CAP6673
Dr Taghi Khoshgoftaar

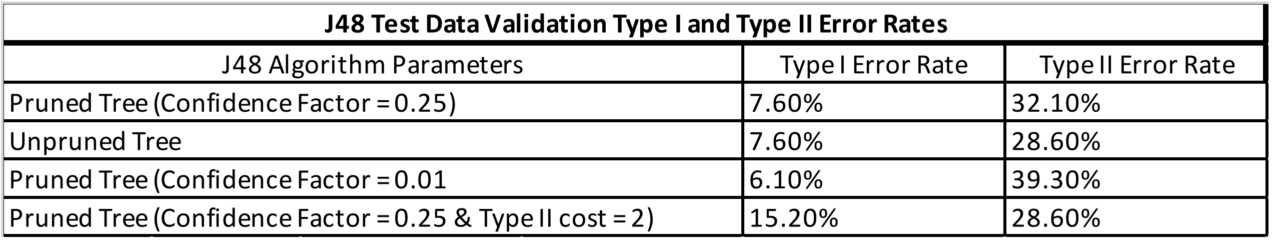
Justin Johnson
Z23136514

Assignment 3
Meta Learning Schemes with Strong vs Weak Learners

I) Introduction

A data set which describes software modules and labels each module as fault prone or non-fault prone is used to train and evaluate bagging and boosting meta learners. Bagging and boosting is applied to both the J48 decision tree (strong learner) and the Decision stump tree (weak learner). Through WEKA's cost sensitive classifier the Type I Type II error cost ratio is varied, and models are trained using the training data set (188 instances). For all examples, Type I misclassification cost is set to 1, and Type II misclassification cost is varied between 0.5 and 10. The optimal cost ratio is identified by selecting the model whose cross validation resulted in a Type I and Type II error rate that are approximately the same with a Type II error rate as low as possible. Finally, all models are evaluated against unfamiliar test data (94 instances). Classification results are also compared to the Pruned Tree model from the previous assignment.

Results from previous assignment are below, our new bagging and boosting results will be compared with row 4.



J48 Test Data Validation Type I and Type II Error Rates

J48 Algorithm Parameters	Type I Error Rate	Type II Error Rate
Pruned Tree (Confidence Factor = 0.25)	7.60%	32.10%
Unpruned Tree	7.60%	28.60%
Pruned Tree (Confidence Factor = 0.01	6.10%	39.30%
Pruned Tree (Confidence Factor = 0.25 & Type II cost = 2)	15.20%	28.60%

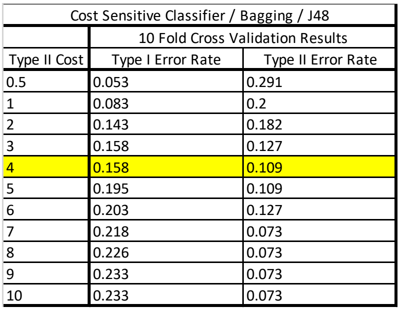
Both bagging and boosting algorithms strive to increase classification accuracy by creating multiple models and combining their results to produce predictions on new data. For this assignment, bagging and boosting is first completed using a default of 10 iterations/models (Parts II - V), and then all steps are repeated using 25 iterations/models (Parts VI - IX).

The bagging ensemble algorithm constructs multiple training sets from the original training data set by randomly sampling the set with re-substitution. It then trains all of the models and produces a final classification result by combining the results of all individual models. In bagging algorithms, each model is given an equal weight, and the models vote on the final classification result. The label that receives the greatest number of votes is selected as the new instances class.

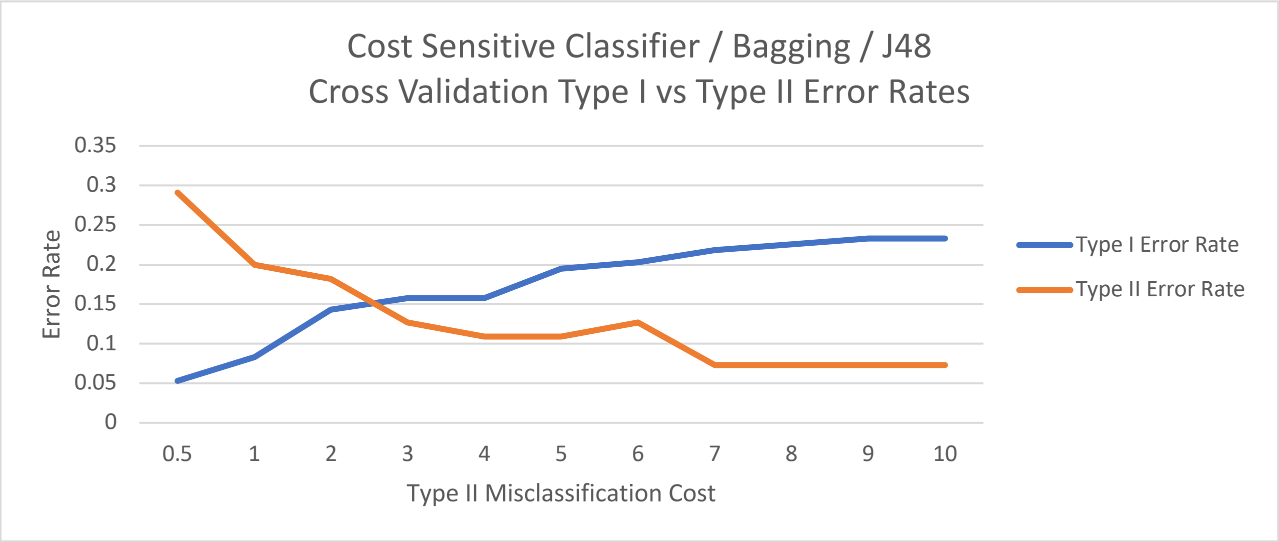
Boosting algorithms are similar to bagging as they too construct multiple models and vote on new instance classification. Unlike bagging however, boosting algorithms construct models iteratively, using previous models to influence future models. The AdaBoost boosting algorithm applies weights to each instance in an iterative manner, using the new weights to construct new models which will complement previous models. Instances that are misclassified are given a greater weight, forcing the next iteration's model to focus more on instances previously misclassified.

II) Cost Sensitive Classifier Combined with Bagging and J48 Decision Tree (10 iterations)

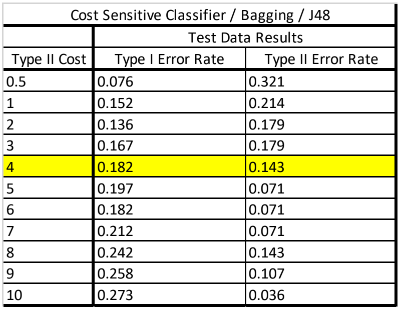
10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10. Bagging algorithm default of 10 iterations is used.



Cost Sensitive Classifier / Bagging / J48		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.053	0.291
1	0.083	0.2
2	0.143	0.182
3	0.158	0.127
4	0.158	0.109
5	0.195	0.109
6	0.203	0.127
7	0.218	0.073
8	0.226	0.073
9	0.233	0.073
10	0.233	0.073



A Type II misclassification cost of 4.0 is identified as the optimal cost ratio, as Type I and Type II error rates are roughly the same and Type II error rate is lowest. Next, the models are evaluated against the test data set and results are analyzed.



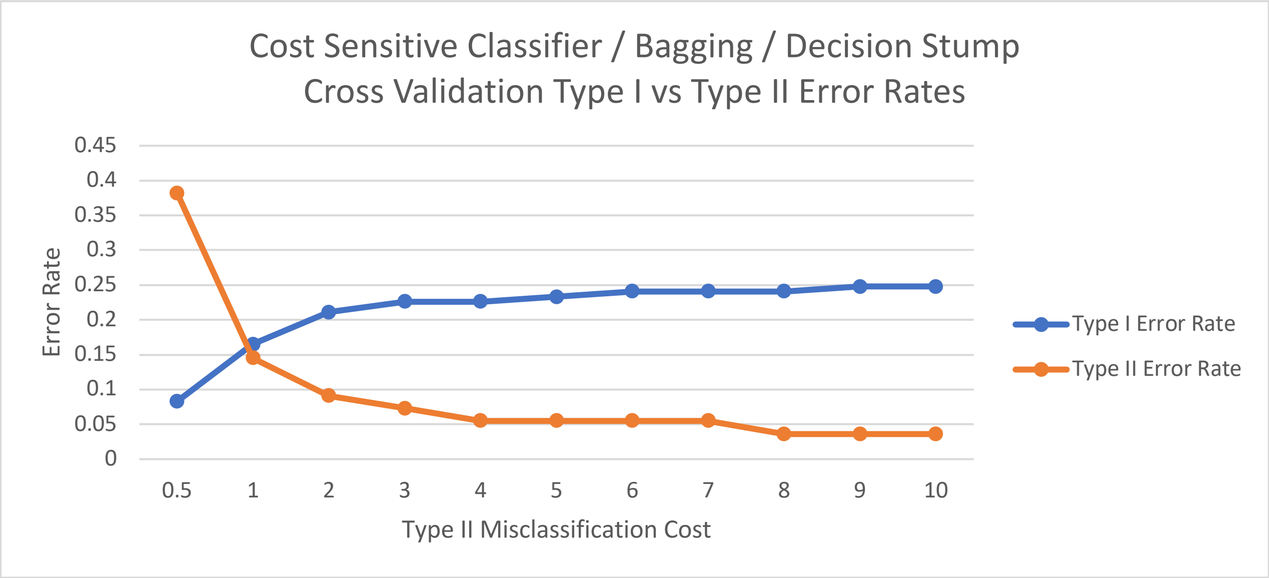
Cost Sensitive Classifier / Bagging / J48		
	Test Data Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.076	0.321
1	0.152	0.214
2	0.136	0.179
3	0.167	0.179
4	0.182	0.143
5	0.197	0.071
6	0.182	0.071
7	0.212	0.071
8	0.242	0.143
9	0.258	0.107
10	0.273	0.036

Highlighted is the model that was selected during cross validation, with Type II error cost of 4.0. This model achieved a Type I error rate of 18.2% and Type II error rate of 14.3% when evaluated with test data. The Type II error rate is approximately 7% less than it would be if a cost ratio of 1 was selected, and Type I error rate only increased by 3%, showing that the chosen cost ratio was effective in improving results. Comparing to the previous assignment, this model performed very well, achieving less Type II misclassifications, reducing from 28.6% in previous assignment down to 14.3%. There was a slight increase in Type I error rate from 15.2% to 18.2%, but this tradeoff is expected, and the increase is insignificant as we are primarily concerned with Type II error rate. Recall that Type II misclassifications, classifying a fault prone module as non-fault prone, are more expensive and should be minimized.

III) Cost Sensitive Classifier Combined with Bagging and Decision Stump Tree (10 iterations)

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10.

Cost Sensitive Classifier / Bagging / Decision Stump		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.083	0.382
1	0.165	0.145
2	0.211	0.091
3	0.226	0.073
4	0.226	0.055
5	0.233	0.055
6	0.241	0.055
7	0.241	0.055
8	0.241	0.036
9	0.248	0.036
10	0.248	0.036



A Type II misclassification cost of 1.0 is identified as the optimal cost ratio, as Type I and Type II error rates are roughly the same. A Type II cost of 2.0 is appealing, as the model only had a 9.1% Type II misclassification rate, however the Type I error rate increased and the required balance between Type I and Type II error rates is not met. Next, the models are evaluated against the test data.

Cost Sensitive Classifier / Bagging / Decision Stump		
	Test Data Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.03	0.536
1	0.182	0.107
2	0.212	0.071
3	0.242	0.036
4	0.258	0.036
5	0.273	0.036
6	0.273	0.036
7	0.258	0.036
8	0.273	0.036
9	0.288	0.036
10	0.288	0.036

Highlighted is the model that was selected during cross validation, with Type II error cost of 1.0. Against the test data, this model scored a Type I error rate of 18.2% and Type II error rate of 10.7%. As depicted in the table, other cost ratios produced lower Type II error rates, but at the expense of increased Type I error rates.

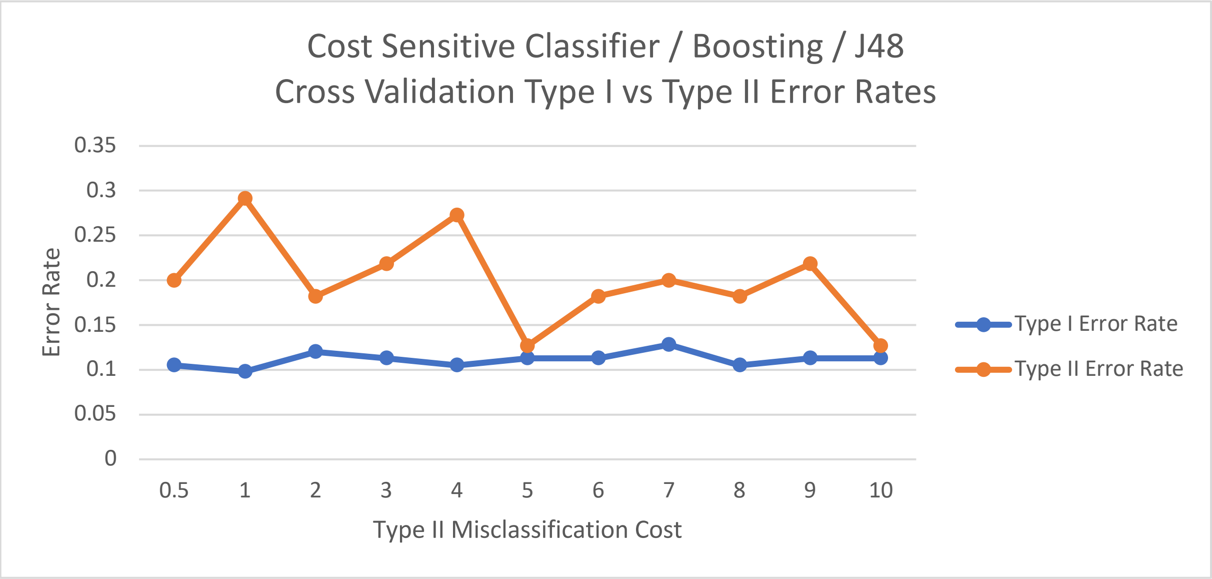
Compared to the J48 model from previous assignment, this model performed very well. The Type II error rate decreased from 28.6% down to 10.7%, and the Type I error rate increased from 15.2% to 18.2%. This model also outperformed the model from Part II, as the Type II misclassification rate reduced from 14.3% down to 10.7%.

The bagging meta learner combined with decision stump tree (weak learner) performed slightly better than the bagging meta learner combined with J48 (strong learner), with a 3.6% decrease in total Type II misclassifications rate, given the selected cost ratios. For most of the models evaluated during cross validation and test data in Part II and Part III, the decision stump tree tends to perform better and produce less Type II misclassification when combined with meta learning methods.

IV) Cost Sensitive Classifier with Boosting (AdaBoostM1) and J48 Decision Tree (10 iterations)

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10.

Cost Sensitive Classifier / Boosting / J48		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.105	0.2
1	0.098	0.291
2	0.12	0.182
3	0.113	0.218
4	0.105	0.273
5	0.113	0.127
6	0.113	0.182
7	0.128	0.2
8	0.105	0.182
9	0.113	0.218
10	0.113	0.127



A Type II misclassification cost of 5.0 is identified as the optimal cost ratio, as Type I and Type II error rates are roughly the same and Type II error is as low as possible. Note the trend of the Type II error rate. The Type II error rate does not decrease as expected, there is a lot of variance in Type II error rate, and the Type I error rate does not increase by more than 1 or 2 %.

Next, the 11 models are evaluated against the test data.

Cost Sensitive Classifier / Boosting / J48		
	Test Data Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.061	0.214
1	0.076	0.321
2	0.061	0.429
3	0.061	0.357
4	0.076	0.393
5	0.061	0.357
6	0.045	0.393
7	0.076	0.321
8	0.061	0.393
9	0.045	0.357
10	0.061	0.357

Highlighted is the model that was selected during cross validation, with Type II error cost of 5.0. Against the test data, this model scored a Type I error rate of 6.1% and Type II error rate of 35.7%. As seen in the above table, and similar to the cross-validation results, there is no apparent trend in the Type I and Type II error rates as the model cost ratios vary from 0.5 to 10. The model whose Type II error cost is 0.5 actually scored the best results in this set.

In comparison with the boosting meta learners from parts II and III, this model incorrectly classified twice as many fault prone models as non-fault prone, increasing Type II error rate from 10-14% to 35.7%.

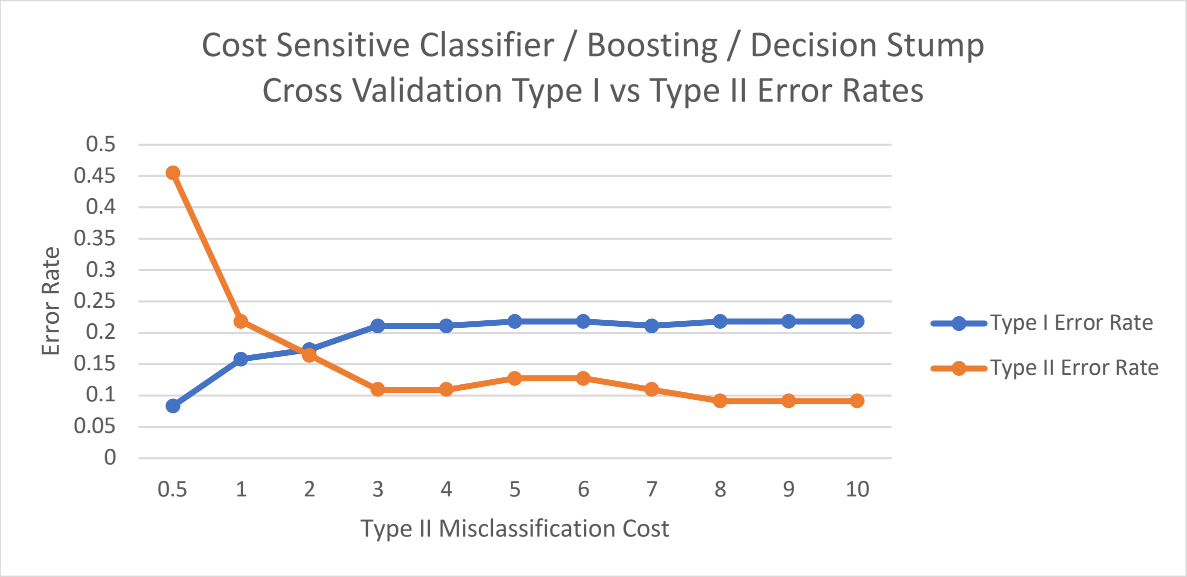
This model also performed much worse than the classification model from the previous assignment. The Type II error rate increased from 28.6% down o 35.7%. Type I error rate is less than the previous assignment's model, but the Type II error rate is too high.

The boosting algorithm attempts to construct strong learners from weak learners, by creating many models that complement each other. Since the J48 decision tree is a strong learner, it is possible that the boosting and J48 combination is creating over-fit models which do not generalize well to new data. In general, it is suggested that boosting algorithms may see degradation in performance when combined with a strong learner.

V) Cost Sensitive Classifier with Boosting (AdaBoostM1) and Decision Stump Tree (10 iterations)

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10.

Cost Sensitive Classifier / Boosting / Decision Stump		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.083	0.455
1	0.158	0.218
2	0.173	0.164
3	0.211	0.109
4	0.211	0.109
5	0.218	0.127
6	0.218	0.127
7	0.211	0.109
8	0.218	0.091
9	0.218	0.091
10	0.218	0.091



Continuing with the previous strategy for selecting an optimal cost ratio, A Type II misclassification cost of 2.0 is chosen, as Type I and Type II error rates are approximately the same with Type II error low. Compared to the 3 other training models in this project, this model with Type II cost of 2.0 returned the highest misclassification rates during cross validation, with a Type II error rate of 16.4% and a Type I error rate of 17.3%. Next, all 11 models are evaluated using unseen test data.

Cost Sensitive Classifier / Boosting / Decision Stump		
	Test Data Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.015	0.5
1	0.212	0.214
2	0.212	0.179
3	0.227	0.071
4	0.242	0.071
5	0.242	0.036
6	0.242	0.036
7	0.258	0.036
8	0.258	0.036
9	0.258	0.036
10	0.258	0.036

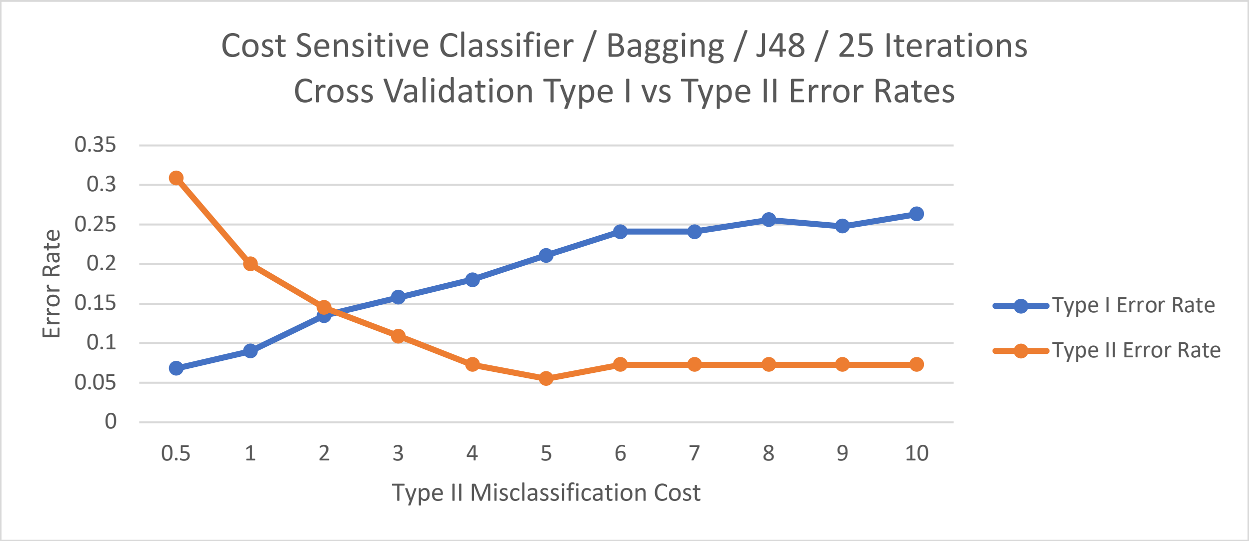
The selected line in above chart indicates the model that was chosen during cross validation, with a Type II cost of 2.0 and Type I cost of 1.0. Against the test data set, the model resulted in Type I error rate of 21.2% and Type II error rate of 17.9%. The boosting meta-learner combined with the decision stump (weak learner) performed better than the boosting meta-learner combined with the J48 (strong learner). Both instances of the bagging models from parts II and III performed better than both of the boosting models from sections IV and V.

As noted in the above table, the boosting learner achieved very low Type II misclassification rates once the Type II cost increased to 3.0, reducing to 7.1% or less. The method used to select cost ratios for this assignment is just one of many methods, and there are methods that will potentially lead to lower Type II error rates. Regardless of the method, it remains true that decreases in Type II error rate is a tradeoff for increases to Type I error rates.

VI) Cost Sensitive Classifier Combined with Bagging (25 iterations) and J48 Decision Tree

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10. Bagging algorithm iterations is set to 25.

Cost Sensitive Classifier / Bagging / J48 / 25 iterations		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.068	0.309
1	0.09	0.2
2	0.135	0.145
3	0.158	0.109
4	0.18	0.073
5	0.211	0.055
6	0.241	0.073
7	0.241	0.073
8	0.256	0.073
9	0.248	0.073
10	0.263	0.073



A Type II misclassification cost of 3.0 is identified as the optimal cost ratio, as Type I and Type II error rates are roughly the same and Type II error rate is lowest. Next, the models are evaluated against the test data set and results are analyzed.

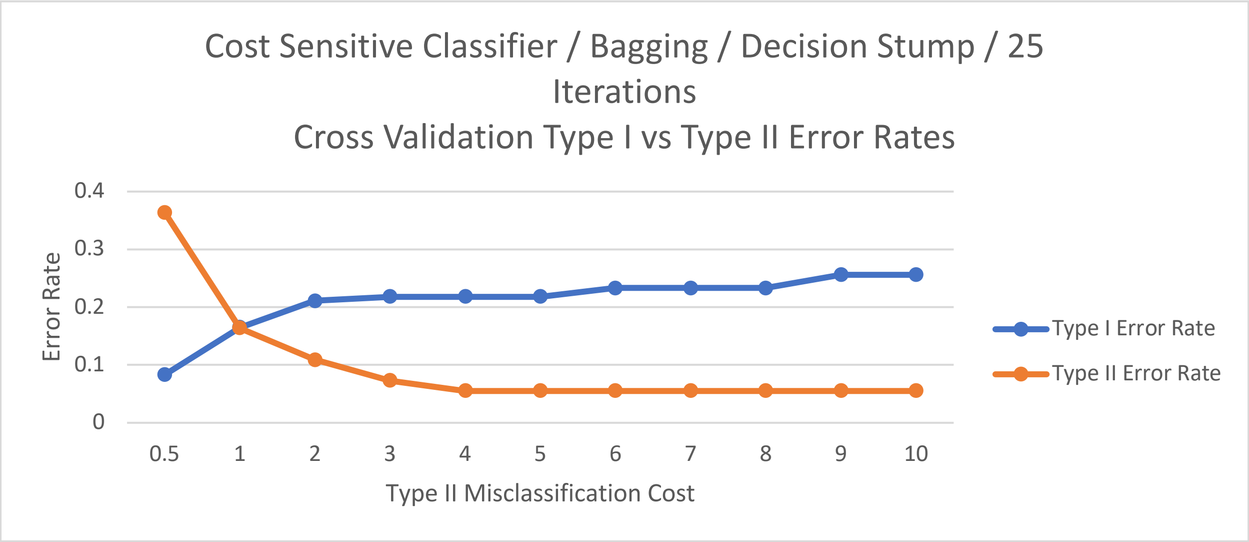
Test Data Results	
Type I Error Rate	Type II Error Rate
0.076	0.357
0.076	0.321
0.152	0.214
0.167	0.143
0.182	0.107
0.167	0.071
0.212	0.107
0.182	0.107
0.242	0.071
0.242	0.107
0.258	0.071

The above chart reveals the model’s evaluation against unseen test data. The highlighted line reflects the model selected during cross-validation. A type II error rate of 14.3% was achieved. When compared to the same algorithm using 10 iterations (Part II), we achieved an equal Type II misclassification and a slight decrease in Type I misclassifications. The added iterations led to an overall improved classification accuracy.

VII) Cost Sensitive Classifier Combined with Bagging (25 iterations) and Decision Stump Tree

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10. Bagging algorithm iterations is set to 25.

Cost Sensitive / Bagging / Decision Stump / 25 Iters		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.083	0.364
1	0.165	0.164
2	0.211	0.109
3	0.218	0.073
4	0.218	0.055
5	0.218	0.055
6	0.233	0.055
7	0.233	0.055
8	0.233	0.055
9	0.256	0.055
10	0.256	0.055



A Type II misclassification cost of 1.0 is identified as the optimal cost ratio, as Type I and Type II error rates are roughly the same and Type II error rate is lowest. One could argue a Type II cost of 2.0 as there is significant decrease in Type II error, however the increase to Type I error is to great. Next, the models are evaluated against the test data set and results are analyzed.

Test Data Results	
Type I Error Rate	Type II Error Rate
0.091	0.25
0.197	0.071
0.242	0.036
0.242	0.036
0.258	0.036
0.258	0.036
0.258	0.036
0.258	0.036
0.273	0.036
0.273	0.036
0.273	0.036

The above chart reveals the model's evaluation against unseen test data. The highlighted line reflects the model selected during cross-validation. A type II error rate of 7.1% was achieved, at the cost of a

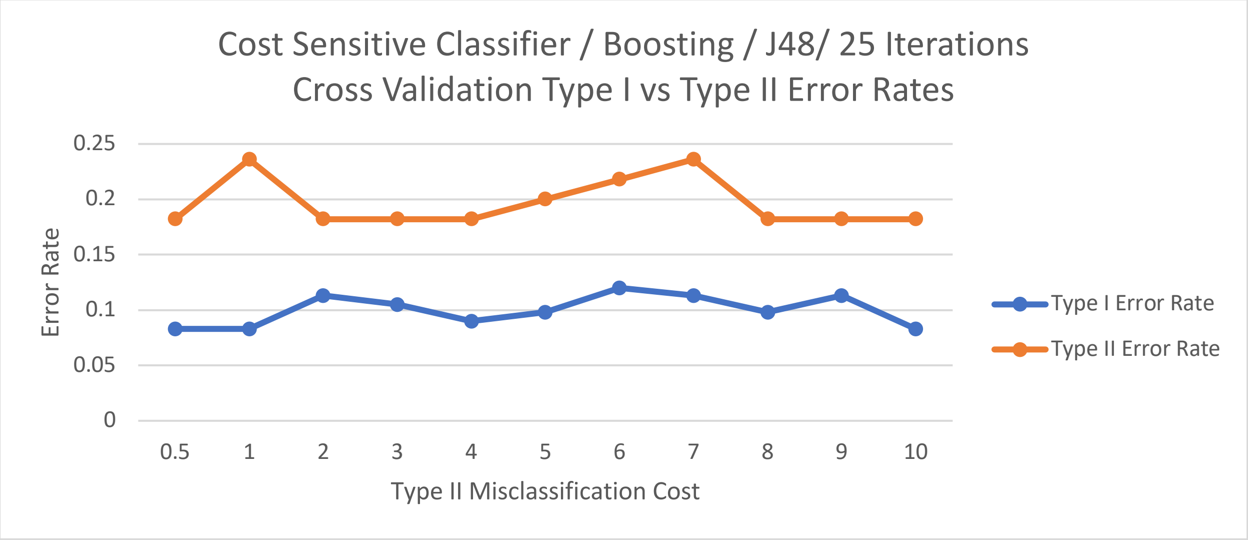
Type I error rate equal to 19.7%. Compared to the same algorithm in Part III, with 10 iterations, we saw a reduction in Type II error rate from 10.7% down to 7.1%, and an increase in Type I error rate from 18.2% up to 19.7%. This increase in Type I error is minimal and is justified by the decrease in Type II error rate. Therefore, the bagging learning with Decision Stump (weak learner) performed better when the number of iterations was increased from 10 to 25.

At this time, this is the best model produced by any of the ensemble methods previously evaluated in this assignment. When compared to the previous assignment's model (J48 / Confidence = 0.25 / CII = 2.0), we have seen a drastic improvement in model performance. Our previous assignment's model produced a Type I error rate of 15.2% and Type II error rate of 28.6%. With just a minimal increase in Type I error rate, we have seen a major decrease in Type II error rate from 28.6% down to 7.1%, significantly fewer Type II misclassifications.

VIII) Cost Sensitive Classifier Combined with Boosting (25 iterations) and J48 Decision Tree

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10. Boosting algorithm iterations is set to 25.

Cost Sensitive Classifier / Boosting / J48 / 25 Iters		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.083	0.182
1	0.083	0.236
2	0.113	0.182
3	0.105	0.182
4	0.09	0.182
5	0.098	0.2
6	0.12	0.218
7	0.113	0.236
8	0.098	0.182
9	0.113	0.182
10	0.083	0.182



In this example, model selection is not so trivial. Type II costs of 2, 3, and 4 all yielded equal Type II error rates. I've selected a Type II cost of 2.0 as the ideal model as Type I and Type II error rates are most similar in this model, but one could also argue that Type II costs of 3 or 4. Due to the unusual trend in Type I and Type II errors, it is predicted that Boosting in combination with J48 will not perform well against new data because of the boosting algorithm combined with a strong learner.

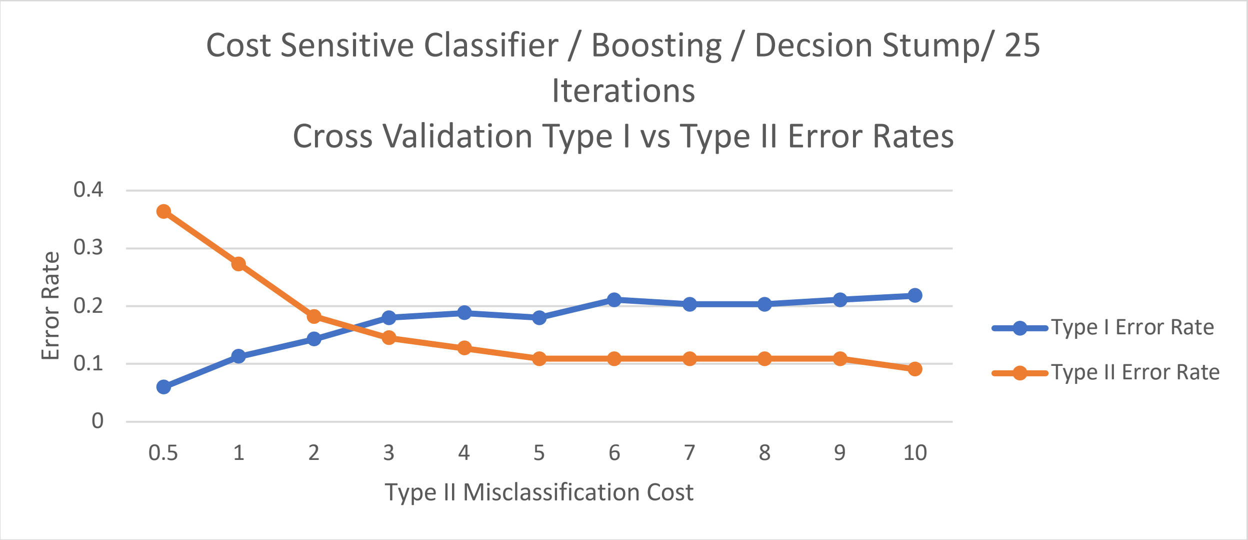
Test Data Results	
Type I Error Rate	Type II Error Rate
0.076	0.321
0.061	0.286
0.045	0.357
0.076	0.286
0.061	0.321
0.061	0.393
0.061	0.321
0.061	0.357
0.076	0.357
0.061	0.286
0.061	0.214

The above chart displays model evaluation against test data. The selected model resulted in Type II error rate of 35.7% and Type I error rate of 4.5%. When compared to Part IV, we achieved the same Type II error rate with a slight reduction in Type I error. The Type II error rate is very high, and in the case of both 10 iterations and 25 iterations it is clear that boosting combined with the strong learner J48 decision tree did not perform well. As suggested earlier, the boosting methodology is known to see degradation when combined with a strong learner.

IX) Cost Sensitive Classifier Combined with Boosting (25 iterations) and Decision Stump Tree

10-fold cross validation is applied to the fit data set and the cost ratio is varied by adjusting the cost of Type II misclassification error between 0.5 and 10. Boosting algorithm iterations is set to 25.

Cost Sensitive / Boosting / Decision Stump / 25 Iters		
	10 Fold Cross Validation Results	
Type II Cost	Type I Error Rate	Type II Error Rate
0.5	0.06	0.364
1	0.113	0.273
2	0.143	0.182
3	0.18	0.145
4	0.188	0.127
5	0.18	0.109
6	0.211	0.109
7	0.203	0.109
8	0.203	0.109
9	0.211	0.109
10	0.218	0.091



A Type II cost of 5.0 was selected. The first cost considered was 3.0, as Type I and Type II error rates are similar with Type II relatively low. After reviewing other options, it was clear that a cost of 5.0 resulted in the same Type I error rate and a Type II error rate 4% lower. As minimizing Type II errors is important, the Type II cost of 5.0 was selected.

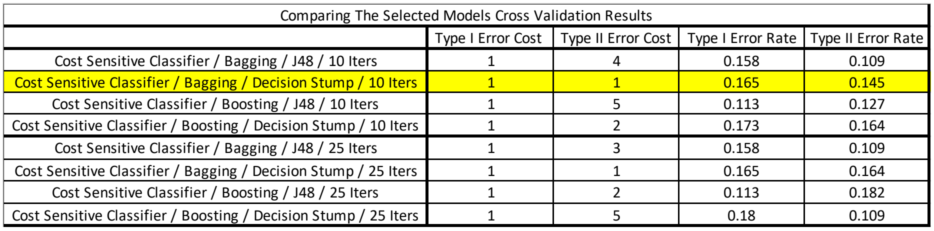
Test Data Results	
Type I Error Rate	Type II Error Rate
0.061	0.357
0.076	0.357
0.152	0.25
0.136	0.25
0.182	0.143
0.212	0.107
0.242	0.071
0.242	0.071
0.242	0.071
0.242	0.036
0.242	0.036

The above table displays results for all models, with the highlighted line reflecting the model selected during cross-validation. Once again, the boosting algorithm performed better when combined with a weak learner, with a Type I error rate of 21.2% and Type II error rate of 10.7%. This algorithm previously scored a Type I error rate of 21.2% and 17.9%, using 10 iterations in Part V. With the added iterations, the model performed better against test data, decreasing Type II error rate from 17.9% down to 10.7%.

X) Conclusions

Bagging and boosting meta learners are used in combination with both weak (decision stump) and strong (J48) decision tree classifiers. Models are first evaluated using a default of 10 iterations, and then again using 25 iterations. WEKA's cost sensitive classifier was used to vary the cost ratio and identify the best cost ratio. For each selected cost ratio, the models are evaluated against unseen test data so that their results can be compared.

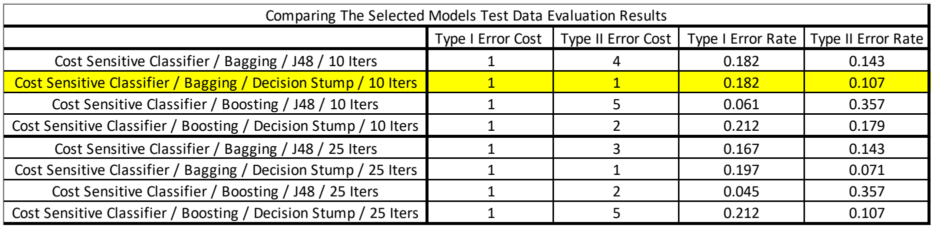
We compare the cross-validation results for all selected models below. At first glance the boosting / J48 with 10 iterations looks promising, with low Type II error rate and approximately equal Type I Type II error rate. However, the plot produced as cost ratio varies is not promising, with a lot of variance and no apparent trend. With that intuition and the below results, the Bagging / Decision Stump / 10 iterations would be selected during cross validation as the best model. It's Type I and Type II error rates are approximately equal at 16.5% and 14.5% respectively, and the Type II error rate is low in comparison to all other models. The same model trained with 25 iterations also performed very well during cross validation, with Type I error rate of 16.5% and Type II error rate of 16.4%. The 10-iteration model was selected because of the slight reduction in Type II error rate. As mentioned previously, this method for model selection is just one approach, other methods may be explored depending on the current use case. Most importantly, model selection must be made during cross-validation evaluation. The test results cannot be used to select the best model.



Comparing The Selected Models Cross Validation Results

	Type I Error Cost	Type II Error Cost	Type I Error Rate	Type II Error Rate
Cost Sensitive Classifier / Bagging / J48 / 10 Iters	1	4	0.158	0.109
Cost Sensitive Classifier / Bagging / Decision Stump / 10 Iters	1	1	0.165	0.145
Cost Sensitive Classifier / Boosting / J48 / 10 Iters	1	5	0.113	0.127
Cost Sensitive Classifier / Boosting / Decision Stump / 10 Iters	1	2	0.173	0.164
Cost Sensitive Classifier / Bagging / J48 / 25 Iters	1	3	0.158	0.109
Cost Sensitive Classifier / Bagging / Decision Stump / 25 Iters	1	1	0.165	0.164
Cost Sensitive Classifier / Boosting / J48 / 25 Iters	1	2	0.113	0.182
Cost Sensitive Classifier / Boosting / Decision Stump / 25 Iters	1	5	0.18	0.109

Below are the test results for all 8 models. The model selected achieved a Type I error rate of 18.2% and Type II error rate of 10.7%. The approach applied for model selection was effective, as this model performed well against the test data in comparison to other models. The Bagging / Decision Stump / 25 iterations model performed the best, but it is impossible to know this during the model selection phase of development. The boosting algorithm combined with the J48 decision tree performed the worst, with Type II misclassification rate of 35.7%. It has been suggested that the boosting algorithm may see degradation when combined with a strong learner. This hypothesis is consistent, as the boosting algorithm applied to the decision stump (weak learner) performed significantly better and is comparable to the bagging models.



Comparing The Selected Models Test Data Evaluation Results

	Type I Error Cost	Type II Error Cost	Type I Error Rate	Type II Error Rate
Cost Sensitive Classifier / Bagging / J48 / 10 Iters	1	4	0.182	0.143
Cost Sensitive Classifier / Bagging / Decision Stump / 10 Iters	1	1	0.182	0.107
Cost Sensitive Classifier / Boosting / J48 / 10 Iters	1	5	0.061	0.357
Cost Sensitive Classifier / Boosting / Decision Stump / 10 Iters	1	2	0.212	0.179
Cost Sensitive Classifier / Bagging / J48 / 25 Iters	1	3	0.167	0.143
Cost Sensitive Classifier / Bagging / Decision Stump / 25 Iters	1	1	0.197	0.071
Cost Sensitive Classifier / Boosting / J48 / 25 Iters	1	2	0.045	0.357
Cost Sensitive Classifier / Boosting / Decision Stump / 25 Iters	1	5	0.212	0.107

Finally, these results are compared to the J48 pruned decision tree (confidence factor of 0.25 and Type II cost of 2.0) from the previous assignment. The model selected performed significantly better, reducing Type II error rate from 28.6% down to 10.7% against test data, with only a small increase in Type I error rate from 15.2% to 18.2%.