`
`@attribute LOC real`	`@attribute LOC real`	`@attribute LOC real`	`@attribute LOC real`
`@attribute ELOC real`	`@attribute ELOC real`	`@attribute ELOC real`	`@attribute ELOC real`
`@attribute FAULTS real`	`@attribute FAULTS real`	`@attribute CLASS {fp,nfp}`	`@attribute CLASS {fp,nfp}`
`@data`	`@data`	`@data`	`@data`
`22,85,203,174,9,0,362,40,0`	`6,12,127,45,10,0,641,55,0`	`22,85,203,174,9,0,362,40,nfp`	`6,12,127,45,10,0,641,55,nfp`
`21,87,186,165,5,0,379,32,0`	`5,5,41,12,1,0,407,17,0`	`21,87,186,165,5,0,379,32,nfp`	`5,5,41,12,1,0,407,17,nfp`
`30,107,405,306,25,0,756,99,0`	`23,28,95,66,4,2,241,20,0`	`30,107,405,306,25,0,756,99,nfp`	`23,28,95,66,4,2,241,20,nfp`
⋮	⋮	⋮	⋮
`20,63,311,259,10,10,603,52,2`	`28,173,632,581,86,0,763,135,2`	`20,63,311,259,10,10,603,52,fp`	`28,173,632,581,86,0,763,135,fp`
`22,77,227,201,13,5,289,38,2`	`37,188,1454,1300,71,0,1189,244,2`	`22,77,227,201,13,5,289,38,fp`	`37,188,1454,1300,71,0,1189,244,`
⋮	⋮	⋮	⋮

In the table above are the files "fit.arff," "test.arff," "fitclass.arff," and "testclass.arff", labeled respectively. I have include the " relation" labeled with respect to each file, have included the block defining the nine attributes for the four files and the first rows of data followed by "...", and then two rows of data where the number of faults is equal to two or greater to illustrate where class switched from nfp to fp. This ninth assigned attribute is the only difference between the two .arff files for each fit and test respectively and I assigned this "CLASS" attribute as a categorical data where if faults is less than or equal to 1 it gets an assignment of "nfp," otherwise it receives "fp" (the number of faults is non-negative).

IB Modeling assignment: Prediction

Fit				
	No Attribute Selection	M-5 method	Greedy	Decision Stump
Linear Regression Model/ Classifier Model (FAULTS=)	$=-0.0517\cdot(\text{NUMUORS})+0.0341\cdot(\text{NUMANDS})$ $-0.0026\cdot(\text{TOTOTORS})+0\cdot(\text{TOTOPANDS})$ $-0.0372\cdot(\text{VG})+0.2118\cdot(\text{NLOGIC})$ $+0.0018\cdot(\text{LOC})+0.005\cdot(\text{ELOC})$ -0.309	$=-0.0516\cdot(\text{NUMUORS})+0.0341\cdot(\text{NUMANDS})$ $-0.0027\cdot(\text{TOTOTORS})-0.0372\cdot(\text{VG})$ $+0.2119\cdot(\text{NLOGIC})+0.0018\cdot(\text{LOC})$ $+0.005\cdot(\text{ELOC})-0.3091$	$=-0.0482\cdot(\text{NUMUORS})+0.0336\cdot(\text{NUMANDS})$ $-0.0021\cdot(\text{TOTOTORS})-0.0337\cdot(\text{VG})$ $+0.2088\cdot(\text{NLOGIC})+0.0019\cdot(\text{LOC})$ $+0.005\cdot-0.3255$	NLOGIC$\leq$14.0: 1.3806818181818181 NLOGIC$>$14.0: 15.333333333333334 NLOGIC is missing:2.271276595744681
Correlation coefficient	0.7969	0.7935	0.7961	0.5941
Mean absolute error	1.6902	1.7017	1.6939	2.2173
Root mean squared error	2.8362	2.8612	2.8425	3.7905
Relative absolute error	58.4755%	58.8734%	58.6027%	76.7091%
Root relative squared error	60.8616%	61.3972%	60.977%	81.3406%
Total Number of Instances	188	188	188	188

Test				
	No Attribute Selection	M-5 method	Greedy	Decision Stump
Correlation coefficient	0..829	0.829	0.8314	0.7111
Mean absolute error	1.8377	1.836	1.8383	2.2952
Root mean squared error	3.7317	3.7324	3.6895	4.1928
Relative absolute error	58.6426%	58.6423%	58.6625%	73.2439%
Root relative squared error	63.6881%	63.7%	62.968%	71.5571%
Total Number of Instances	94	94	94	94

In the first part of the above table, labeled "Fit," is a summary of the output for the four desired models (labeled as columns of the first row) and their performance measures for each respective model built on the fit data prepared for prediction. In the second part of the above table, labeled "Test," is a summary of the performance measures using the model trained on the fit data set and used for prediction on the test data. The test set uses the same regression equation (classifier model in the case of "Decision Stump") for the test set so I have omitted this from the table. For the fit data set, there are nine attributes and 188 instances; and for the test data set, there are nine attributes but 94 instances.

The first column is the linear regression model using the "No Attribute Selection" for model selection. This model uses all eight attributes for prediction of the ninth attribute (FAULTS), but as Matthew pointed out in his presentation, the TOTOPANDS is multiplied by 0 and it follows regardless of the value for TOTOPANDS it will be multiplied by zero therefore not influencing the model for prediction.

The second column is the linear regression model using the "M-5 method" for model selection. This model uses seven of the eight attributes for prediction of the ninth attribute (FAULTS) omitting the variable (attribute) TOTOPANDS. Consequently, the linear regression equation is nearly identical to the model using "No Attribute Selction", with coefficients for each attribute equal or within ± 0.0001 (likely a result of TOTOPANDS' coefficient in "No Attribute Selection" being rounded to 0 when it is likely non-zero but very small).

The third column is the linear regression model using the "Greedy" method for atrribute selection which used six of the eight attributes for prediction of the ninth attribute faults, omitting TOTOPANDS and ELOC. This linear regression model does differ some from the prior two but the coefficient's of any attribute differ by at most ± 0.0035 and the intercept differ by ± 0.0164. For this reason these three models are all quite similar.

The fourth column uses the Decision Stump classifier in Weka, and this model used only one of the eight attributes, NLOGIC, for prediction of the ninth attribute FAULTS. It predicts a value for the number of faults given the value of the attribute NLOGIC. If NLOGIC is less than or equal to 14 it predicts there will be 1.3806818181818181 many faults in the module (or if we round, 1 fault since faults are integer values); if If NLOGIC is greater than 14 it predicts there will be 15.333333333333334 many faults in the module (or 15 faults for similar reasoning) and if NLOGIC does not have an assignment it predicts 2.271276595744681 many faults in the module (or 2 faults when rounded to the nearest whole number). For this reason, this model is very crude as it only uses one attribute(variable) for prediction.

Comparing models, it is immediately obvious that the Decision Stump classifier has the worst performance measures since it has the lowest correlation coefficient of the four models ($r = .5941$) as well as the highest error numbers for all four performance measures related to error. As noted above this is likely a result of this model only using one variable, NLOGIC, for prediction.

The remaining three linear regression models have very similar performance measures, which is not too surprising as discussed above, they have similar regression equations used for prediction. Using the performance measures it appears "No Attribute Selection" is the best model since it has the highest correlation coefficient and lowest error measure for three of the four error performance measurements. Between the remaining two models, "M-5 method" and "Greedy", the linear regression model using "Greedy for model selection is preferred as it has better performance measures in all measurements Weka outputs. This is somewhat surprising since the "No Attribute Selection" model is nearly identical to the "M-5 method" model but somehow the"Greedy" model is between the two. This could be for two reasons.

Primarily it is possible that the three linear regression models are so similar the are nearly indistinguishable from one another up to some error and secondly including addition variables can sometimes influence the performance measures. I explained the first reason extensively above. For the second reason, in regression including superfluous variables (attributes) is highly discouraged for several reasons including, the variable may be statistically significant due to random chance and or the inclusion of an addition variable may have a confounding effect on another variable already in the model. Weka does not output p-values but the "M-5 method" and "Greedy" method both chose not to include the attribute TOTOPANDS which suggest to me this variable is likely insignificant and should not be included in the model for prediction.

To summarize, I followed Matthew's procedure in Weka using the prepared documents form **part IA** so I will not elaborate on the methodology. The model using the Decision Stump classifier was clearly the worst model likely since it only uses one attribute for prediction. Of the remaining three linear regression models, The No Attribute Selection model had the highest performance measures (with the exception of Root mean square error) but for the reason discussed above the inclusion of the attribute TOTOPANDS in the No Attribute Selection model suggest perhaps the Greedy model is in fact the best. However, it should be noted that the coefficients and intercepts of the regression equation of the No Attribute Selection model is nearly identical to the M-5 method model and that the inclusion of the variable TOTOPANDS does not negatively effect the performance errors (deviation).

In the second part of the table, Weka used the prediction model from the fit data to predict the number of faults in the 94 instances of the test data, as well as outputting the performance measures using the actual faults data in test. For four of five performance measures, the Greedy model was the best with an even higher correlation coefficient than suggest from the fit model but with greater error measures. The No Attribute Selection and M-5 method model are nearly identical since again they have nearly identical coefficients and intercepts. The Decision Stump classifier again performed the poorest of the four. Finally, all four models had higher correlation coefficients (predicts the test data set better than the fit data set) but also high error measures (deviation).

Michael Fitzgerald
February 24, 2016
Homework 2
CAP 6673
Dr. Khoshgoftaar

	Part 1	**Part 2**	**Part 3**
Unpruned	False	True	False
Confidence Factor	0.25	0.25	0.01
Number of Nodes	6	7	3
Number of Leaves	8	9	4
Confusion Matrix	$\begin{array}{cc}44&11\\12&121\end{array}$	$\begin{array}{cc}44&11\\12&121\end{array}$	$\begin{array}{cc}40&15\\13&120\end{array}$
Type I Misclassification	$\frac{12}{133}$=0.09	$\frac{12}{133}$=0.09	$\frac{13}{133}$=0.098
Type II Misclassification	$\frac{11}{55}$=0.2	$\frac{11}{55}$=0.2	$\frac{15}{55}$=0.273
Correctly Classified Instances	165	165	160
Correctly Classified %	87.766%	87.766%	85.1064%
Incorrectly Classified Instances	23	22	28
Incorrectly Classified %	12.234%	12.234%	14.8936%
TEST			
Confusion Matrix	$\begin{array}{cc}19&9\\5&61\end{array}$	$\begin{array}{cc}20&8\\5&61\end{array}$	$\begin{array}{cc}17&11\\4&62\end{array}$
Type I Misclassification	$\frac{5}{66}$=0.076	$\frac{5}{66}$=0.076	$\frac{4}{66}$=0.061
Type II Misclassification	$\frac{9}{28}$=0.321	$\frac{8}{28}$=0.286	$\frac{11}{28}$=0.393
Correctly Classified Instances	80	81	79
Correctly Classified %	85.1064%	86.1702%	84.0426%
Incorrectly Classified Instances	14	13	15
Incorrectly Classified %	14.8936%	13.8298%	15.9574%

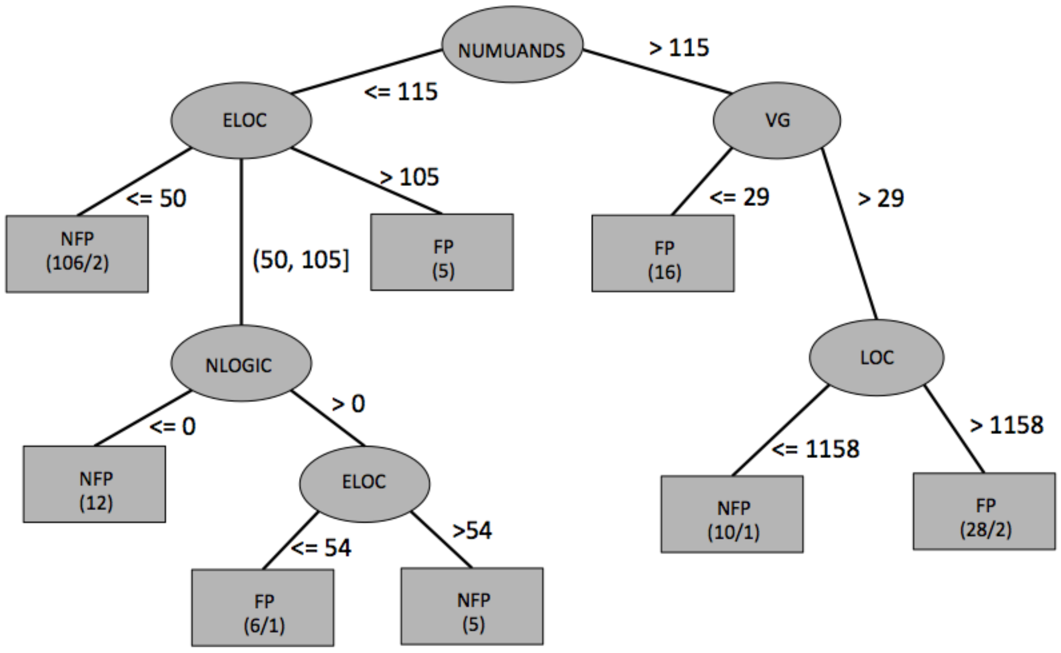
Part 1:

From the table above in the column labeled "**Part 1**," where the default settings are unpruned equals false and the confidence factor is equal to 0.25, Weka creates a decision tree with six nodes and eight leaves. Now it should be noted that Weka output the size of the tree as 15 but this is a result of the "ELOC" attribute being spilt into three branches, ≤ 50, in the interval $(50, 105]$, and > 105 (See diagram below). Weka outputs one decision node split on the numeric value "50" for ELOC and the the next node is for when ELOC > 50 is again a node using the attribute ELOC. I have combined this into a single node.

Below is the representation of the decision tree. Each circle (oval) is a node with an attribute to be tested in it. Working from the top down an instance is classified by its numeric value at each node, where the numbers for each branch are immediately to the right of each branch or line segment. For example, the first attribute to be tested is "NUMUANDS;" if the value for this attribute is ≤ 115 (less than or equal to 115) we continue down the left branch in the diagram below the tree to the next node. Otherwise "NUMUANDS" is greater than 115 and we go down the right branch to the next node. We continue this process until we arrive at a rectangular box (representing a leaf) where our instance is classified as "FP" or "NFP" by which ever is in the box.

The first number in each leaf box represent the number of the 188 instances that meet the conditions leading to that box and thus classified as whats denoted in the box. The second number represents the number of instances that were misclassified but met the conditions leading to the box. If there is no second number, then that specific leaf has classified all instances correctly.

Again from the table above we see the confusion matrix, the table also indicates that our Type I error or misclassification is equal to 0.09 and our Type II error or misclassification is equal to 0.2. The model correctly classifies 165 of the 188 instances or $\approx 88\%$ and misclassifies 23 of the 188 instances or $\approx 12\%$. The second part of the table above, below the heading "TEST," represents the classification of the Test data using the model for classification built using the fit data set. From the table, we see our Type I error or misclassification is equal to 0.076 and our Type II error or misclassification is equal to 0.321. The model correctly classifies 80 of the 94 instances or $\approx 85\%$ and misclassifies 14 of the 94 instances or $\approx 15\%$.



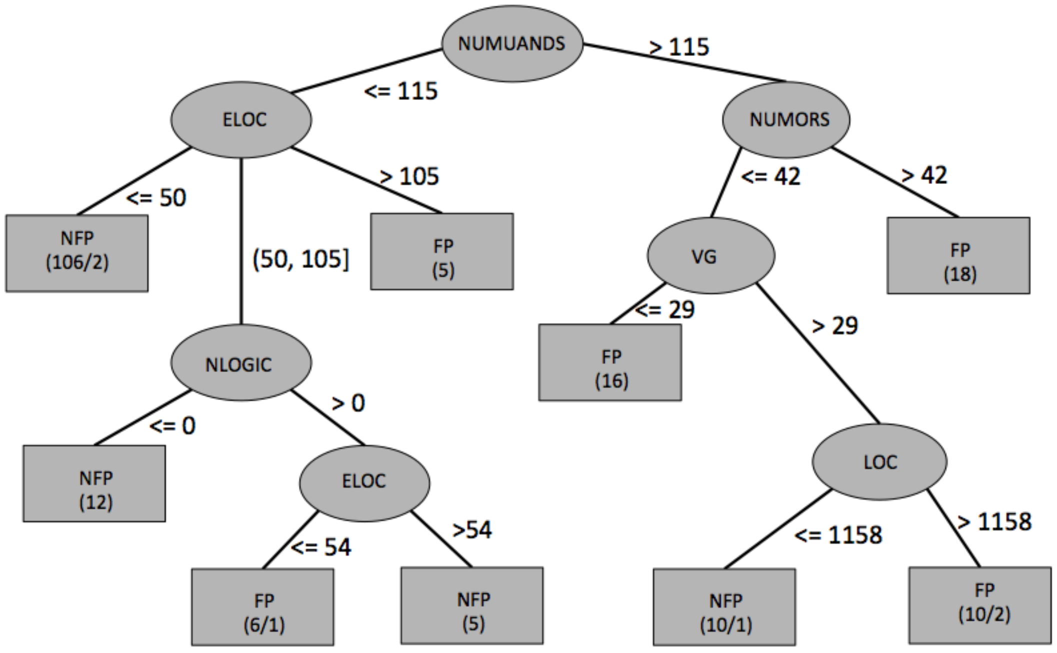
Part 2:

Repeating the procedure from **Part 1**, but changing the settings in J48 to unpruned equal true yields the results in the column of the table above labeled "**Part 2**." The confidence factor remains equal to the default 0.25. As we can see from the table, the resulting decision tree has seven nodes and nine leaves. As with before, Weka outputs the size of the tree as equal to 17 but again this is a result of it treating the splitting of "ELOC" into three branches (on the left branch where NUMUANDS ≤ 115) as two separate nodes implying Weka has the decision tree as having 8 nodes. As in part one, I have combined this into one node with three branches, where "ELOC" is ≤ 50, in the interval $(50, 105]$, or > 105. The decision tree is included below and is quite similar to the decision tree from **Part I** but has the additional node testing the attribute "NUMORS" if initially "NUMUANDS" is greater than 115.

From the above table, we can see that **Part 2** has identically the same Confusion Matrix and consequently, Type I Misclassification, Type II Misclassification, Correctly Classified Instances, Incorrectly Classified Instances, and their respective percentages are identical for the fit data set.

The Test using the model for classification trained on the Fit data set however differs. The model correctly classifies 81 of the 94 instances ($\approx 86\%$) and incorrectly classifies 13 of the 19 instances ($\approx 14\%$). The Type I error is the same as in **Part 1** but the Type II error is equal to 0.286 as it correctly classified one additional "fault-prone" instance as fault-prone compared to the model in **Part 1**. This is fairly surprising since the unpruned tree generally is a worse model as it tends to over-fit the training set but actually in this case, the unpruned tree in fact performed better on the test data set.

Comparing the tree in **Part 2** to the tree generated in **Part I** they are quite similar but the unpruned tree (**Part 2**) has one additional node. As the textbook outlines on page 192-193; C4.5 (equivalently J48) subtree raising is often used in the decision tree-building system C4.5. Again comparing the two decision trees we see that the pruned tree from **Part I** has the subtree from the node "VG" raised to replace the node "NUMORS" in the unpruned tree. Comparing the number we see that the resulting 18 correctly classified instance where NUMORS > 42 in the unpruned tree are classified in the same leaf at the bottom of the pruned tree since all 18 of these instances meet the qualifications VG > 29 and LOC > 1158. As a result, all 18 of these instances in the pruned tree are also correctly classified as fault-prone leading to identical confusion matrices for **Part 1 & 2** on the fit data set. It should be noted though that this is note always the case.

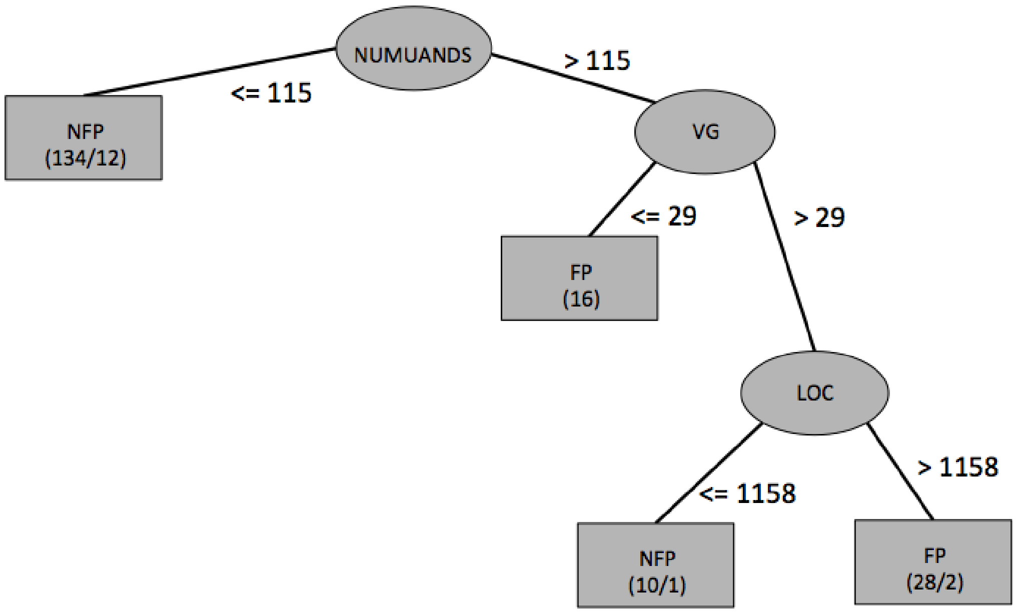


Part 3:

From the table above in the column labeled "**Part 3**," where the default setting for unpruned equals false but the confidence factor is changed to 0.01, Weka creates a decision tree with three nodes and four leaves. The main difference as you can see in the decision tree below is that all instance that have the attribute NUMUANDS $\leq$ 115 are classified as non-fault prone. Consequently, we do not have the problem of Weka splitting the node "ELOC" twice as in **Part 1 & 2**.

From the table, above we observe the confusion matrix for **Part 3**, which indicates that our Type I error or misclassification is equal to 0.098 and our Type II error or misclassification is equal to 0.273. The model correctly classifies 160 of the 188 instances ($\approx$ 85%) and misclassifies 28 of the 188 instances ($\approx$ 15%).

The second part of the table above, as before, represents the classification of the test data set using the model for classification built on the fit data set. From the table, we see our Type I error or misclassification is equal to 0.061 and our Type II error or misclassification is equal to 0.3293. The model correctly classifies 79 of the 94 instances ($\approx$ 84%) and misclassifies 15 of the 94 instances($\approx$ 16%).



The C4.5 (J48) classifier uses estimated error to decide whether or not to pruned either replacing nodes with leafs or subtree raising (as was done in **Part 2**). It uses an ad hoc method but rather than copy paragraphs from the book, I will just include the estimated error formula at each node (or rather a upper confidence limit)

$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

Here E is the numbers of error out of the total instance N, $f = \frac{E}{N}$, and z the related z-score from the standard normal distribution. As c, the confidence factor shrinks, z increases, and in the error calculation above e shrinks as a result of the denominator growing faster than the numerator. This error is compared to the parent node and the subtree node is eliminated if the error of the parent tree is smaller. On page 199, the textbook summarizes this with altering the confidence factor to a lower value in C4.5 results in more drastic pruning. This fact is reflected in our example where changing the confidence factor from 0.25 to 0.01 results in drastic pruning of the initial left branch from **Part 1**. The size of the tree decrease from 6 nodes and 8 leaves to 3 nodes and 4 leaves when the confidence factor is decreased. As a result, the number of instances misclassified increases as we have made a simpler decision tree. Additionally, using our independent test set on the model create from the fit set, we see that we have more misclassified instances then in **Part 1 & 2**, implying that changing the confidence factor resulted in an over-pruned decision tree. The Type I error however did decrease while the Type II error increased in the Test set.

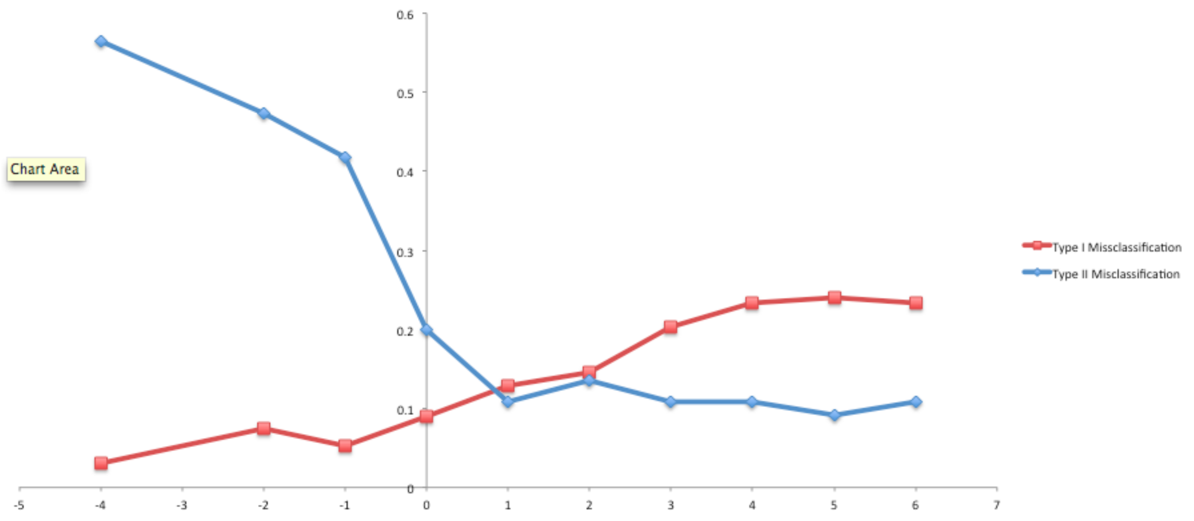
Part 4:

Cost Matrix	$\begin{array}{cc}0&\frac{1}{5}\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&\frac{1}{3}\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&\frac{1}{2}\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&1\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&2\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&3\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&4\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&5\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&6\\ \hline 1&0\end{array}$	$\begin{array}{cc}0&7\\ \hline 1&0\end{array}$
Confusion Matrix	$\begin{array}{cc}24&31\\ \hline 4&129\end{array}$	$\begin{array}{cc}29&26\\ \hline 10&123\end{array}$	$\begin{array}{cc}32&23\\ \hline 7&126\end{array}$	$\begin{array}{cc}44&11\\ \hline 12&121\end{array}$	$\begin{array}{cc}49&6\\ \hline 17&116\end{array}$	$\begin{array}{cc}47&8\\ \hline 18&115\end{array}$	$\begin{array}{cc}49&6\\ \hline 27&106\end{array}$	$\begin{array}{cc}49&6\\ \hline 31&102\end{array}$	$\begin{array}{cc}50&5\\ \hline 32&101\end{array}$	$\begin{array}{cc}49&6\\ \hline 31&101\end{array}$
Type II Misclassification	0.564	0.473	0.418	0.2	0.109	0.135	0.109	0.109	0.091	0.109
Type I Misclassification	0.03	0.075	0.053	0.09	0.128	0.145	0.203	0.233	0.241	0.233
Correctly Classified Instances	153	152	158	165	165	162	155	151	151	151
Correctly Classified %	81.383%	80.8511%	84.0426%	87.766%	87.766%	86.1702%	82.4468%	80.3191%	80.3191%	80.3191%
Incorrectly Classified Instances	35	36	30	23	23	26	33	37	37	37
Incorrectly Classified %	18.617%	19.1489%	15.9574%	12.234%	12.234%	13.8298%	17.5532%	19.6809%	19.6809%	19.6809%
Kappa Statistic	0.4746	0.4942	0.5785	0.706	0.7208	0.6828	0.6186	0.5805	0.5844	0.5805
Mean Absolute Error	0.1997	0.2234	0.1953	0.1452	0.1473	0.1597	0.2029	0.2262	0.2251	0.2275
Root Mean Square Error	0.3997	0.4038	0.3739	0.3344	0.3216	0.3526	0.3901	0.4125	0.4144	0.4181
Relative Absolute Error	48.1083%	53.8234%	47.0571%	34.9951%	35.4822%	38.4884%	48.8781%	54.493%	54.2337%	54.8163%
Root Relative Square Error	87.8252%	88.7268%	82.1608%	73.4704%	70.6674%	77.4891%	85.7109%	90.6341%	91.0558%	91.8731%
TR Rate	$\begin{array}{c}0.436\\ \hline 0.97\\ \hline 0.814\end{array}$	$\begin{array}{c}0.527\\ \hline 0.925\\ \hline 0.809\end{array}$	$\begin{array}{c}0.582\\ \hline 0.947\\ \hline 0.84\end{array}$	$\begin{array}{c}0.8\\ \hline 0.91\\ \hline 0.878\end{array}$	$\begin{array}{c}0.891\\ \hline 0.972\\ \hline 0.878\end{array}$	$\begin{array}{c}0.855\\ \hline 0.865\\ \hline 0.862\end{array}$	$\begin{array}{c}0.891\\ \hline 0.797\\ \hline 0.824\end{array}$	$\begin{array}{c}0.891\\ \hline 0.767\\ \hline 0.803\end{array}$	$\begin{array}{c}0.909\\ \hline 0.759\\ \hline 0.803\end{array}$	$\begin{array}{c}0.891\\ \hline 0.767\\ \hline 0.803\end{array}$
FP Rate	$\begin{array}{c}0.03\\ \hline 0.564\\ \hline 0.408\end{array}$	$\begin{array}{c}0.075\\ \hline 0.473\\ \hline 0.356\end{array}$	$\begin{array}{c}0.053\\ \hline 0.418\\ \hline 0.311\end{array}$	$\begin{array}{c}0.09\\ \hline 0.2\\ \hline 0.168\end{array}$	$\begin{array}{c}0.128\\ \hline 0.109\\ \hline 0.115\end{array}$	$\begin{array}{c}0.135\\ \hline 0.145\\ \hline 0.142\end{array}$	$\begin{array}{c}0.203\\ \hline 0.109\\ \hline 0.137\end{array}$	$\begin{array}{c}0.233\\ \hline 0.109\\ \hline 0.145\end{array}$	$\begin{array}{c}0.241\\ \hline 0.091\\ \hline 0.135\end{array}$	$\begin{array}{c}0.233\\ \hline 0.109\\ \hline 0.145\end{array}$
Precision	$\begin{array}{c}0.857\\ \hline 0.806\\ \hline 0.821\end{array}$	$\begin{array}{c}0.744\\ \hline 0.826\\ \hline 0.802\end{array}$	$\begin{array}{c}0.821\\ \hline 0.846\\ \hline 0.838\end{array}$	$\begin{array}{c}0.786\\ \hline 0.917\\ \hline 0.878\end{array}$	$\begin{array}{c}0.742\\ \hline 0.951\\ \hline 0.89\end{array}$	$\begin{array}{c}0.723\\ \hline 0.935\\ \hline 0.873\end{array}$	$\begin{array}{c}0.645\\ \hline 0.946\\ \hline 0.858\end{array}$	$\begin{array}{c}0.613\\ \hline 0.944\\ \hline 0.847\end{array}$	$\begin{array}{c}0.61\\ \hline 0.953\\ \hline 0.852\end{array}$	$\begin{array}{c}0.613\\ \hline 0.944\\ \hline 0.847\end{array}$
TEST										
Confusion Matrix	$\begin{array}{cc}6&22\\ \hline 0&66\end{array}$	$\begin{array}{cc}9&19\\ \hline 0&66\end{array}$	$\begin{array}{cc}20&8\\ \hline 6&60\end{array}$	$\begin{array}{cc}19&9\\ \hline 5&61\end{array}$	$\begin{array}{cc}20&8\\ \hline 10&56\end{array}$	$\begin{array}{cc}20&8\\ \hline 8&58\end{array}$	$\begin{array}{cc}27&1\\ \hline 19&47\end{array}$	$\begin{array}{cc}27&1\\ \hline 19&47\end{array}$	$\begin{array}{cc}27&1\\ \hline 19&47\end{array}$	$\begin{array}{cc}27&1\\ \hline 47&19\end{array}$
Type II Misclassification	0.786	0.679	0.286	0.321	0.286	0.286	0.036	0.036	0.036	0.036
Type I Misclassification	0	0	0.091	0.076	0.152	0.121	0.288	0.288	0.288	0.288
Correctly Classified Instances	72	75	80	80	76	78	74	74	74	74
Correctly Classified %	76.5957%	79.7872%	85.1064%	85.1064%	80.8511%	82.9787%	78.7234%	78.7234%	78.7234%	78.7234%
Incorrectly Classified Instances	22	19	14	14	18	16	20	20	20	20
Incorrectly Classified %	23.4043%	20.2128%	14.8936%	14.8936%	19.1489%	17.0213%	21.2766%	21.2766%	21.2766%	21.2766%

Observing the table above we see what happen when the cost of Type II errors is increase and decreases relative to the cost of Type I. As the questions states, we want to get the Type I error approximately equal to the Type II error with Type II error as low as possible. If we start with the weights 1 and 1 we have identical the model from **Part I**. Now if we decrease the cost of the Type II error we see that Type I error decrease while Type II error increases drastically(moving left in columns of the table from the initial 1 to 1 weight found on the off diagonal of the confusion matrix). Moving in the other direction (right in columns of the table), increasing the cost of the Type II error relative to the Type I error we see that the errors are "close" relatively for when cost of Type II is between 1-3 times that of Type I errors (columns 5,6, and 7 of the above table). After this Type I error becomes larger than Type II and the two stay approximately the same for when the cost of Type II is 4 to 7 times that of Type I. Though it is not included in the table above, I continued testing when the cost Type II error is an even larger multiple of Type I error and the two values continued to diverge.

Below is the a graph of the errors, labeled accordingly in the legend. The x-axis is the ratio of the weight of Type II to Type I ($\frac{C_{II}}{C_I}$) and the y-axis the error for each. Note that the scale of the x-axis is logarithmic since I want the graph to be centered around when the cost were equal ($log(1) = 0$) and so that the fractions are equally spaced relative to the whole numbers.

From the graph we see that the curves intersect between the ratio $\frac{1}{1}$ and $\frac{2}{1}$. Testing in this interval the optimal weighting appears to be when the cost of Type II error is 1.5 times as large as Type I error ($\frac{1.5}{1}$). The resulting optimal output is in the table following the graph. The Type I error equals 0.135 and the Type II error equals 0.127. From the graph we see that the curves become close when Type II error is 3 times that of Type I error but the Type II error in this case is equal to 0.135. Thus we prefer the ratio $\frac{1.5}{1}$, since Type II error are lower in this case. Running this cost ratio on the test data however yield the same output as in the table where Type II is half that of Type I $\left(\begin{array}{c|c} 0 & \frac{1}{2} \\ \hline 1 & 0 \end{array}\right)$. It should be noted that the test data is half the size of fit data so misclassification of one instances impacts the Type I error and Type II error more significantly.



	Confusion Matrix	Type I Misclassification	Type II Misclassification	Correctly Classified Instances	% Correctly Classified Instances	Incorrectly Classified Instances	% Incorrectly Classified Instances
$\begin{array}{c\|c} 0 & 1.5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 48 & 7 \\ \hline 18 & 115 \end{array}$	0.135	0.127	163	86.7021%	25	13.2979%

Michael Fitzgerald
March 23, 2016
Homework 3
CAP 6673
Dr. Khoshgoftaar

J48 with Bagging 10 Iterations

Cost Matrix	$\begin{matrix}0 & \frac{1}{5}\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & \frac{1}{2}\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 1\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 2\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 2.95\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 3\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 4\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 10\\ 1 & 0\end{matrix}$
Confusion Matrix	$\begin{matrix}22 & 33\\ 4 & 129\end{matrix}$	$\begin{matrix}37 & 18\\ 9 & 124\end{matrix}$	$\begin{matrix}41 & 14\\ 12 & 121\end{matrix}$	$\begin{matrix}46 & 9\\ 15 & 118\end{matrix}$	$\begin{matrix}48 & 7\\ 19 & 114\end{matrix}$	$\begin{matrix}47 & 8\\ 20 & 113\end{matrix}$	$\begin{matrix}49 & 6\\ 23 & 110\end{matrix}$	$\begin{matrix}50 & 5\\ 31 & 102\end{matrix}$
Type II Misclassification	0.6	0.327	0.255	0.164	0.127	0.145	0.109	0.091
Type I Misclassification	0.03	0.068	0.09	0.113	0.143	0.15	0.173	0.233
Correctly Classified Instances	151	161	162	164	162	160	159	152
Correctly Classified %	80.3191%	85.6383%	86.1702%	87.234%	86.1702%	85.1064%	84.5745%	80.8511%
Incorrectly Classified Instances	37	27	26	24	26	28	29	36
Incorrectly Classified %	19.6809%	14.3617%	13.8298%	12.766%	13.8298%	14.8936%	15.4255%	19.1489%
Kappa Statistic	0.4376	0.6356	0.6623	0.7012	0.686	0.6618	0.6583	0.5937
Mean Absolute Error	0.2109	0.1783	0.1659	0.1671	0.1712	0.1705	0.1754	0.2216
Root Mean Square Error	0.3823	0.3294	0.3145	0.3157	0.3235	0.325	0.3365	0.3957
Relative Absolute Error	50.8041%	42.9464%	39.9775%	40.2615%	41.2521%	41.0877%	42.2475%	53.4006%
Root Relative Square Error	83.9956%	72.3716%	69.103%	69.3677%	71.078%	71.4109%	73.9365%	86.9561%
TR Rate	0.4 / 0.97 / 0.803	0.673 / 0.932 / 0.856	0.745 / 0.91 / 0.862	0.836 / 0.887 / 0.872	0.873 / 0.857 / 0.862	0.855 / 0.85 / 0.851	0.891 / 0.827 / 0.846	0.909 / 0.767 / 0.809
FP Rate	0.03 / 0.6 / 0.433	0.068 / 0.327 / 0.251	0.09 / 0.255 / 0.206	0.113 / 0.164 / 0.149	0.143 / 0.127 / 0.132	0.15 / 0.145 / 0.147	0.173 / 0.109 / 0.128	0.233 / 0.091 / 0.133
Precision	0.846 / 0.796 / 0.811	0.804 / 0.873 / 0.853	0.774 / 0.896 / 0.86	0.754 / 0.929 / 0.878	0.716 / 0.942 / 0.876	0.701 / 0.934 / 0.866	0.681 / 0.948 / 0.87	0.617 / 0.953 / 0.855
Recall	0.4 / 0.97 / 0.803	0.673 / 0.932 / 0.856	0.745 / 0.91 / 0.862	0.836 / 0.887 / 0.872	0.873 / 0.857 / 0.862	0.855 / 0.85 / 0.851	0.891 / 0.827 / 0.846	0.909 / 0.767 / 0.809
TEST								
Confusion Matrix	$\begin{matrix}11 & 17\\ 0 & 66\end{matrix}$	$\begin{matrix}17 & 11\\ 4 & 62\end{matrix}$	$\begin{matrix}21 & 7\\ 6 & 60\end{matrix}$	$\begin{matrix}25 & 3\\ 9 & 57\end{matrix}$	$\begin{matrix}24 & 4\\ 9 & 57\end{matrix}$	$\begin{matrix}25 & 3\\ 8 & 58\end{matrix}$	$\begin{matrix}26 & 2\\ 11 & 55\end{matrix}$	$\begin{matrix}27 & 1\\ 19 & 47\end{matrix}$
Type II Misclassification	0.607	0.393	0.25	0.107	0.143	0.107	0.071	0.036
Type I Misclassification	0	0.061	0.091	0.136	0.136	0.121	0.167	0.288
Correctly Classified Instances	77	79	81	82	81	83	81	74
Correctly Classified %	81.9149%	84.0426%	86.1702%	87.234%	86.1702%	88.2979%	86.1702%	78.7234%
Incorrectly Classified Instances	17	15	13	12	13	11	13	20
Incorrectly Classified %	18.0851%	15.9574%	13.8298%	12.766%	13.8298%	11.7021%	13.8298%	21.2766%

The optimal cost ratio for the J48 classifier using a meta bagging learner is weight Type II errors 2.95 times that of Type I errors. From the above table we see that this is the optimal value as I tried many more but there is a general trend that the Type II errors decrease as the they received more weight and the Type I errors increased (as they received less weight). Note that between the weight of 3 and 2.95, 2.95 is preferred as both Type I and Type II errors are less (though not as $\approx$ equal as it relates to percent difference). It should be noted however that both the weight 2 and weight 3 of Type II error cost performed better when tested on the test data. The test data had the same general trend and perhaps I overfit the weighting/optimal cost ratio on the training data which is reflected in the test data.

Now as this compares to **part IV** of the previous assignment, it in fact performed worse as the optimal cost ratio had both Type I and Type II error higher. But looking at the test data for the table as a whole the bagging technique performed much better on the test data then it did in the previous assignment.

J48 with Bagging 25 Iterations

Cost Matrix	0 0.5 / 1 0	0 1 / 1 0	0 1.6 / 1 0	0 1.65 / 1 0	0 1.75 / 1 0	0 2.25 / 1 0	0 2.5 / 1 0	0 3 / 1 0	0 4 / 1 0	0 10 / 1 0
Confusion Matrix	36 19 / 9 124	42 13 / 11 122	48 7 / 15 118	48 7 / 16 117	48 7 / 18 115	47 8 / 19 114	48 7 / 20 113	48 7 / 21 112	51 4 / 26 107	51 4 / 34 99
Type II Misclassification	0.345	0.236	0.127	0.127	0.127	0.145	0.127	0.127	0.073	0.073
Type I Misclassification	0.068	0.083	0.113	0.12	0.135	0.143	0.15	0.158	0.195	0.256
Correctly Classified Instances	160	164	166	165	163	161	161	160	158	150
Correctly Classified %	85.1064%	87.234%	88.2979%	87.766%	86.7021%	85.6383%	85.6383%	85.1064%	84.0426%	79.7872%
Incorrectly Classified Instances	28	24	22	23	25	27	27	28	30	38
Incorrectly Classified %	14.8936%	12.766%	11.7021%	12.234%	13.2979%	14.3617%	14.3617%	14.8936%	15.9574%	20.2128%
Kappa Statistic	0.6199	0.6883	0.7289	0.718	0.6965	0.6723	0.6755	0.6652	0.655	0.579
Mean Absolute Error	0.1901	0.1781	0.1737	0.1777	0.1788	0.1793	0.181	0.1821	0.1873	0.2307
Root Mean Square Error	0.3312	0.3137	0.3059	0.3103	0.3124	0.3112	0.3154	0.3184	0.329	0.3861
Relative Absolute Error	45.7992%	42.9091%	41.8528%	42.8145%	43.0848%	43.1876%	43.6138%	43.8804%	45.1345%	55.5912%
Root Relative Square Error	72.7807%	68.9388%	67.2253%	68.1853%	68.6422%	68.383%	69.3046%	69.9703%	72.286%	84.8323%
TR Rate	0.655 / 0.932 / 0.851	0.764 / 0.917 / 0.872	0.873 / 0.887 / 0.883	0.873 / 0.88 / 0.878	0.873 / 0.865 / 0.867	0.855 / 0.857 / 0.856	0.873 / 0.85 / 0.856	0.873 / 0.842 / 0.851	0.927 / 0.805 / 0.84	0.927 / 0.744 / 0.798
FP Rate	0.068 / 0.345 / 0.264	0.083 / 0.236 / 0.191	0.113 / 0.127 / 0.123	0.12 / 0.127 / 0.125	0.135 / 0.127 / 0.13	0.143 / 0.145 / 0.145	0.15 / 0.127 / 0.134	0.158 / 0.127 / 0.136	0.195 / 0.073 / 0.109	0.256 / 0.073 / 0.126
Precision	0.8 / 0.867 / 0.847	0.792 / 0.904 / 0.871	0.762 / 0.944 / 0.891	0.75 / 0.944 / 0.887	0.727 / 0.943 / 0.88	0.712 / 0.934 / 0.869	0.706 / 0.942 / 0.873	0.696 / 0.941 / 0.869	0.662 / 0.964 / 0.876	0.6 / 0.961 / 0.856
Recall	0.655 / 0.932 / 0.851	0.764 / 0.917 / 0.872	0.873 / 0.887 / 0.883	0.873 / 0.88 / 0.878	0.873 / 0.865 / 0.867	0.855 / 0.857 / 0.856	0.873 / 0.85 / 0.856	0.873 / 0.842 / 0.851	0.927 / 0.805 / 0.84	0.927 / 0.744 / 0.798
TEST										
Confusion Matrix	17 11 / 5 61	21 7 / 6 60	23 5 / 8 58	23 5 / 8 58	23 5 / 8 58	23 5 / 8 58	24 4 / 9 57	24 4 / 8 58	25 3 / 11 55	27 1 / 18 48
Type II Misclassification	0.393	0.25	0.179	0.179	0.179	0.179	0.143	0.143	0.107	0.036
Type I Misclassification	0.076	0.091	0.121	0.121	0.121	0.121	0.136	0.121	0.167	0.273
Correctly Classified Instances	78	81	81	81	81	81	81	82	80	75
Correctly Classified %	82.9787%	86.1702%	86.1702%	86.1702%	86.1702%	86.1702%	86.1702%	87.234%	85.1064%	79.7872%
Incorrectly Classified Instances	16	13	13	13	13	13	13	12	14	19
Incorrectly Classified %	17.0213%	13.8298	13.8298%	13.8298%	13.8298%	13.8298%	13.8298%	12.766%	14.8936%	20.2128%

The optimal cost ratio for the J48 classifier using a meta bagging learner (25 iterations) is weight Type II errors 1.6 times that of Type I errors. From the above table, we see that this is the optimal value as I tried many more but there is a general trend that the Type II errors decrease as the they received more weight and the Type I errors increased (as they received less weight). Further the weight of 1.6. is preferred to both 1.65 and 1.75 as Type I error does not change but Type II error is lowest in cost matrix using 1.6 weight for Type II errors. It should be noted however that these three cost matrices along with the 2.25 weighted one performed identically on the test data set and in fact the weight 3 cost matrix performed best on the independent test data set but this should not influence the selection as the independent test set is for evaluation of the models only!

Now as this compares to part IV of the previous assignment, the optimal cost using the bagging technique with 25 iterations has a lower Type II error than that of **part IV** of assignment 2 however has a higher Type I error. Regardless though, the majority of J48 with bagging performed better than the optimal cost ratio in assignment 2 as expected.

Decision Stump with Bagging 10 Iterations

Cost Matrix	$\begin{array}{c\|c}0&0.4\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&1\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&1.14\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&1.15\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&1.19\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&2\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&3\\\hline 1&0\end{array}$	$\begin{array}{c\|c}0&10\\\hline 1&0\end{array}$
Confusion Matrix	$\begin{array}{c\|c}28&27\\\hline 5&128\end{array}$	$\begin{array}{c\|c}45&10\\\hline 22&111\end{array}$	$\begin{array}{c\|c}46&9\\\hline 20&113\end{array}$	$\begin{array}{c\|c}48&7\\\hline 22&111\end{array}$	$\begin{array}{c\|c}46&9\\\hline 22&111\end{array}$	$\begin{array}{c\|c}51&4\\\hline 29&104\end{array}$	$\begin{array}{c\|c}52&3\\\hline 32&101\end{array}$	$\begin{array}{c\|c}54&1\\\hline 35&98\end{array}$
Type II Misclassification	0.491	0.182	0.164	0.127	0.164	0.073	0.055	0.018
Type I Misclassification	0.038	0.165	0.15	0.165	0.165	0.218	0.241	0.263
Correctly Classified Instances	156	156	159	159	157	155	153	152
Correctly Classified %	82.9787%	82.9787%	84.5745%	84.5745%	83.5106%	82.4468%	81.383%	80.8511%
Incorrectly Classified Instances	32	32	29	29	31	33	35	36
Incorrectly Classified %	17.0213%	17.0213%	15.4255%	15.4255%	16.4894%	17.5532%	18.617%	19.1489%
Kappa Statistic	0.5341	0.6135	0.648	0.6549	0.6275	0.6258	0.6105	0.6084
Mean Absolute Error	0.2296	0.2252	0.2266	0.2272	0.2299	0.2294	0.2304	0.256
Root Mean Square Error	0.3541	0.332	0.3317	0.3325	0.3372	0.3479	0.3635	0.4114
Relative Absolute Error	55.3121%	54.2618%	54.5968%	54.7405%	55.3838%	55.2672%	55.5126%	61.6892%
Root Relative Square Error	77.7991%	72.943%	72.8925%	73.0712%	74.1044%	76.4442%	79.8847%	90.405%
TR Rate	$\begin{array}{c}0.509\\\hline 0.962\\\hline 0.83\end{array}$	$\begin{array}{c}0.818\\\hline 0.835\\\hline 0.83\end{array}$	$\begin{array}{c}0.836\\\hline 0.85\\\hline 0.846\end{array}$	$\begin{array}{c}0.873\\\hline 0.835\\\hline 0.846\end{array}$	$\begin{array}{c}0.836\\\hline 0.835\\\hline 0.835\end{array}$	$\begin{array}{c}0.927\\\hline 0.782\\\hline 0.824\end{array}$	$\begin{array}{c}0.945\\\hline 0.759\\\hline 0.814\end{array}$	$\begin{array}{c}0.982\\\hline 0.737\\\hline 0.809\end{array}$
FP Rate	$\begin{array}{c}0.038\\\hline 0.491\\\hline 0.358\end{array}$	$\begin{array}{c}0.165\\\hline 0.182\\\hline 0.177\end{array}$	$\begin{array}{c}0.15\\\hline 0.164\\\hline 0.16\end{array}$	$\begin{array}{c}0.165\\\hline 0.127\\\hline 0.138\end{array}$	$\begin{array}{c}0.165\\\hline 0.164\\\hline 0.164\end{array}$	$\begin{array}{c}0.218\\\hline 0.073\\\hline 0.115\end{array}$	$\begin{array}{c}0.241\\\hline 0.055\\\hline 0.109\end{array}$	$\begin{array}{c}0.263\\\hline 0.018\\\hline 0.09\end{array}$
Precision	$\begin{array}{c}0.848\\\hline 0.826\\\hline 0.832\end{array}$	$\begin{array}{c}0.672\\\hline 0.917\\\hline 0.845\end{array}$	$\begin{array}{c}0.697\\\hline 0.926\\\hline 0.859\end{array}$	$\begin{array}{c}0.686\\\hline 0.941\\\hline 0.866\end{array}$	$\begin{array}{c}0.676\\\hline 0.925\\\hline 0.852\end{array}$	$\begin{array}{c}0.638\\\hline 0.963\\\hline 0.868\end{array}$	$\begin{array}{c}0.619\\\hline 0.971\\\hline 0.868\end{array}$	$\begin{array}{c}0.607\\\hline 0.99\\\hline 0.878\end{array}$
Recall	$\begin{array}{c}0.509\\\hline 0.962\\\hline 0.83\end{array}$	$\begin{array}{c}0.818\\\hline 0.835\\\hline 0.83\end{array}$	$\begin{array}{c}0.836\\\hline 0.85\\\hline 0.846\end{array}$	$\begin{array}{c}0.873\\\hline 0.835\\\hline 0.846\end{array}$	$\begin{array}{c}0.836\\\hline 0.835\\\hline 0.835\end{array}$	$\begin{array}{c}0.927\\\hline 0.782\\\hline 0.824\end{array}$	$\begin{array}{c}0.945\\\hline 0.759\\\hline 0.814\end{array}$	$\begin{array}{c}0.982\\\hline 0.737\\\hline 0.809\end{array}$
TEST								
Confusion Matrix	$\begin{array}{c\|c}25&3\\\hline 8&58\end{array}$	$\begin{array}{c\|c}26&2\\\hline 13&53\end{array}$	$\begin{array}{c\|c}26&2\\\hline 14&52\end{array}$	$\begin{array}{c\|c}26&2\\\hline 14&52\end{array}$	$\begin{array}{c\|c}26&2\\\hline 14&52\end{array}$	$\begin{array}{c\|c}27&1\\\hline 17&49\end{array}$	$\begin{array}{c\|c}27&1\\\hline 17&49\end{array}$	$\begin{array}{c\|c}27&1\\\hline 18&48\end{array}$
Type II Misclassification	0.107	0.071	0.071	0.071	0.071	0.036	0.036	0.036
Type I Misclassification	0.121	0.197	0.212	0.212	0.212	0.258	0.258	0.273
Correctly Classified Instances	83	79	78	78	78	76	76	75
Correctly Classified %	88.2979%	84.0426%	82.9787%	82.9787%	82.9787%	80.8511%	80.8511%	79.7872%
Incorrectly Classified Instances	11	15	16	16	16	18	18	19
Incorrectly Classified %	11.7021%	15.9574%	17.0213%	17.0213%	17.0213%	19.1489%	19.1489%	20.2128%

Next the optimal cost ratio for the Decision stump classifier using the meta learner bagging (10 iterations) was the cost matrix with weight of Type II error equal to 1.15 . Now the weight of 1.14 had a higher Type II but lower Type I but the over all error weight was lower in the cost matrix associated to 1.15 but both performed identical on the test data set but the cost ratio of 1 (equal weights) performed best on the independent test data set. As one may expect the Decision Stump learner performed worse than J48 as it is a much cruder model that classifies based off one attribute. With that said, the Decision Stump fairly close to J48 on the test data set.

We cannot compare to **assignment 2 part IV** since we did not classify using Decision Stump in that assignment. If one we're to compare the two, Decision Stump with bagging performed arguable better on the independent test set that J48 without bagging.

Decision Stump with Bagging 25 Iterations

Cost Matrix	$\begin{matrix}0 & 0.5\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 1\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 1.05\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 1.1\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 1.15\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 1.2\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 2\\ 1 & 0\end{matrix}$	$\begin{matrix}0 & 10\\ 1 & 0\end{matrix}$
Confusion Matrix	$\begin{matrix}35 & 20\\ 11 & 122\end{matrix}$	$\begin{matrix}44 & 11\\ 20 & 113\end{matrix}$	$\begin{matrix}47 & 8\\ 21 & 112\end{matrix}$	$\begin{matrix}46 & 9\\ 20 & 113\end{matrix}$	$\begin{matrix}46 & 9\\ 22 & 111\end{matrix}$	$\begin{matrix}46 & 9\\ 23 & 110\end{matrix}$	$\begin{matrix}50 & 5\\ 28 & 105\end{matrix}$	$\begin{matrix}54 & 1\\ 35 & 98\end{matrix}$
Type II Misclassification	0.364	0.2	0.145	0.164	0.164	0.164	0.091	0.018
Type I Misclassification	0.083	0.15	0.158	0.15	0.165	0.173	0.211	0.263
Correctly Classified Instances	157	157	159	159	157	156	155	152
Correctly Classified %	83.5106%	83.5106%	84.5745%	84.5745%	83.5106%	82.9787%	82.4468%	80.8511%
Incorrectly Classified Instances	31	31	29	29	31	32	33	36
Incorrectly Classified %	16.4894%	16.4894%	15.4255%	15.4255%	16.4894%	17.0213%	17.5532%	19.1489%
Kappa Statistic	0.5816	0.6199	0.6515	0.648	0.6275	0.6174	0.6223	0.6084
Mean Absolute Error	0.2287	0.2285	0.2283	0.2275	0.2279	0.2287	0.2274	0.2624
Root Mean Square Error	0.3436	0.3326	0.3334	0.3313	0.3319	0.3345	0.3461	0.4077
Relative Absolute Error	55.1048%	55.0627%	54.9977%	54.8094%	54.9144%	55.1085%	54.7936%	63.2085%
Root Relative Square Error	75.4964%	73.0895%	73.253%	72.7926%	72.9376%	73.5055%	76.0612%	89.5913%
TR Rate	0.636 / 0.917 / 0.835	0.8 / 0.85 / 0.835	0.855 / 0.842 / 0.846	0.836 / 0.85 / 0.846	0.836 / 0.835 / 0.835	0.836 / 0.827 / 0.83	0.909 / 0.789 / 0.824	0.982 / 0.737 / 0.809
FP Rate	0.083 / 0.364 / 0.281	0.15 / 0.2 / 0.185	0.158 / 0.145 / 0.149	0.15 / 0.164 / 0.16	0.165 / 0.164 / 0.164	0.173 / 0.164 / 0.166	0.211 / 0.091 / 0.126	0.263 / 0.018 / 0.09
Precision	0.761 / 0.859 / 0.83	0.688 / 0.911 / 0.846	0.691 / 0.933 / 0.862	0.697 / 0.926 / 0.859	0.676 / 0.925 / 0.852	0.667 / 0.924 / 0.849	0.641 / 0.955 / 0.863	0.607 / 0.99 / 0.878
Recall	0.636 / 0.917 / 0.835	0.8 / 0.85 / 0.835	0.855 / 0.842 / 0.846	0.836 / 0.85 / 0.846	0.836 / 0.835 / 0.835	0.836 / 0.827 / 0.83	0.909 / 0.789 / 0.824	0.982 / 0.737 / 0.809
TEST								
Confusion Matrix	$\begin{matrix}22 & 6\\ 6 & 60\end{matrix}$	$\begin{matrix}26 & 2\\ 13 & 53\end{matrix}$	$\begin{matrix}26 & 2\\ 13 & 53\end{matrix}$	$\begin{matrix}26 & 2\\ 13 & 53\end{matrix}$	$\begin{matrix}26 & 2\\ 13 & 53\end{matrix}$	$\begin{matrix}26 & 2\\ 13 & 53\end{matrix}$	$\begin{matrix}26 & 2\\ 14 & 52\end{matrix}$	$\begin{matrix}27 & 1\\ 17 & 49\end{matrix}$
Type II Misclassification	0.214	0.071	0.071	0.071	0.071	0.071	0.071	0.036
Type I Misclassification	0.091	0.197	0.197	0.197	0.197	0.197	0.212	0.258
Correctly Classified Instances	82	79	79	79	79	79	78	76
Correctly Classified %	87.234%	84.0426%	84.0426%	84.0426%	84.0426%	84.0426%	82.9787%	80.8511%
Incorrectly Classified Instances	12	15	15	15	15	15	16	18
Incorrectly Classified %	12.766%	15.9574%	15.9574%	15.9574%	15.9574%	15.9574%	17.0213%	19.1489%

Decision Stump classifier using the meta learner bagging (25 iterations) had the best cost ratio between Type II error 1-1.15 times that. Reading from the table we want Type I error $\approx$ Type II error with Type II as low as possible. Thus we would pick the weight of 1.05. It should be noted though that they all performed identical on the Test data set. Again we have the inverse relationship that as the Type II error receives more weight it decrease and Type I error increase.

For similar reasons above, we cannot compare Decision Stump with the past assignment but again it performed about equal on the independent test set (similar to the 10 iterations). As it compares to the (10 iteration) Decision Stump with bagging, the two optimal models are about the same (though different cost weights) where the 25 iterations has an overall higher error (.149 compared to .138) but a lower Type II error (.158 vs. .165). The 25 iteration performed slightly better on the test data set however.

J48 with Boosting (AdaboostM1) 10 iterations

Cost Matrix	$\begin{array}{c\|c} 0 & 1 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 10 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 26 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 26.3 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 28 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 53 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 75 \\ \hline 1 & 0 \end{array}$
Confusion Matrix	$\begin{array}{c\|c} 39 & 16 \\ \hline 13 & 120 \end{array}$	$\begin{array}{c\|c} 48 & 7 \\ \hline 15 & 118 \end{array}$	$\begin{array}{c\|c} 48 & 7 \\ \hline 15 & 118 \end{array}$	$\begin{array}{c\|c} 50 & 5 \\ \hline 15 & 118 \end{array}$	$\begin{array}{c\|c} 49 & 6 \\ \hline 14 & 119 \end{array}$	$\begin{array}{c\|c} 48 & 7 \\ \hline 11 & 122 \end{array}$	$\begin{array}{c\|c} 51 & 4 \\ \hline 15 & 118 \end{array}$	$\begin{array}{c\|c} 48 & 7 \\ \hline 16 & 117 \end{array}$
Type II Misclassification	0.291	0.127	0.127	0.091	0.109	0.127	0.073	0.127
Type I Misclassification	0.098	0.113	0.113	0.113	0.105	0.083	0.113	0.12
Correctly Classified Instances	159	166	166	168	168	170	169	165
Correctly Classified %	84.5745%	88.2979%	88.2979%	89.3617%	89.3617%	90.4255%	89.8936%	87.766%
Incorrectly Classified Instances	29	22	22	20	20	18	19	23
Incorrectly Classified %	15.4255%	11.7021%	11.7021%	10.6383%	10.6383%	9.5745%	10.1064%	12.234%
TR Rate	$\begin{array}{c} 0.709 \\ \hline 0.902 \\ \hline 0.846 \end{array}$	$\begin{array}{c} 0.873 \\ \hline 0.887 \\ \hline 0.883 \end{array}$	$\begin{array}{c} 0.873 \\ \hline 0.887 \\ \hline 0.883 \end{array}$	$\begin{array}{c} 0.909 \\ \hline 0.887 \\ \hline 0.894 \end{array}$	$\begin{array}{c} 0.891 \\ \hline 0.895 \\ \hline 0.894 \end{array}$	$\begin{array}{c} 0.873 \\ \hline 0.917 \\ \hline 0.904 \end{array}$	$\begin{array}{c} 0.927 \\ \hline 0.887 \\ \hline 0.899 \end{array}$	$\begin{array}{c} 0.873 \\ \hline 0.88 \\ \hline 0.878 \end{array}$
FP Rate	$\begin{array}{c} 0.098 \\ \hline 0.291 \\ \hline 0.234 \end{array}$	$\begin{array}{c} 0.113 \\ \hline 0.127 \\ \hline 0.123 \end{array}$	$\begin{array}{c} 0.113 \\ \hline 0.127 \\ \hline 0.123 \end{array}$	$\begin{array}{c} 0.113 \\ \hline 0.091 \\ \hline 0.097 \end{array}$	$\begin{array}{c} 0.105 \\ \hline 0.109 \\ \hline 0.108 \end{array}$	$\begin{array}{c} 0.083 \\ \hline 0.127 \\ \hline 0.114 \end{array}$	$\begin{array}{c} 0.113 \\ \hline 0.073 \\ \hline 0.084 \end{array}$	$\begin{array}{c} 0.12 \\ \hline 0.127 \\ \hline 0.125 \end{array}$
TEST								
Confusion Matrix	$\begin{array}{c\|c} 19 & 9 \\ \hline 5 & 61 \end{array}$	$\begin{array}{c\|c} 18 & 10 \\ \hline 4 & 62 \end{array}$	$\begin{array}{c\|c} 18 & 10 \\ \hline 4 & 62 \end{array}$	$\begin{array}{c\|c} 21 & 7 \\ \hline 4 & 62 \end{array}$	$\begin{array}{c\|c} 20 & 8 \\ \hline 5 & 61 \end{array}$	$\begin{array}{c\|c} 20 & 8 \\ \hline 7 & 59 \end{array}$	$\begin{array}{c\|c} 18 & 10 \\ \hline 7 & 59 \end{array}$	$\begin{array}{c\|c} 22 & 6 \\ \hline 8 & 58 \end{array}$
Type II Misclassification	0.321	0.357	0.357	0.25	0.286	0.286	0.357	0.214
Type I Misclassification	0.076	0.061	0.061	0.061	0.076	0.106	0.106	0.121
Correctly Classified Instances	80	80	80	83	81	79	77	80
Correctly Classified %	85.1064%	85.1064%	85.1064%	88.2979%	86.1702%	84.0426%	81.9149%	85.1064%
Incorrectly Classified Instances	14	14	14	11	13	15	17	14
Incorrectly Classified %	14.8936%	14.8936%	14.8936%	11.7021%	13.8298%	15.9574%	18.0851%	14.8936%

The classifier J48 with the meta learner Booster (AdaBoostM1) was perhaps the most interest since the two types of error did not diverge as Type II error received higher and higher weight. For this reason, it made it much more difficult to optimize the cost ratio and as one observes in the table, the Type II and Type I errors get close frequently as the cost of Type II error increases from 1 to 100. It does however "optimize" at 26, 26.3, 28, and 53 where their is not exactly transitivity as some have better Type II errors but worse Type I errors or slightly worse of one error but substantially better of the other. Not surprisingly the all performed quite similarly on the Test data set with the weighting of 26 performing best (highest correct classification as well as minimal for both errors though they are not equal).

As it compares with **part IV of assignment 2**, J48 with the meta learner AdaBoostM1 performed better bother at minimizing Type II errors while keeping Type I $\approx$ Type II on the training fit data set but also performed better on the test data set. The J48 with boosting (10 iterations) was able to obtain the lowest weighted average FP rate with 0.097, 0.108, and 0.084 for the weight 26, 26.3, and 53 respectively.

J48 with Boosting(AdaboostM1) 25 iterations

Cost Matrix	$\begin{array}{cc}0 & 0.5\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 1\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 21\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 26\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 56\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 69.75\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 70\\ \hline 1 & 0\end{array}$	$\begin{array}{cc}0 & 90\\ \hline 1 & 0\end{array}$
Confusion Matrix	$\begin{array}{cc}45 & 10\\ \hline 11 & 122\end{array}$	$\begin{array}{cc}42 & 13\\ \hline 11 & 122\end{array}$	$\begin{array}{cc}46 & 9\\ \hline 17 & 116\end{array}$	$\begin{array}{cc}44 & 11\\ \hline 15 & 118\end{array}$	$\begin{array}{cc}46 & 9\\ \hline 15 & 118\end{array}$	$\begin{array}{cc}47 & 8\\ \hline 14 & 119\end{array}$	$\begin{array}{cc}47 & 8\\ \hline 15 & 118\end{array}$	$\begin{array}{cc}47 & 8\\ \hline 11 & 122\end{array}$
Type II Misclassification	0.182	0.236	0.164	0.2	0.164	0.145	0.145	0.145
Type I Misclassification	0.083	0.083	0.128	0.113	0.113	0.105	0.113	0.083
Correctly Classified Instances	167	164	162	162	164	166	165	169
Correctly Classified %	88.8298%	87.234%	86.1702%	86.1702%	87.234%	88.2979%	87.766%	89.8936%
Incorrectly Classified Instances	21	24	26	26	24	22	23	19
Incorrectly Classified %	11.1702%	12.766%	13.8298%	13.8298%	12.766%	11.7021%	12.234%	10.1064%
TR Rate	$\begin{array}{c}0.818\\ \hline 0.917\\ \hline 0.888\end{array}$	$\begin{array}{c}0.764\\ \hline 0.917\\ \hline 0.872\end{array}$	$\begin{array}{c}0.836\\ \hline 0.872\\ \hline 0.862\end{array}$	$\begin{array}{c}0.8\\ \hline 0.887\\ \hline 0.862\end{array}$	$\begin{array}{c}0.836\\ \hline 0.887\\ \hline 0.872\end{array}$	$\begin{array}{c}0.855\\ \hline 0.895\\ \hline 0.883\end{array}$	$\begin{array}{c}0.855\\ \hline 0.887\\ \hline 0.878\end{array}$	$\begin{array}{c}0.855\\ \hline 0.917\\ \hline 0.899\end{array}$
FP Rate	$\begin{array}{c}0.083\\ \hline 0.182\\ \hline 0.153\end{array}$	$\begin{array}{c}0.083\\ \hline 0.236\\ \hline 0.191\end{array}$	$\begin{array}{c}0.128\\ \hline 0.164\\ \hline 0.153\end{array}$	$\begin{array}{c}0.113\\ \hline 0.2\\ \hline 0.174\end{array}$	$\begin{array}{c}0.113\\ \hline 0.164\\ \hline 0.149\end{array}$	$\begin{array}{c}0.105\\ \hline 0.145\\ \hline 0.134\end{array}$	$\begin{array}{c}0.113\\ \hline 0.145\\ \hline 0.136\end{array}$	$\begin{array}{c}0.083\\ \hline 0.145\\ \hline 0.127\end{array}$
TEST								
Confusion Matrix	$\begin{array}{cc}19 & 9\\ \hline 5 & 61\end{array}$	$\begin{array}{cc}20 & 8\\ \hline 4 & 62\end{array}$	$\begin{array}{cc}18 & 10\\ \hline 4 & 62\end{array}$	$\begin{array}{cc}17 & 11\\ \hline 5 & 61\end{array}$	$\begin{array}{cc}19 & 9\\ \hline 5 & 61\end{array}$	$\begin{array}{cc}20 & 8\\ \hline 4 & 62\end{array}$	$\begin{array}{cc}20 & 8\\ \hline 4 & 62\end{array}$	$\begin{array}{cc}20 & 8\\ \hline 4 & 62\end{array}$
Type II Misclassification	0.321	0.286	0.357	0.393	0.321	0.286	0.286	0.286
Type I Misclassification	0.076	0.061	0.061	0.076	0.76	0.061	0.061	0.061
Correctly Classified Instances	80	82	80	78	80	82	82	82
Correctly Classified %	85.1064%	87.234%	85.1064%	82.9787%	85.1064%	87.234%	87.234%	87.234%
Incorrectly Classified Instances	14	12	14	16	14	12	12	12
Incorrectly Classified %	14.8936%	12.766%	14.8936%	17.0213%	14.8936%	12.766%	12.766%	12.766%

As mentioned above the J48 classifier with boosting, the Type I and Type II errors (as a percentage) never diverged as the weight of Type II error increased and this held true for the boosting 25 iterations as well. The best weighting was in fact when the cost of Type II error was 90 times that of Type I, since for other weights 90 had equal or better Type I error while also having the lowest Type II error. It also performed best on the Test data. Compared with the previous assignment, the J48 classifier performed much better on the test data set but for some reason J48 with boosting (25 iterations) would not let the Type I error fall below 14.5%.

Decision Stump with Boosting(AdaboostM1) 10 iterations

Cost Matrix	$\begin{array}{c\|c} 0 & 0.5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 1 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 1.25 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 1.5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 1.75 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 2 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 2.5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 8 \\ \hline 1 & 0 \end{array}$
Confusion Matrix	$\begin{array}{c\|c} 30 & 25 \\ \hline 11 & 122 \end{array}$	$\begin{array}{c\|c} 43 & 12 \\ \hline 21 & 112 \end{array}$	$\begin{array}{c\|c} 46 & 9 \\ \hline 23 & 110 \end{array}$	$\begin{array}{c\|c} 45 & 10 \\ \hline 24 & 109 \end{array}$	$\begin{array}{c\|c} 47 & 8 \\ \hline 25 & 108 \end{array}$	$\begin{array}{c\|c} 46 & 9 \\ \hline 23 & 110 \end{array}$	$\begin{array}{c\|c} 47 & 8 \\ \hline 28 & 105 \end{array}$	$\begin{array}{c\|c} 50 & 5 \\ \hline 29 & 104 \end{array}$
Type II Misclassification	0.455	0.218	0.164	0.182	0.145	0.164	0.145	0.091
Type I Misclassification	0.083	0.158	0.173	0.18	0.188	0.173	0.211	0.218
Correctly Classified Instances	152	155	156	154	155	156	152	154
Incorrectly Classified Instances	36	33	32	34	33	32	36	34
TR Rate	$\begin{array}{c} 0.545 \\ \hline 0.917 \\ \hline 0.809 \end{array}$	$\begin{array}{c} 0.782 \\ \hline 0.842 \\ \hline 0.824 \end{array}$	$\begin{array}{c} 0.836 \\ \hline 0.827 \\ \hline 0.83 \end{array}$	$\begin{array}{c} 0.818 \\ \hline 0.82 \\ \hline 0.819 \end{array}$	$\begin{array}{c} 0.855 \\ \hline 0.812 \\ \hline 0.824 \end{array}$	$\begin{array}{c} 0.836 \\ \hline 0.827 \\ \hline 0.83 \end{array}$	$\begin{array}{c} 0.855 \\ \hline 0.789 \\ \hline 0.809 \end{array}$	$\begin{array}{c} 0.909 \\ \hline 0.782 \\ \hline 0.819 \end{array}$
FP Rate	$\begin{array}{c} 0.083 \\ \hline 0.455 \\ \hline 0.346 \end{array}$	$\begin{array}{c} 0.158 \\ \hline 0.218 \\ \hline 0.201 \end{array}$	$\begin{array}{c} 0.173 \\ \hline 0.164 \\ \hline 0.166 \end{array}$	$\begin{array}{c} 0.18 \\ \hline 0.182 \\ \hline 0.181 \end{array}$	$\begin{array}{c} 0.188 \\ \hline 0.158 \\ \hline 0.158 \end{array}$	$\begin{array}{c} 0.173 \\ \hline 0.164 \\ \hline 0.166 \end{array}$	$\begin{array}{c} 0.211 \\ \hline 0.145 \\ \hline 0.164 \end{array}$	$\begin{array}{c} 0.218 \\ \hline 0.091 \\ \hline 0.128 \end{array}$
TEST								
Confusion Matrix	$\begin{array}{c\|c} 14 & 14 \\ \hline 1 & 65 \end{array}$	$\begin{array}{c\|c} 22 & 6 \\ \hline 14 & 52 \end{array}$	$\begin{array}{c\|c} 23 & 5 \\ \hline 14 & 52 \end{array}$	$\begin{array}{c\|c} 23 & 5 \\ \hline 14 & 52 \end{array}$	$\begin{array}{c\|c} 27 & 1 \\ \hline 15 & 51 \end{array}$	$\begin{array}{c\|c} 23 & 5 \\ \hline 14 & 52 \end{array}$	$\begin{array}{c\|c} 23 & 5 \\ \hline 14 & 52 \end{array}$	$\begin{array}{c\|c} 27 & 1 \\ \hline 17 & 49 \end{array}$
Type II Misclassification	0.5	0.214	0.179	0.179	0.036	0.179	0.179	0.036
Type I Misclassification	0.015	0.212	0.212	0.212	0.227	0.212	0.212	0.258
Correctly Classified Instances	79	74	75	75	78	75	75	76
Correctly Classified %	84.0426%	78.7234%	79.7872%	79.7872%	82.9787%	79.7872%	79.7872%	80.8511%
Incorrectly Classified Instances	15	20	19	19	16	19	19	18
Incorrectly Classified %	15.9574%	21.2766%	20.2128%	20.2128%	17.2013%	20.2128%	20.2128%	19.1489%

The Decision stump classifier with the meta learner AdaBoostM1 (10 iterations) obtained an optimal cost ratio when the weight of Type II error was 1.25 that of Type I error. Of the 8 model consider this was the worse both at minimizing Type II errors on the training data and for performance on the Test data set. However unlike J48 with boosting, the Decision Stump classifier did diverge as Type II error receive more and more weight.

Decision Stump with Boosting(AdaboostM1) 25 iterations

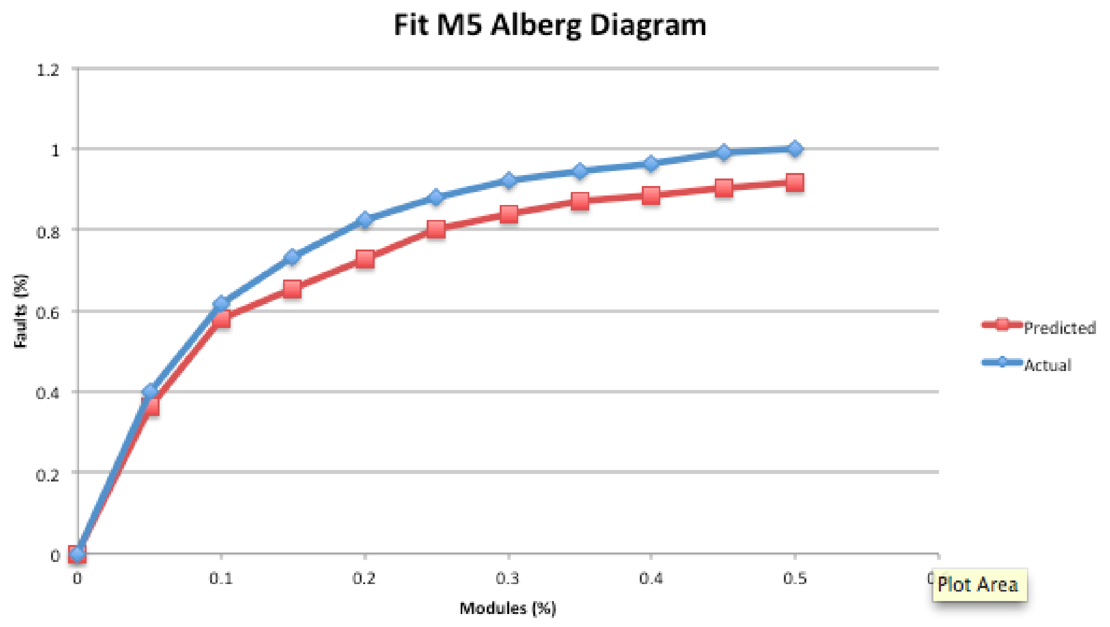
Cost Matrix	$\begin{array}{c\|c} 0 & 0.5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 1 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 2 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 2.5 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 2.6 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 3 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 3.25 \\ \hline 1 & 0 \end{array}$	$\begin{array}{c\|c} 0 & 8 \\ \hline 1 & 0 \end{array}$
Confusion Matrix	$\begin{array}{c\|c} 35 & 20 \\ \hline 8 & 125 \end{array}$	$\begin{array}{c\|c} 40 & 15 \\ \hline 15 & 118 \end{array}$	$\begin{array}{c\|c} 45 & 10 \\ \hline 19 & 114 \end{array}$	$\begin{array}{c\|c} 46 & 9 \\ \hline 21 & 112 \end{array}$	$\begin{array}{c\|c} 46 & 9 \\ \hline 22 & 111 \end{array}$	$\begin{array}{c\|c} 47 & 9 \\ \hline 24 & 109 \end{array}$	$\begin{array}{c\|c} 47 & 8 \\ \hline 22 & 111 \end{array}$	$\begin{array}{c\|c} 49 & 6 \\ \hline 27 & 106 \end{array}$
Type II Misclassification	0.364	0.273	0.182	0.164	0.164	0.145	0.145	0.109
Type I Misclassification	0.06	0.113	0.143	0.158	0.165	0.18	0.165	0.203
Correctly Classified Instances	160	158	159	158	157	156	158	155
Incorrectly Classified Instances	28	30	29	30	31	32	30	33
TR Rate	$\begin{array}{c} 0.636 \\ \hline 0.94 \\ \hline 0.851 \end{array}$	$\begin{array}{c} 0.727 \\ \hline 0.887 \\ \hline 0.84 \end{array}$	$\begin{array}{c} 0.818 \\ \hline 0.857 \\ \hline 0.846 \end{array}$	$\begin{array}{c} 0.836 \\ \hline 0.842 \\ \hline 0.84 \end{array}$	$\begin{array}{c} 0.836 \\ \hline 0.835 \\ \hline 0.836 \end{array}$	$\begin{array}{c} 0.855 \\ \hline 0.82 \\ \hline 0.83 \end{array}$	$\begin{array}{c} 0.855 \\ \hline 0.835 \\ \hline 0.84 \end{array}$	$\begin{array}{c} 0.891 \\ \hline 0.797 \\ \hline 0.824 \end{array}$
FP Rate	$\begin{array}{c} 0.06 \\ \hline 0.364 \\ \hline 0.275 \end{array}$	$\begin{array}{c} 0.113 \\ \hline 0.273 \\ \hline 0.226 \end{array}$	$\begin{array}{c} 0.143 \\ \hline 0.182 \\ \hline 0.17 \end{array}$	$\begin{array}{c} 0.158 \\ \hline 0.164 \\ \hline 0.162 \end{array}$	$\begin{array}{c} 0.165 \\ \hline 0.164 \\ \hline 0.164 \end{array}$	$\begin{array}{c} 0.18 \\ \hline 0.145 \\ \hline 0.156 \end{array}$	$\begin{array}{c} 0.165 \\ \hline 0.145 \\ \hline 0.151 \end{array}$	$\begin{array}{c} 0.203 \\ \hline 0.109 \\ \hline 0.137 \end{array}$
TEST								
Confusion Matrix	$\begin{array}{c\|c} 18 & 10 \\ \hline 4 & 62 \end{array}$	$\begin{array}{c\|c} 18 & 10 \\ \hline 5 & 61 \end{array}$	$\begin{array}{c\|c} 21 & 7 \\ \hline 10 & 56 \end{array}$	$\begin{array}{c\|c} 23 & 5 \\ \hline 13 & 53 \end{array}$	$\begin{array}{c\|c} 22 & 6 \\ \hline 10 & 56 \end{array}$	$\begin{array}{c\|c} 21 & 7 \\ \hline 9 & 57 \end{array}$	$\begin{array}{c\|c} 23 & 5 \\ \hline 11 & 55 \end{array}$	$\begin{array}{c\|c} 26 & 2 \\ \hline 16 & 50 \end{array}$
Type II Misclassification	0.357	0.357	0.25	0.179	0.214	0.25	0.179	0.071
Type I Misclassification	0.061	0.076	0.152	0.197	0.152	0.136	0.167	0.242
Correctly Classified Instances	80	79	77	76	78	78	78	76
Correctly Classified %	85.1064%	84.0426%	81.9149%	80.8511%	82.9787%	82.9787%	82.9787%	80.8511%
Incorrectly Classified Instances	14	15	17	18	16	16	16	18
Incorrectly Classified %	14.8936%	15.9574%	18.0851%	19.1489%	17.2013%	17.0213%	170213%	19.1489%

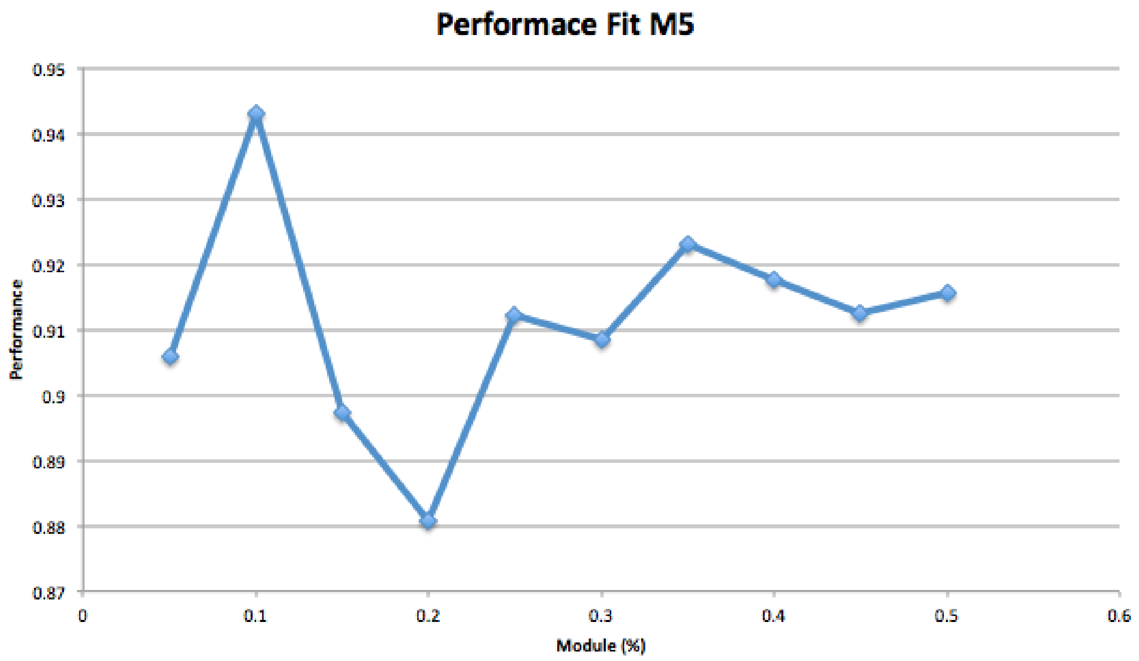
The Decision stump classifier with the meta learner AdaBoostM1 (25 iterations) obtained an optimal cost ratio when the weight of Type II error was 2.5 times that of Type I error. While Type II error is slightly higher for 3.25 vs. 2.5; 3.25 performed better on the independent Test data set. However we should take the lower Type II error **if** they are approximately equal which they are in this case. Again the Decision Stump classifier performed slightly worse but still surprisingly well when you consider how crude a model it is.

Michael Fitzgerald
April 6, 2016
Homework 4
CAP 6673
Dr. Khoshgoftaar

M5 FIT

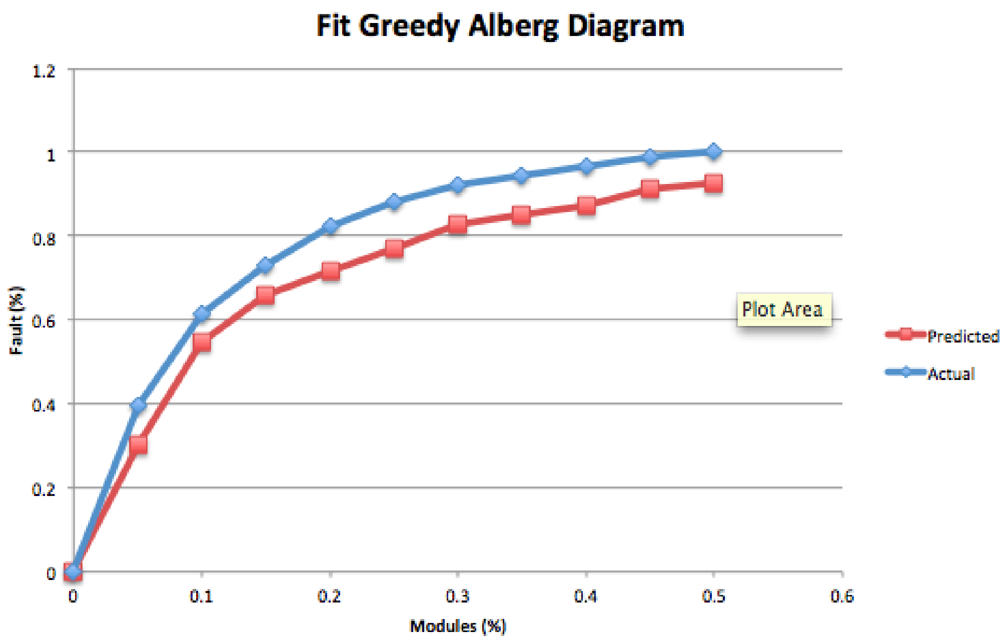
c	index	$G(c)$	$\hat{G}(c)$	$\phi(c)$	$\frac{\hat{G}(c)}{F_{tot}}$	$\frac{G(c)}{F_{tot}}$
0.50	94	427	391	0.915691	0.915691	1
0.55	103	422	385	0.912322	0.901639	0.98829
0.60	113	412	378	0.917476	0.885246	0.964871
0.65	122	403	372	0.923077	0.871194	0.943794
0.70	132	393	357	0.908397	0.836066	0.920375
0.75	141	376	343	0.912234	0.803279	0.880562
0.80	150	352	310	0.880682	0.725995	0.824356
0.85	160	312	280	0.897436	0.655738	0.730679
0.90	169	263	248	0.942966	0.580796	0.615925
0.95	179	170	154	0.905882	0.360656	0.398126

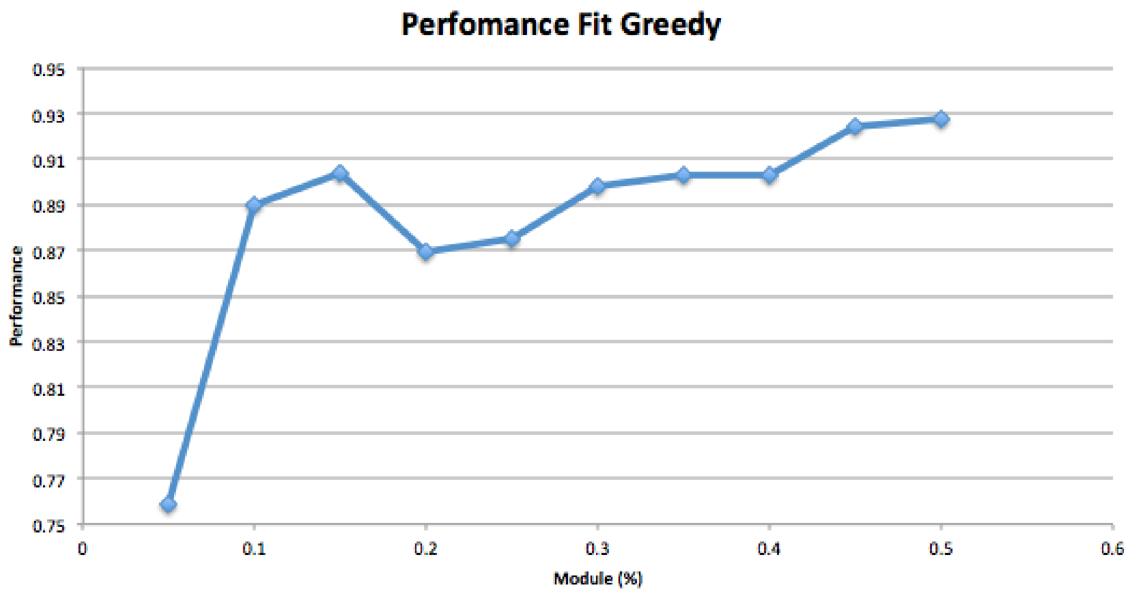




Greedy Fit

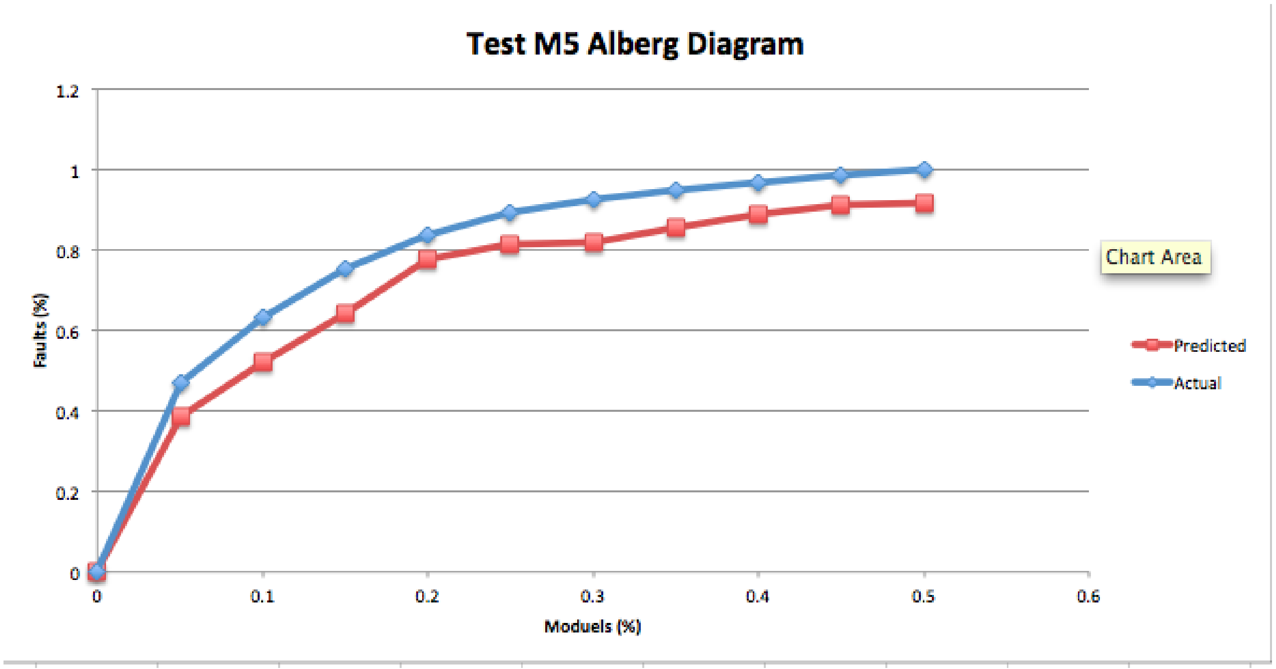
c	index	$G(c)$	$\hat{G}(c)$	$\phi(c)$	$\frac{\hat{G}(c)}{F_{tot}}$	$\frac{G(c)}{F_{tot}}$
0.50	94	427	396	0.9274	0.9274	1
0.55	103	422	390	0.924171	0.913349	0.98829
0.60	113	412	372	0.902913	0.871194	0.964871
0.65	122	403	364	0.903226	0.852459	0.943794
0.70	132	393	353	0.898219	0.826698	0.920375
0.75	141	376	329	0.875	0.770492	0.880562
0.80	150	352	306	0.869318	0.716628	0.824356
0.85	160	312	282	0.903846	0.660422	0.730679
0.90	169	263	234	0.889734	0.548009	0.615925
0.95	179	170	129	0.758824	0.302108	0.398126

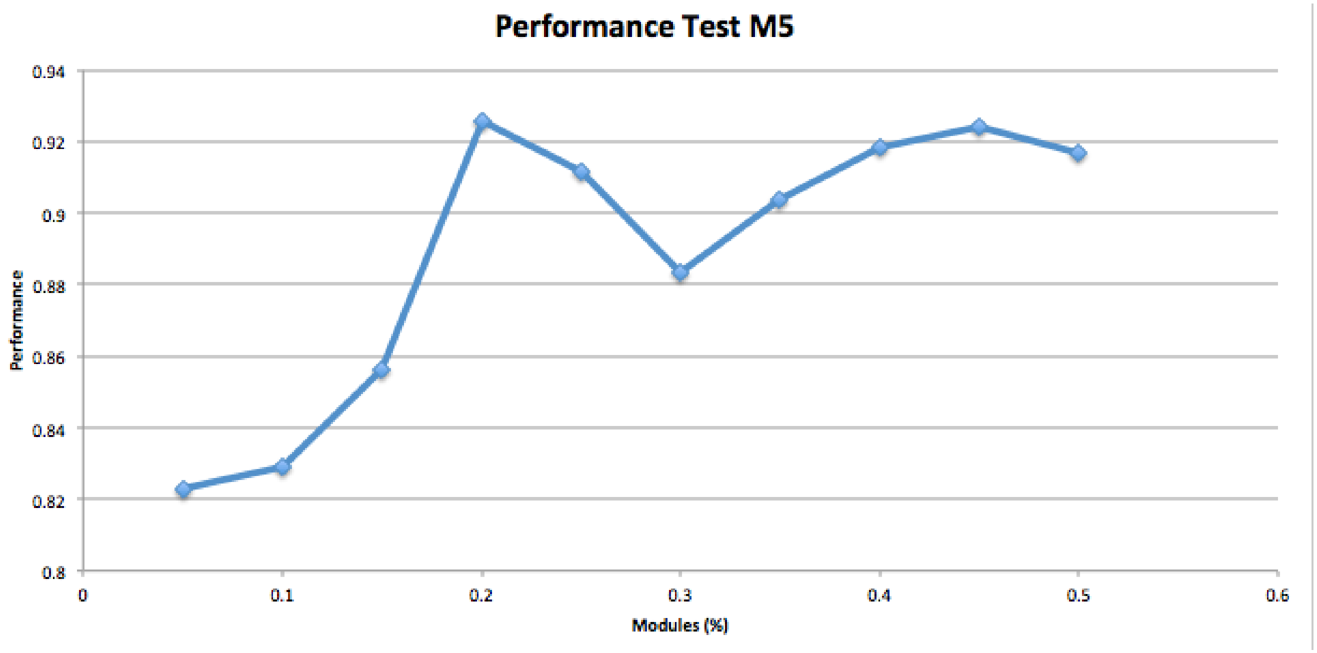




M5 Test

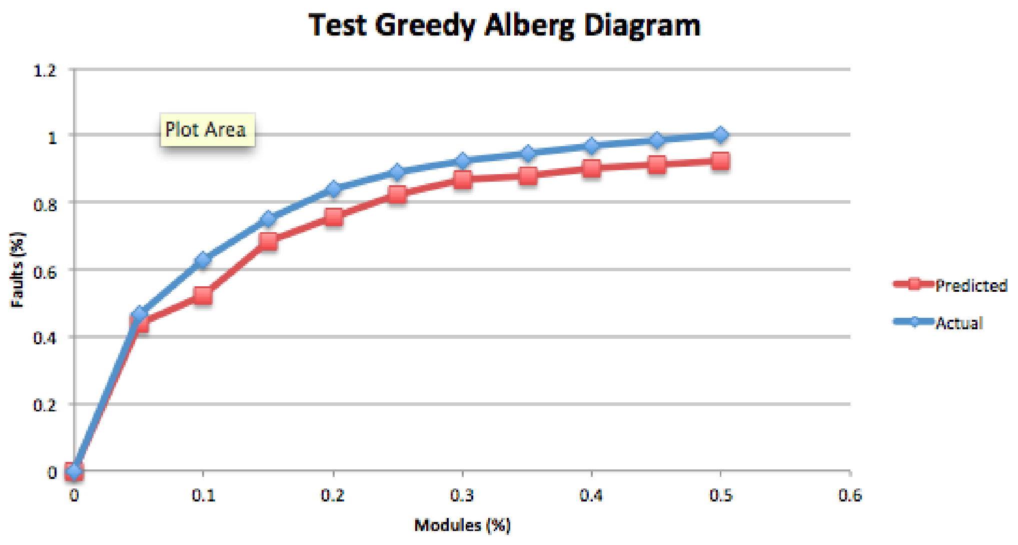
c	index	$G(c)$	$\hat{G}(c)$	$\phi(c)$	$\frac{\hat{G}(c)}{F_{tot}}$	$\frac{G(c)}{F_{tot}}$
0.50	47	241	221	0.91701245	0.91701245	1
0.55	52	237	219	0.92405063	0.90871369	0.98340249
0.60	56	233	214	0.91845494	0.8879668	0.96680498
0.65	61	228	206	0.90350877	0.85477178	0.94605809
0.70	66	223	197	0.88340807	0.81742739	0.9253112
0.75	71	215	196	0.91162791	0.81327801	0.89211618
0.80	75	202	187	0.92574257	0.77593361	0.83817427
0.85	80	181	155	0.85635359	0.64315353	0.75103734
0.90	85	152	126	0.82894737	0.52282158	0.63070539
0.95	89	113	93	0.8230085	0.38589212	0.46887967

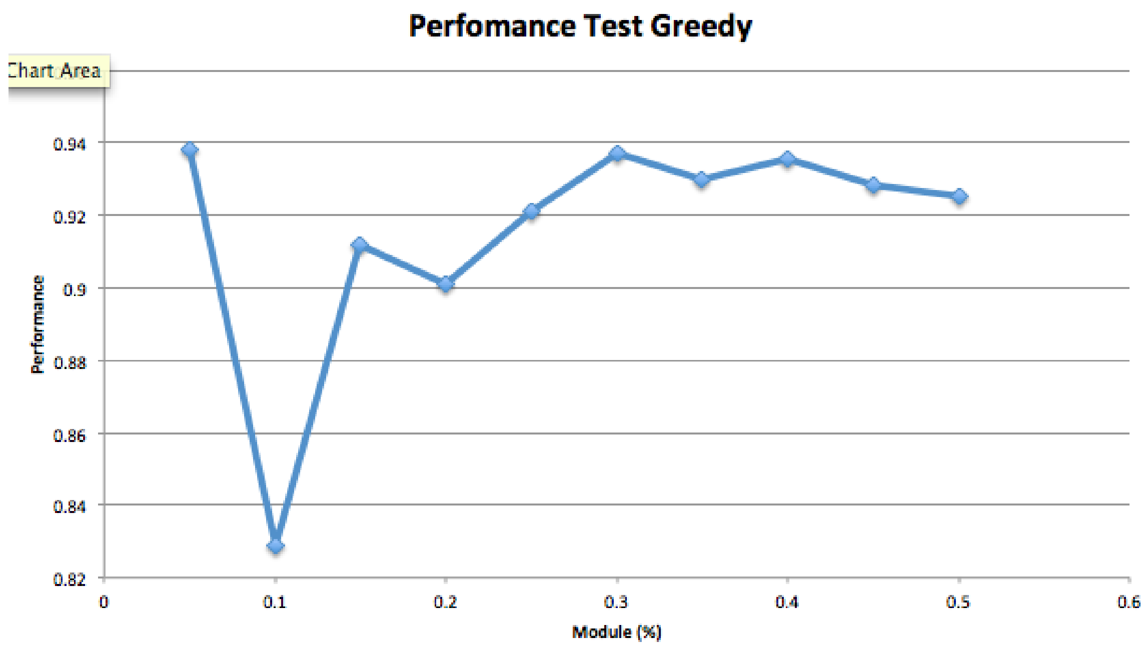




Greedy Test

c	index	$G(c)$	$\hat{G}(c)$	$\phi(c)$	$\frac{\hat{G}(c)}{F_{tot}}$	$\frac{G(c)}{F_{tot}}$
0.50	47	241	223	0.9253112	0.9253112	1
0.55	52	237	220	0.92827004	0.91286307	0.98340249
0.60	56	233	218	0.93562232	0.90456432	0.96680498
0.65	61	228	212	0.92982456	0.87966805	0.94605809
0.70	66	223	209	0.93721973	0.86721992	0.9253112
0.75	71	215	198	0.92093023	0.82157676	0.89211618
0.80	75	202	182	0.9009901	0.75518672	0.83817427
0.85	80	181	165	0.91160221	0.6846473	0.75103734
0.90	85	152	126	0.82894737	0.52282158	0.63070539
0.95	89	113	106	0.9380531	0.43983402	0.46887967





For the same reason outlined in both papers, the median faults was substantially below the mean. So for this reason in our module-ordering models (MOM), we did not analyze the model for modules below the median i.e. $c < 50$. The results for each model are above with each page representing the MOM based on the data set specified and linear regression model specified. First on each page is a table with columns c our percentile, index which specifies the index where each cut off would occur, $G(c)$ is the the number of actual faults occurring above the percentile cutoff based on the perfect ranking, $\hat{G}(c)$ is the number of actual faults occurring above the percentile cutoff based on the ordering of modules by predicted fault using the specified linear regression model, $\phi(c)$ is the performance metric and is equal to the ratio of $\hat{G}(c)$ to $G(c)$, and finally the last two columns are the ratio of $G(c)$ and $\hat{G}(c)$ to the total number of faults in the data set F_{tot}. Below the table in the first graph is Alberg Diagram where again the x-axis is the % of modules at each c and the two lines are the predicted number of faults and actual number of faults as a percentage (that is, it is the plot of the last two columns of the table, $G(c)/F_{tot}$ and $\hat{G}(c)/F_{tot}$). Finally, the second graph on each page is the performance of the model ordering model, that is just the plot of $\phi(c)$ against the % of modules on the x-axis.

The performance graphs show us how close each model came to a perfect ordering of the modules for each data set respectfully. First looking strictly at the tables and graphs of the fit data set used to build the Greedy and M-5 linear models and then were used for predicting faults; based on the performance graphs we see that the linear regression using M5 performed equal or better for all values of c with the exception of $c = 0.85, 0.55$, and 0.5. Now in all three instances M5 was still quite close at these values of c (within 0.007) and the lower thresholds of c ($c = 0.55, 0.5$) are less likely in practice as companies would have to expend significantly more resources to look at this many modules and according to Pareto's Law, one expects 20 % of the modules will contain 80 % of the faults. Thus on the fit data set, the linear regression models using M5 for attribute selection is preferred to Greedy method for the module-ordering model since it performance better for almost every value of c. Note that the performance graphs for the M5 Fit and Greedy Fit do not have the same y-scale so the variability of the M5 graph is somewhat misleading and perhaps it would have been better to plot the two graphs side by side on the same y-scale (or even just on the same graph). This was an oversight on my part.

Now on the test data set, the performance graphs of Greedy were higher than that of M5 for all values of c with the exception of $c = 0.8$. So on the test data set the linear regression model using the Greedy method of attribute selection is preferred to that of the linear regression model using the M5 model, for the module-ordering models. However the fit and test data sets came from the same data set where one-third of the data set was selected at random and called "test" and the remaining two-thirds was called "fit" (for the purpose of training and evaluating a classification model). Thus we would expected the same attribute-selector (M5 or Greedy) to be preferred in both data sets but this is not what occurred. The Fit data set is twice the size of the Test data set so it suggest that M5 is more likely the better model for module-ordering. To confirm this one could combined the data sets back into the original data set and observe which arribute-selector performs better for module ordering.

Finally, the Alberg Diagrams, support the assertion above based on the performance charts. The Alberg Diagram plot the percentage of faults in the percentage of modules (1-c) for both the actual (perfect ordering) and the predict (ordering based on model $\hat{R}$). If these two graphs are close together for a value of c that means the model did a good job on that value of c and this is reflected in the performance graphs having a high performance (y-value) for that same c (x-value). Looking at the Alberg Diagram's for the fit data set we see that the curves are relatively closer for all values of c for the M5 graph when compared to the Greedy graph. And on the test data set, we have the opposite in that the curves of the model using Greedy are closer throughout; confirming what was describe above analysis of the performance graphs. From a visual perspective, it would have been beneficial to have both graphs stacked but on the same x-scale since then the performance spikes are a given value of c would be reflected immediately above in the Alberg Diagram's becoming closer (or further) apart. This was another oversight on my part.

All computations and graphs were done in Microsoft Excel using the procedure for module-ordering models described in reference #5, and reference #9 from the course website.