Joseph Cantrell

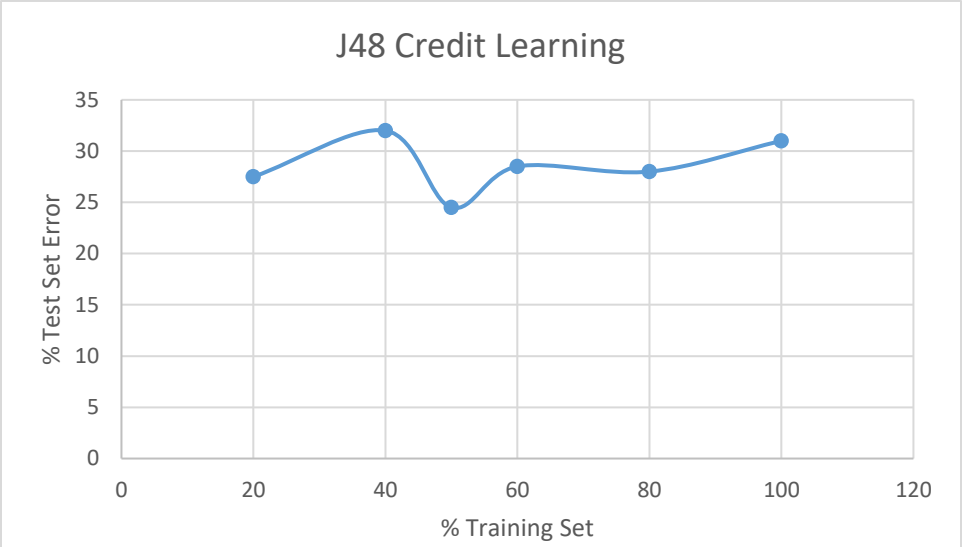
CS 4641

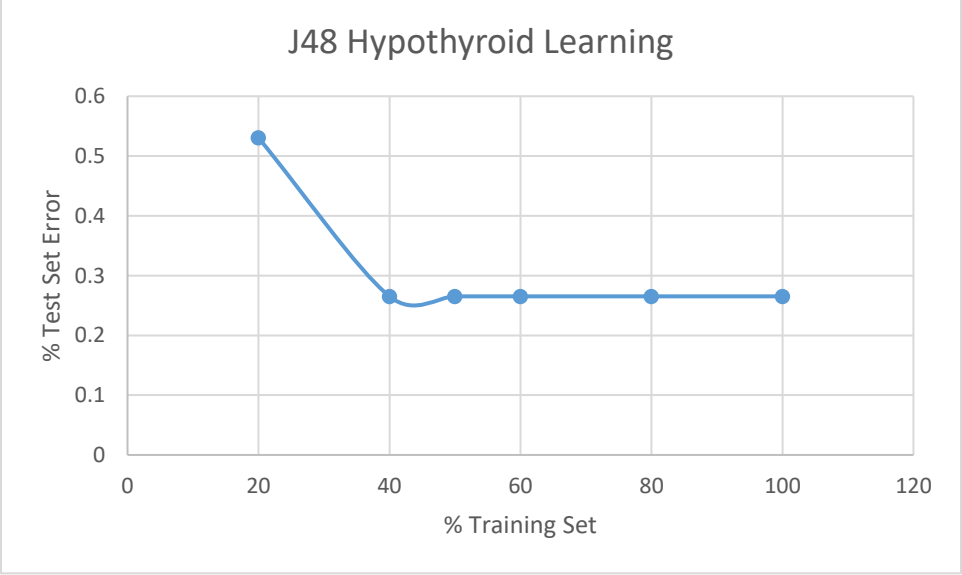
Supervised Learning Algorithm Comparison

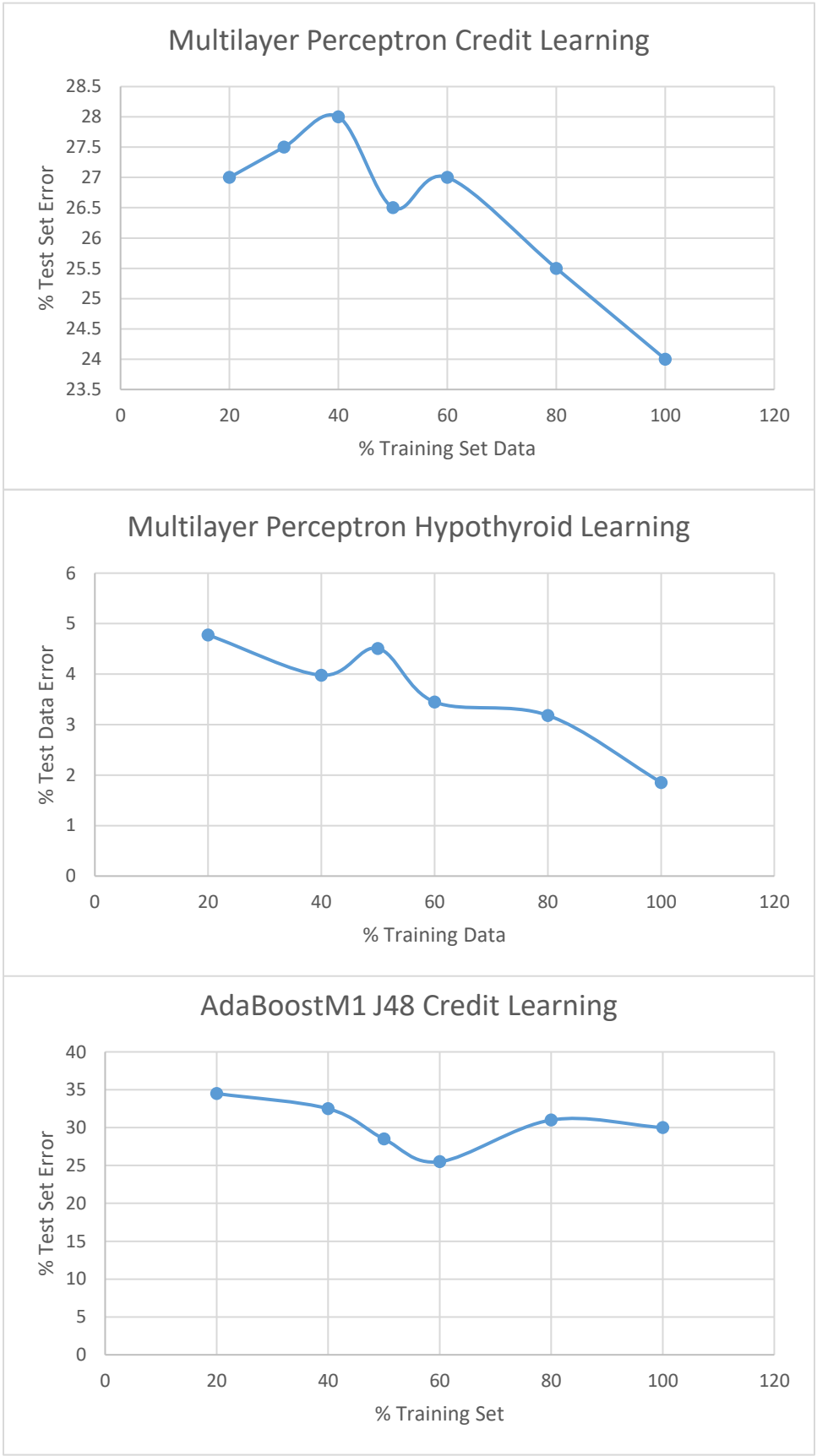
This experiment tests the performance of five different machine learning algorithms on two different data sets. The software Weka has algorithms for a Decision Tree (J48), Neural Network (Multilayer Perceptron), Boosted Algorithm (Boosted J48), Support Vector Machine (SMO), and k-Nearest Neighbors (IBk). The data sets were selected to highlight interesting similarities and differences between the five algorithms. The first data set is a hypothyroid disease dataset from Australia with 3,772 records. It has thirty classifying attributes and four hypothyroid result classes: negative, compensated, primary, and secondary. The second is a German credit rating dataset with 20 classifying attributes and 1000 records. It has two result classes: good and bad. This data is not classified as easily as the hypothyroid data; its error percentage data is consistently more clustered and more than ten times greater. These differences make the data sets interesting in comparing the different machine learning classification algorithms.

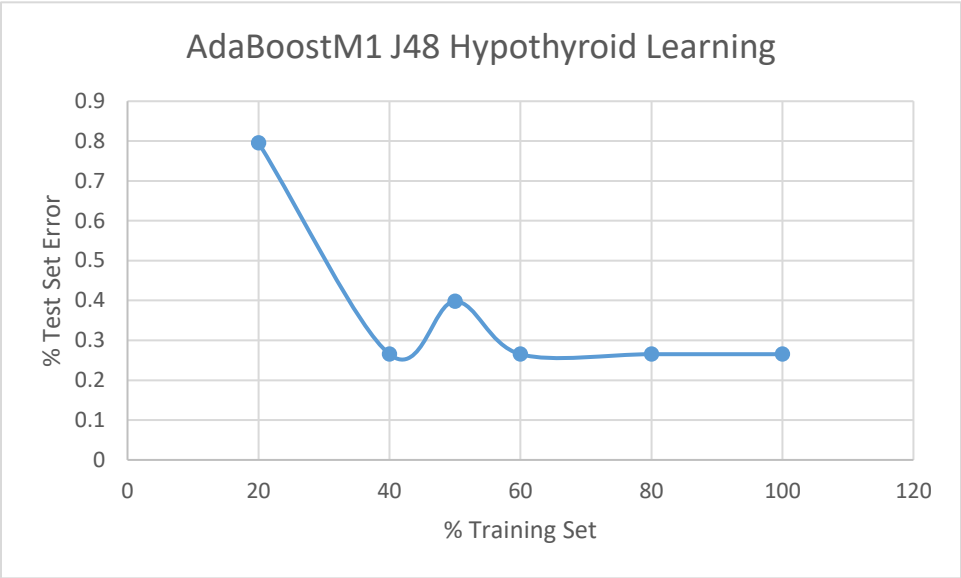
These data sets were each split into a testing and training set with percentages 20% and 80% respectively. Each training set was then further partitioned into 80%, 60%, 50%, 40%, and 20% of the data. 20% of the available training data is 16% of the total available data; this is 20% smaller than the test set size which gives lots of over-fitting bias. The complete training set was run through each of the five algorithms changing one parameter at a time to find an optimal set of parameters. This optimal set of parameters was then used to train each algorithm on each training set partition. No cross-validation was used due to the limitations of the ABAGAIL

software, but this would help to minimize overfitting error. Overfitting occurs when the neural network models the training data too well and thus does not extrapolate well to a testing set. These trained algorithms were then used to classify the unseen test set. All algorithms showed a general trend of a decrease in error given more training data as evidenced by the given learning curves.






Multilayer Perceptron Credit Learning
% Test Set Error
28.5
28
27.5
27
26.5
26
25.5
25
24.5
24
23.5
0
20
40
60
80
100
120
% Training Set Data
Multilayer Perceptron Hypothyroid Learning
% Test Data Error
6
5
4
3
2
1
0
0
20
40
60
80
100
120
% Training Data
AdaBoostM1 J48 Credit Learning
% Test Set Error
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
120
% Training Set



In the case of the credit data set, the J48 and AdaBoostM1 J48 algorithms do not suffer much test performance given smaller amounts of training data. Even as low as 20% of the available training data does not incur much performance loss. By contrast the hypothyroid test data set experienced a sharp increase in error when exposed to the 20% training data set. These two algorithms' error percentages asymptote much more quickly than the Multilayer Perceptron and IBk algorithms. Multilayer Perceptron and IBk perform much better given larger data sets while decision trees do not need as many examples to make good inferences. The variances in the linearity of the learning curves is explained by over-fitting the training data. When testing the unseen test set, the algorithm could have been trained to model the training data too well to effectively generalize it. Smaller amounts of training examples cause more overfitting to occur as the algorithm adapts to a more limited scope of data. This overfitting results in increased error; ten fold cross validation was used in testing all algorithms to mitigate this effect.

Lowest Percent Error Achieved

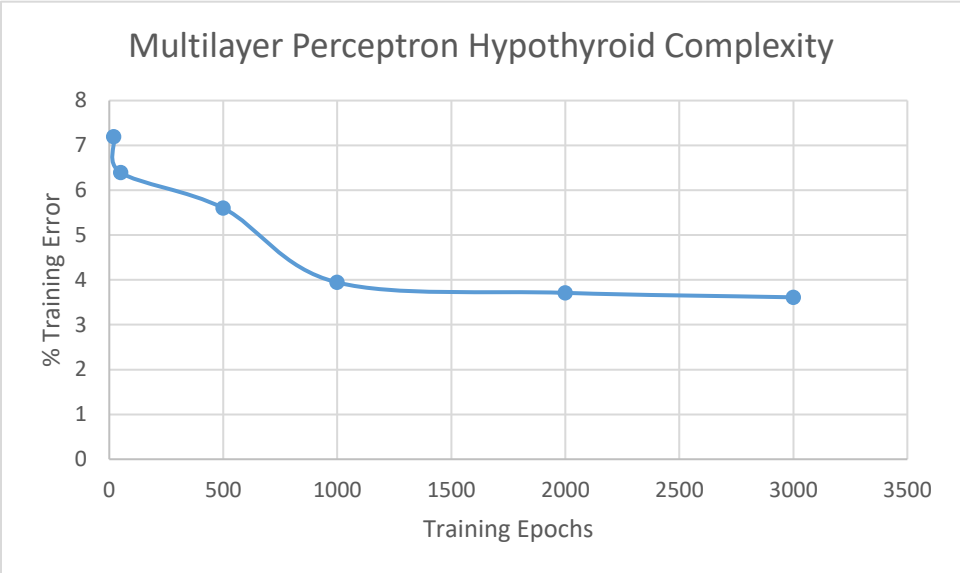
	J48	AdaBoostM1 J48	Multilayer Perceptron	SMO	IBk
Hypothyroid	.2653	.2653	1.8568	1.7241	5.4377
Credit	24.5	25.5	24	25	26.5

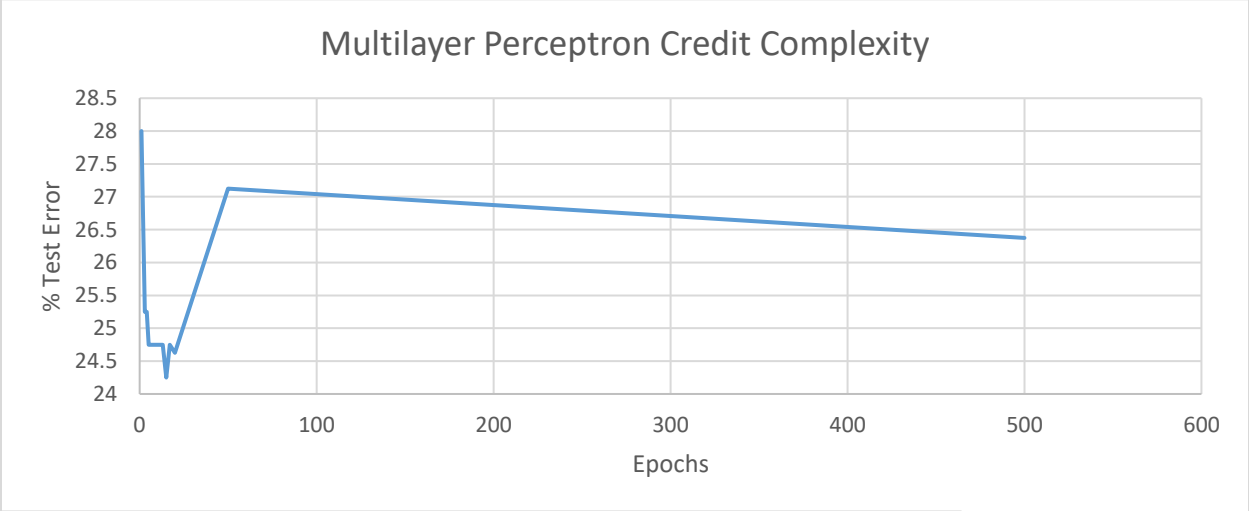
The Multilayer Perceptron neural network algorithm gives the best performance on the credit score data. It gives a lowest percentage error of 24% using 100% of the available training data. The slope of the curve is also sharply pointed downward which shows more data would give even better classification results. This sharp downward error slope is also present in the hypothyroid data, but with the available training data it yielded an error seven times greater than the J48 algorithm. While the other algorithms did not take long to run, the neural network took a long time to process and train on the input data because of the algorithm's complexity. Multilayer Perceptron trains the algorithm a specified number of epochs and testing shows a greater number of epochs gives less error. Because the hypothyroid tests' change in error asymptotes heavily after 1000 epochs, this number was chosen as the optimum parameter. Further epochs linearly increase the training time of the algorithm to levels which are unnecessarily time consuming. In the case of the credit data, the curve has a concave shape with an optimum number of epochs, 20, at the bottom. The optimum learning rate is also found at the lowest point on its concave curve.

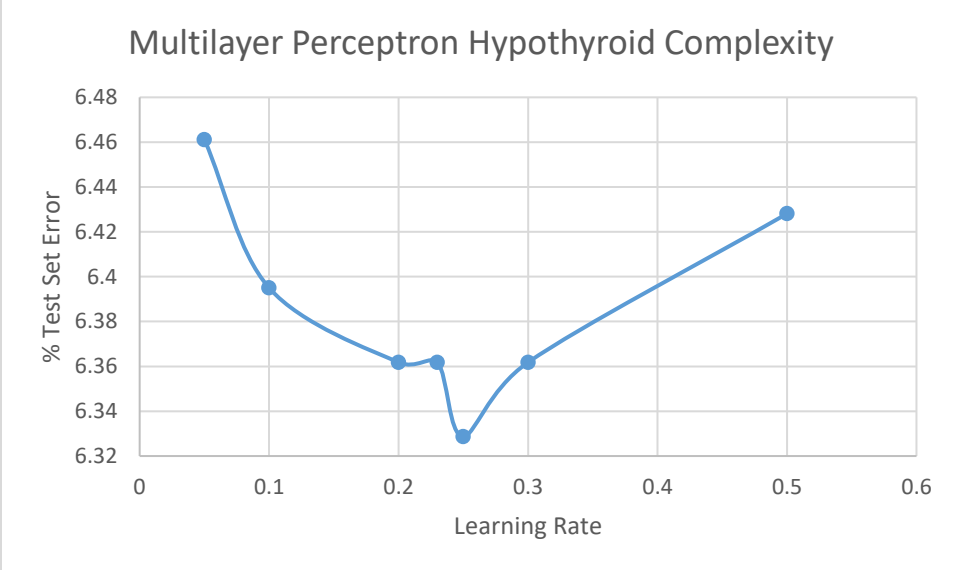
Algorithm Training Time (seconds)

	J48	AdaBoostM1 J48	Multilayer Perceptron	SMO	IBk
Hypothyroid	.01	.27	42.95	7.99	~0

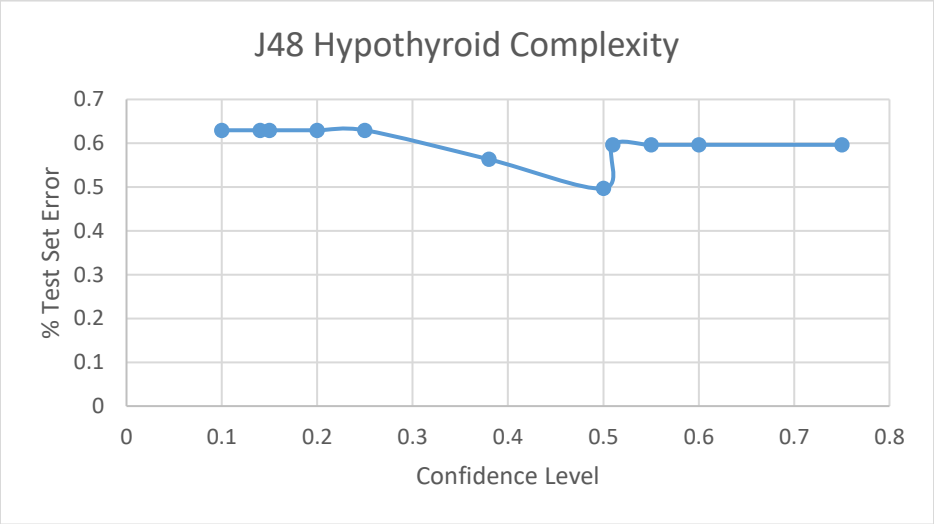
Credit	~0	.05	.63	2.26	~0

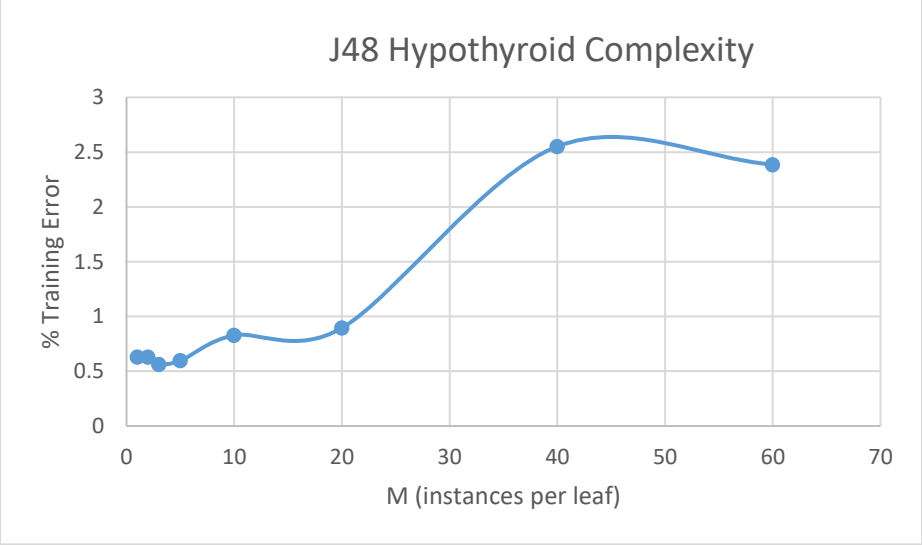


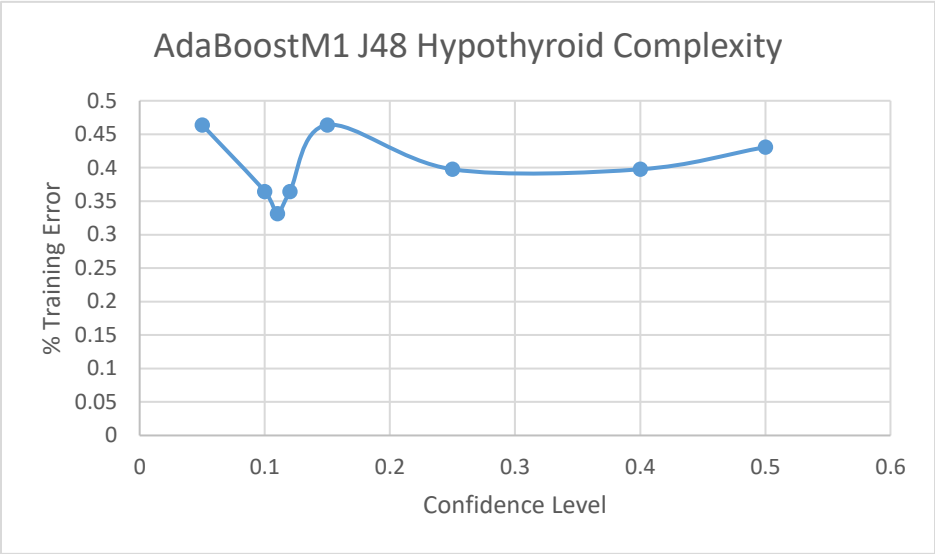




The J48 algorithm and AdaBoostM1 J48 algorithms give the best performance for the hypothyroid data set. Both algorithms give an error of .2653% using all of the available training data. The first parameter for this algorithm is the confidence level, a fraction which prunes the decision tree less as it rises. This parameter has a particular level which gives a better accuracy than anywhere else on the curve. In the case of the hypothyroid, this was at C=.5. By contrast, increasing the parameter M, the number of records in each decision tree leaf, steadily increases error in the test set. These algorithms trained the second fastest out of the five and were very close to zero seconds.

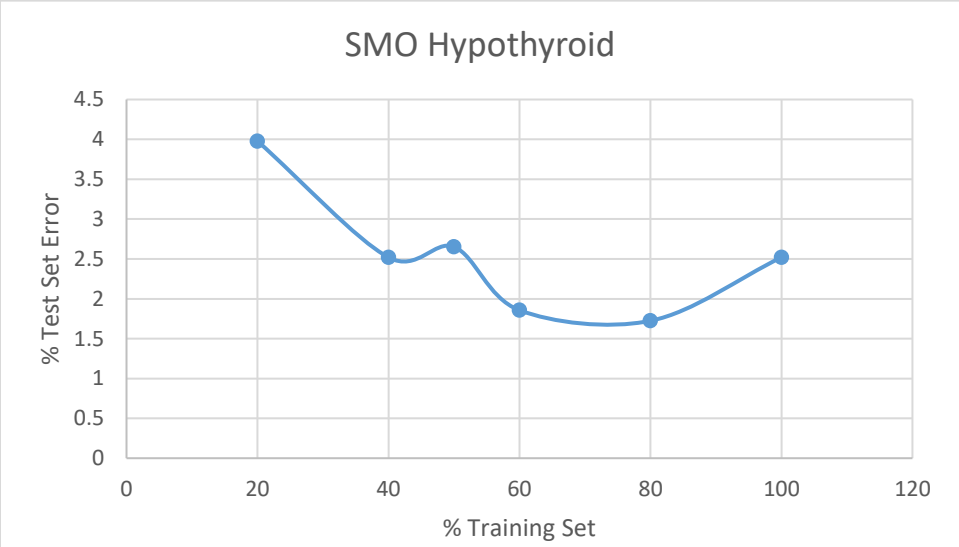


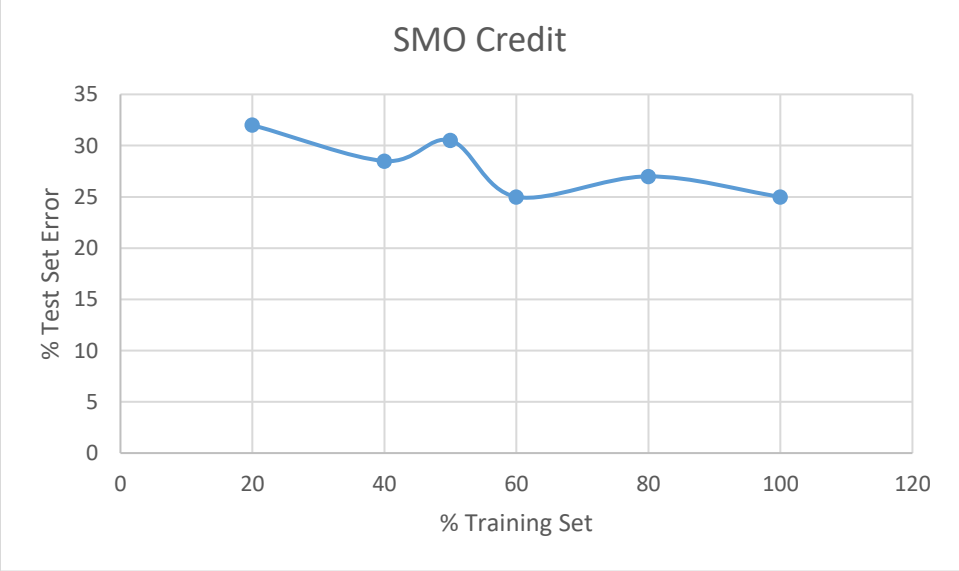


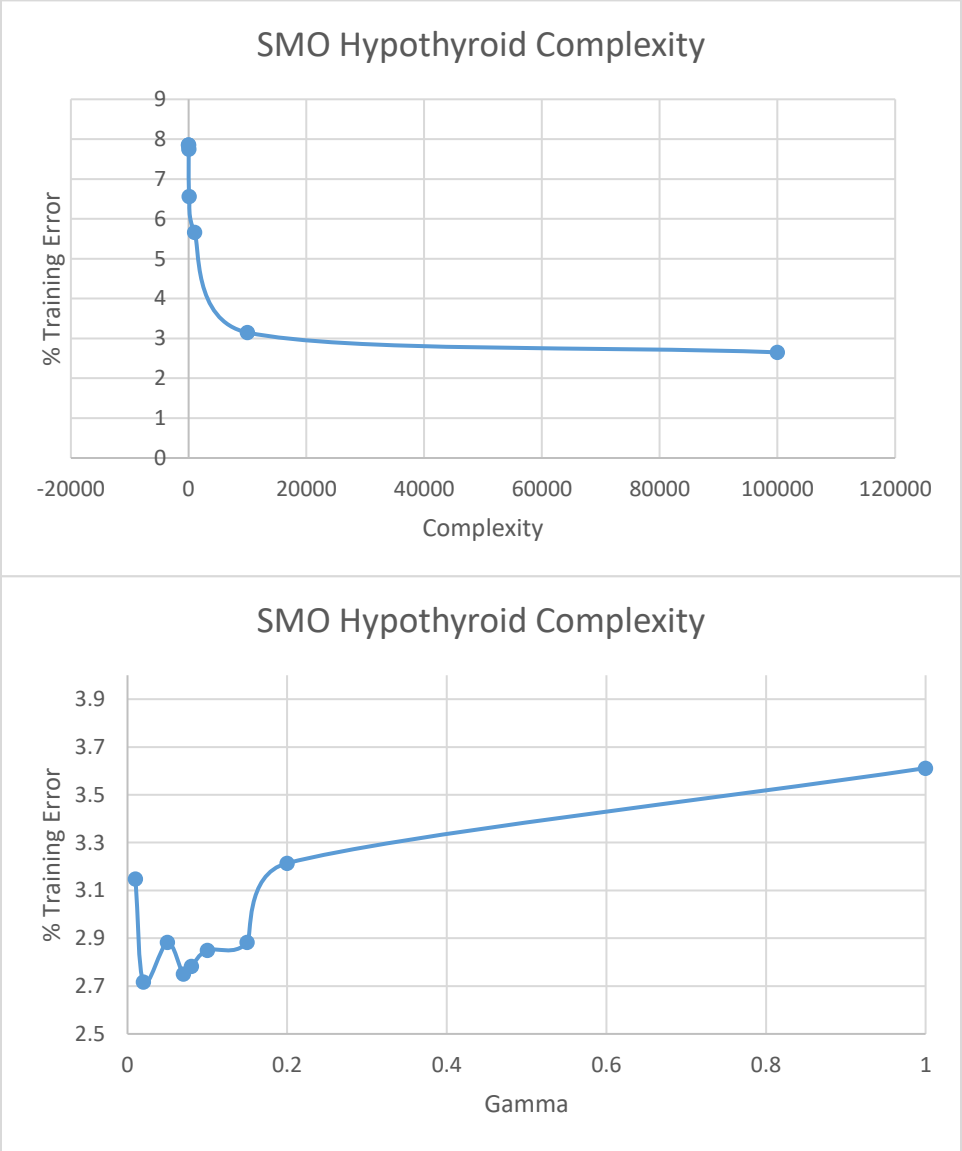


The Sequential Minimization Optimization algorithm (SMO) with the RBF Kernel displayed mediocre performance. When using the hypothyroid data set, it gave a lower error when using 60% and 80% of the training data than the complete set. When using the credit data set, the 60% and 100% tied for the lowest percent error. The concave upward shapes given here are the result of the algorithm over-fitting the training data. At 100% hypothyroid training data, the algorithm overfits and gives a higher error than at 60% and 80%. These lower percentages do not match the training data as closely as 100% and thus do not generalize this noise to the unseen test data set. More training examples would mitigate this effect more than cross-validation alone. The SMO parameters were tested for optimality. The parameter tested first is complexity, the number of support vectors used in determining a separation hyperplane between result classes. Increasing this parameter increases the amount of time require to train the network. Changing this parameter by powers of 10 yielded the curve shown below. Increasing complexity decreased error until about 10,000. Testing with a complexity of 100,000 yielded more error than 10,000 and also drastically increased the training time of the algorithm.

The gamma parameter also heavily modified the data. It was found through testing that a high complexity and low gamma gave the best results. For the hypothyroid data set, this was 100,000 complexity and .01 gamma. For the credit data set, this was 10,000 complexity and .001 gamma.

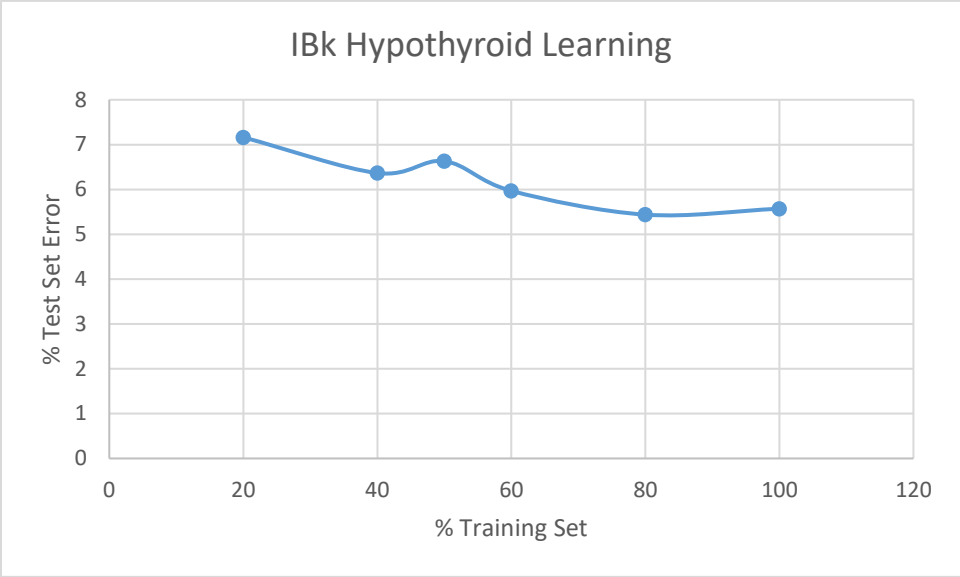


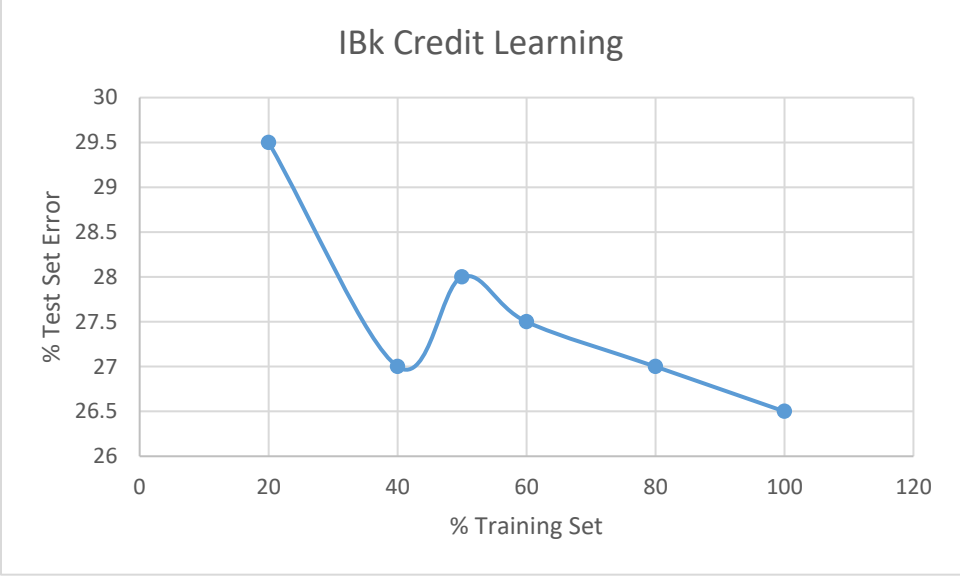


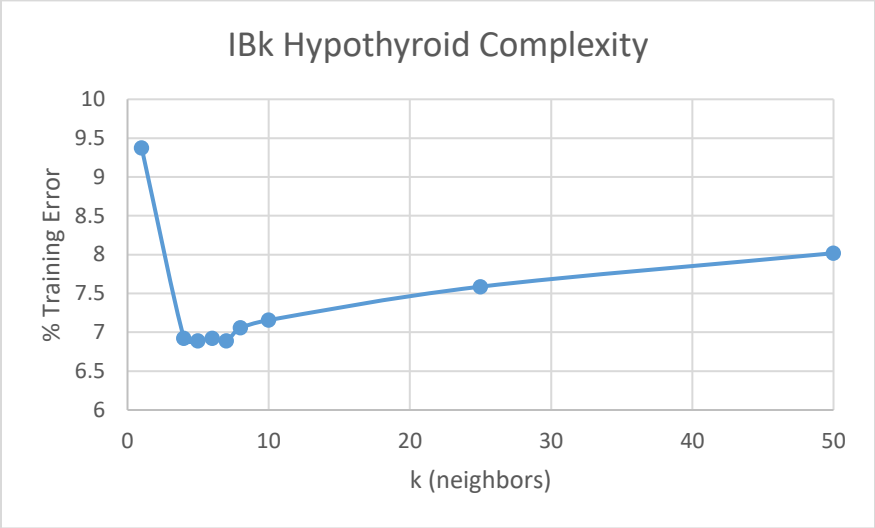


The IBk k-Nearest Neighbor algorithm performed the worst for both data sets. Using the hypothyroid data at best the algorithm gave twenty times more error than the J48 algorithm. The credit data, by contrast, had ten percent more error than the J48 algorithm at best. The data decreased steadily when exposed to more hypothyroid data and decreased sharply when exposed to more credit data. As evidenced by the slope of the credit curve at 100%, the algorithm could perform much better with the credit test data if it was given more training examples. The optimal amount of neighbors, k, was decided through testing. It was

found to be 5 for the hypothyroid data and 4 for the credit data. The plot showing error for each value of k is concave up in nature for both data sets, meaning an optimal k lies at the bottom of each curve. This algorithm was the fastest and took very close to zero seconds to run for both data sets.







These five algorithms all had varying performance on the data. For the hypothyroid data, the decision tree and boosted decision tree algorithms performed the best with the lowest amount of error. The J48 algorithm without boosting performed the best in terms of time as well, and would thus be the best algorithm for interpreting the hypothyroid data set. Decision trees model both datasets very well for very little training time even when given as little as 20% of the data and would also work well for many other datasets given this versatility. By contrast, a neural network models data well given lots of examples and training time. Nonetheless, the neural network performed best on the credit score data set which has fewer examples than the hypothyroid data set. This is due to specific properties unique to the credit data set and does not generalize well to other data sets as the hypothyroid data's best algorithm is not a neural network. As evidenced by the learning curves, cross-validation could not remove all the over-fitting bias present; only a training set of all relevant data could do so. Better classification generalization could be achieved with all five algorithms using more training examples as shown by the prominent downward error trend on all learning curve graphs.