Jeffrey Minowa
Professor Charles Isbell
CS 4641: Machine Learning
10 February 2019

Supervised Learning

Overview: In the following paper, I will be describing how data was processed, and how five distinct supervised learning algorithms learn by following trends from information and all algorithms were taken from WEKA https://www.cs.waikato.ac.nz/ml/weka/.

- "Adult Data Set" from UCI Machine Learning Repository is a dataset of 32,000 instances with attributes that define a person's income of <= 50k and >50k using continuous characteristics.
- "Titanic: Machine Learning from Disaster" from Kaggle is a dataset of 800 instances that has a binary classification of surviving or dying on the Titanic.

The following terms will be used throughout the paper: testing and training will be used to describe randomized data that was trained on 80 % of the data and tested on the remaining 20% and trained on 80% and tested on the same data respectively, and the accuracy will be defined as the correctly classified instances / total number of instances.

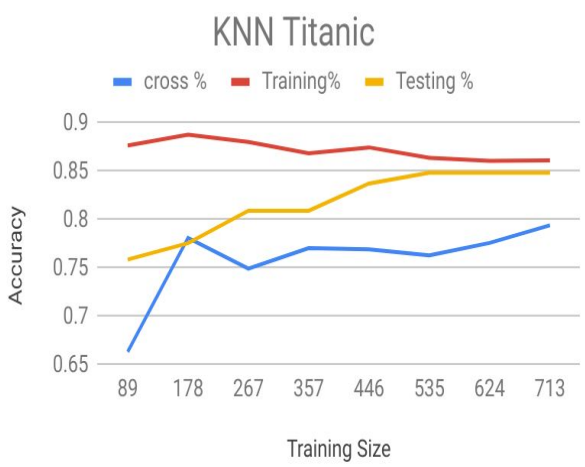
Preprocessing: In the preprocessing stage, for Titanic, certain columns of information were deleted from the dataset because they were deemed as noise or insignificant in classifying survival. Columns that were dropped were the ticket id of the passenger, the port embarked. Cabin number was also dropped because more than half the values were null.
Regarding the adult data set, values were grouped together to remove unnecessary distinct variables such as the multiple age values. Though it is worth noting that the grouping of variables may have caused inherent bias towards where the split of group occurred.

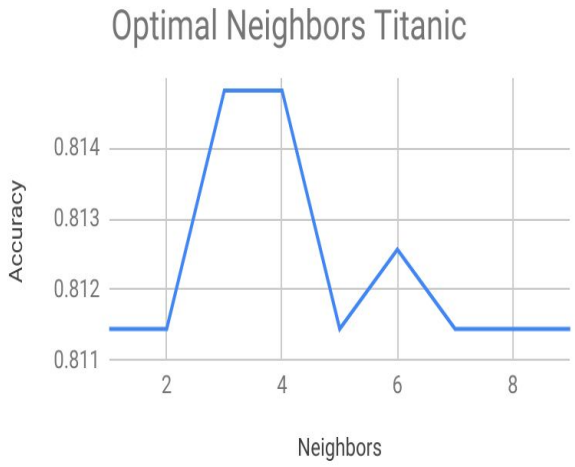
Hyperparameter tuning: I tuned the available hyperparameters to find the optimal collection of values to produce the greatest accuracy. This was done iteratively for each classifier that had

available hyperparameters to tune i.e. KNN, neural net, and decision tree and comparing the accuracy by using cross validation of 5 folds.

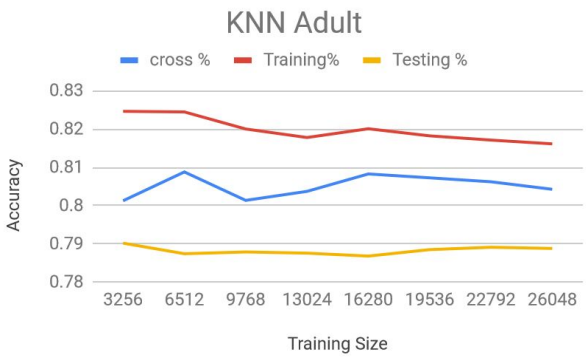
KNN: K-nearest neighbors algorithm is a lazy learner that performs well given its simplicity. The idea behind KNN is to take input values, and then classify test values based on the k neighbors that surround the new value by proximity calculated using 1/distance. This allows for the tuning of the hyperparameter k with k being tested from 0 to .1 of the test data size.

For the Titanic dataset, the tuning of the k can be seen as overfit from a low k and underfit from a large k, yielding 3 as the optimal k. In regards to the data, it begins to converge as the training size increases. This makes sense because as more values are learned, an increased number of similar values are presented for the new data to be classified against.



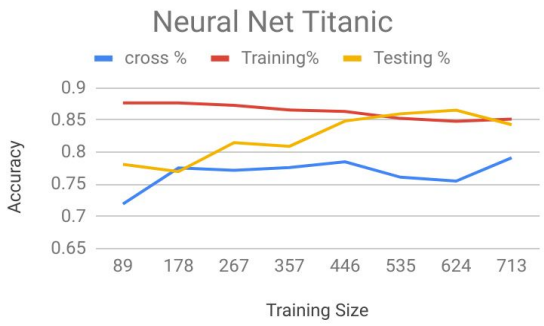


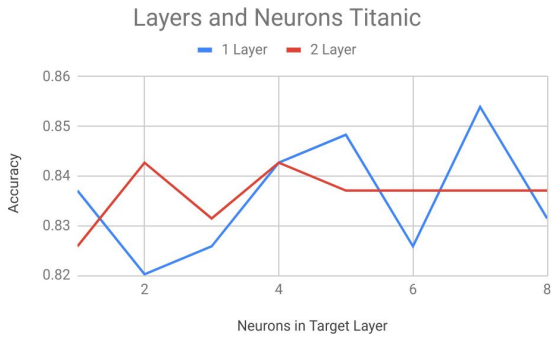
For the adult data set, there is a consistent accuracy which hovers around .8 accuracy. Because this is a larger dataset, the algorithm does well when large generalizations are made. Yielding a larger accuracy as K grows.



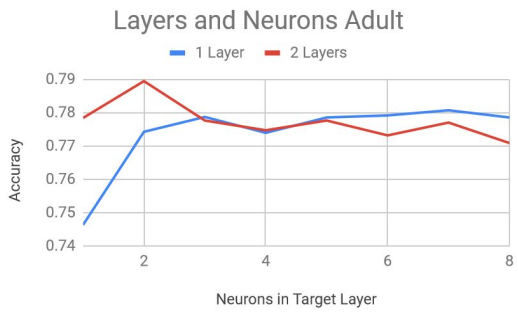
Neural Net: Neural nets are theoretically capable of classifying any function by imitating functions via matrix manipulation. For this analysis, a multilayer perceptron was used to classify data, a feedforward neural network model through a weighted graph with backpropagation for training. This allows for tuning of the hyperparameters for the number of hidden layers as well as the number of perceptrons in each layer.

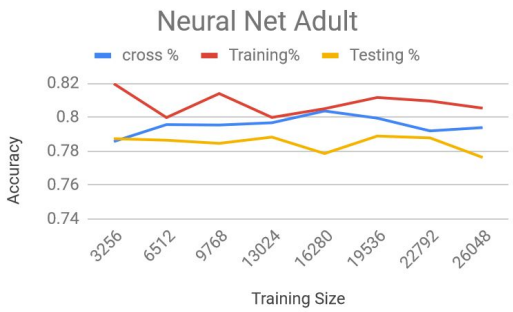
For the titanic dataset, the best combination of hidden layers and perceptrons is a layer of 7 perceptrons. A peculiar aspect of the graph is that, at points, the testing data does better than training data. Some reasoning for this is the may have been a biased split between the train data and test data or that neural nets become generalized classifiers, resulting in a loss of training accuracy while accurately determining testing data.



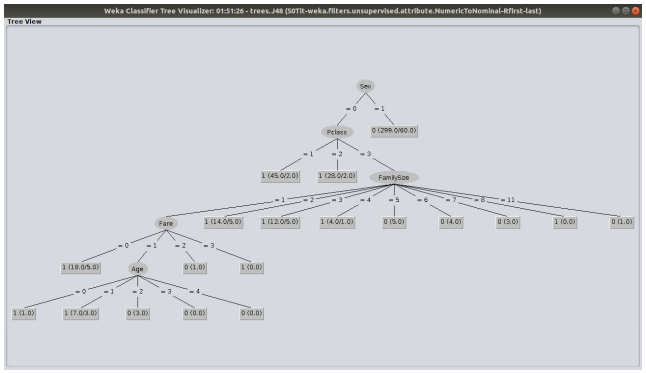


For the adult dataset, the best combination of hidden layers and perceptrons was 7 perceptrons for the first layer and 2 perceptrons for the second layer. I also determined that the data had no overfitting, as the two lines followed the same trend and did not look like it was diverging, resulting in testing accuracy hovering around .78 accuracy.



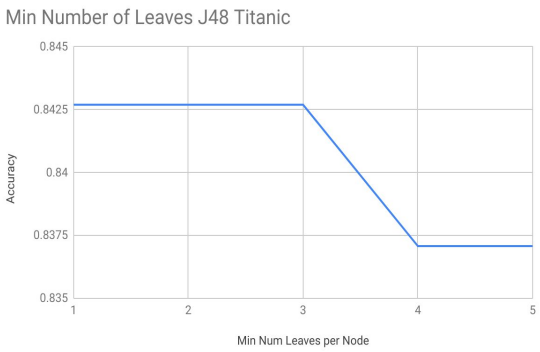


Decision Tree: A decision tree is a supervised learner that predicts using a tree-like model. The tree greedily splits to maximize the chance of reaching a conclusion. It is also worth noting that the tree can be visually represented, allowing for users to see where the splits occur as opposed to the black box of machine learning. Depicted is the result of one of the tree's visualizations.

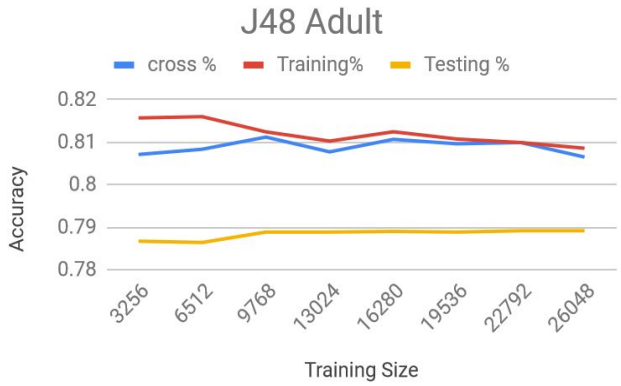


Pruning: Pruning is used to decrease the depth of the decision tree, restricting the depth removes unneeded nodes and redundancies found in the tree which improves classification by reducing overfitting. The minimum number of leaves per node was a tuned hyperparameter with the decision tree, J48. If the minimum number of leaves is too large, underfitting occurs, and if the minimum number of is too small, overfitting occurs.

For the titanic dataset, the graph shows the data becoming more generalized as more information is given to the tree.



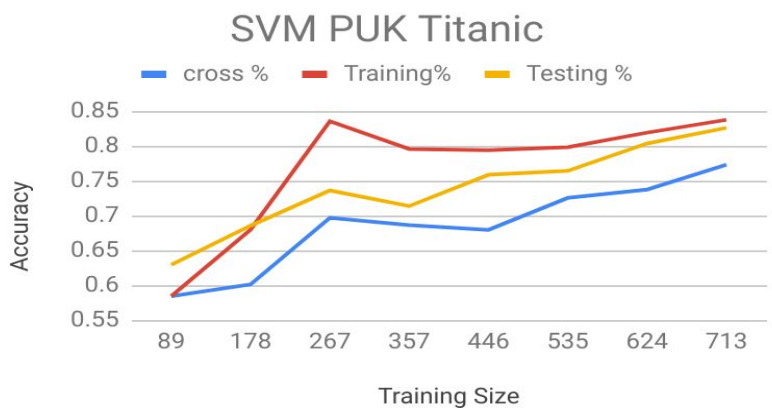
For the adult dataset, parameter tuning had no effect on the accuracy, and the testing consistently performed less than the training with no signs of convergence or divergence.



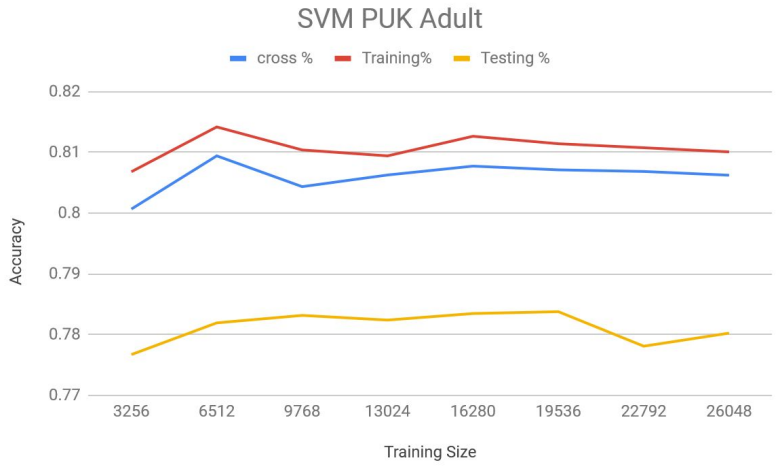
SVM: SVMS use kernels to map objects into different dimensional spaces and then uses a support vector to split the data into two classifications. The goal of the support vector is the create as wide a gap as possible between classified data points to both classify different data as well as to leave room for new instances that have yet to be seen.

Here we used two different kernels to split the data: Puk and NormalizedPolyKernal. NormalizedPolyKernel takes the values of coordinates and divides them by the normalized value of the coordinates.

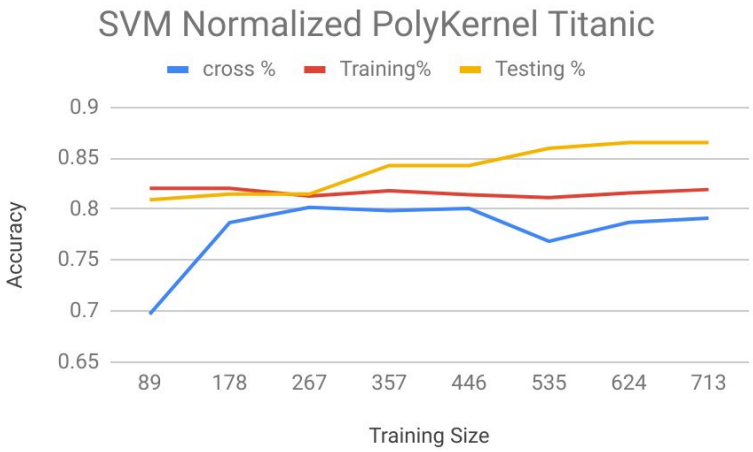
Puk: For the Titanic dataset, Puk showed an upward trend with both the training and testing and also became closer to converging as more data was learned.



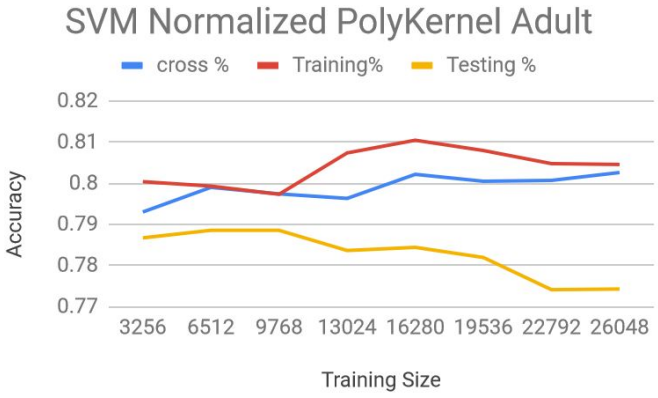
For the adult dataset, the training and testing data had similar trends, and the testing was consistently below the training. The data shows no signs of overfitting or underfitting.



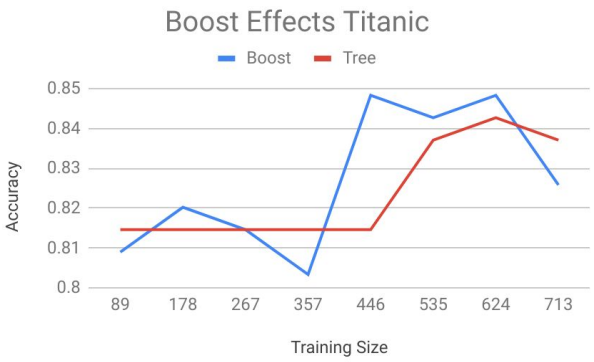
NormalizedPolyKernel: For the Titanic dataset, the training data does worse than the testing data; the training data also does not decrease in accuracy. This shows that the classifier has been generalized well enough that the testing data performs well.

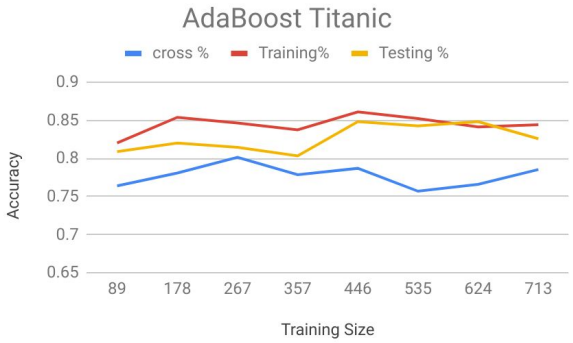


For the adult dataset, the train and test start to diverge. This may be a result of overfitting which would cause the increase in training size to make the accuracy of test worse.

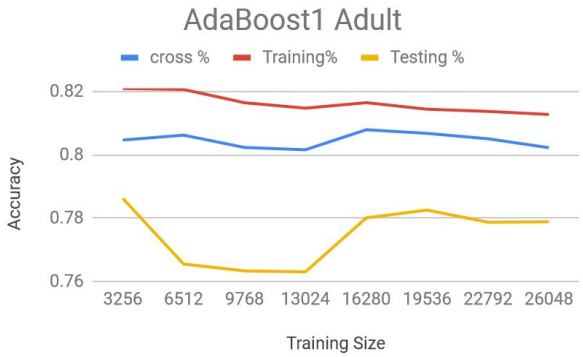


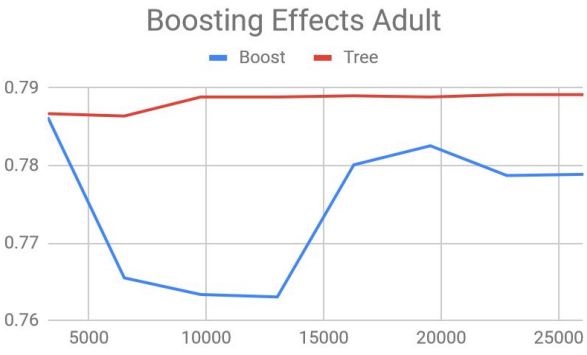
Boosting: Boosting is the use of a classifier and weighing difficult instances to answer higher than easy answers, where difficult instances are those classified incorrectly, and easy answers are those classified correctly. In this case, we use the J48 as mentioned earlier as our classifier for boosting. The performance of boosting should be higher, granted the classifier is given a hard question more more often as can be seen in the Titanic dataset.





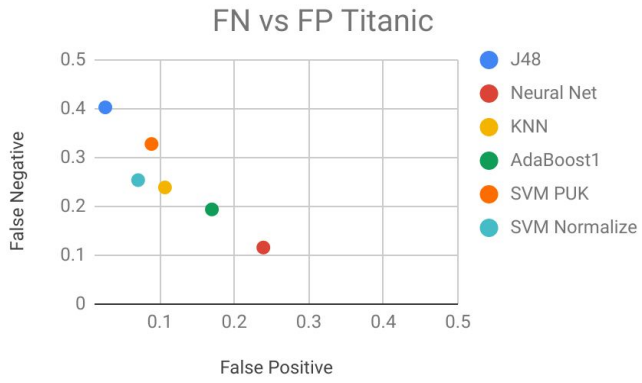
Though the downside of this algorithm is that noise heavily affects the algorithm as can be seen in the adult data set. Because of the number of unknowns in the adult attributes, the booster cannot determine where to classify the unknown data. The testing underperforms on the adult dataset, and is one of the worst classifiers for the adult data set for this reason.



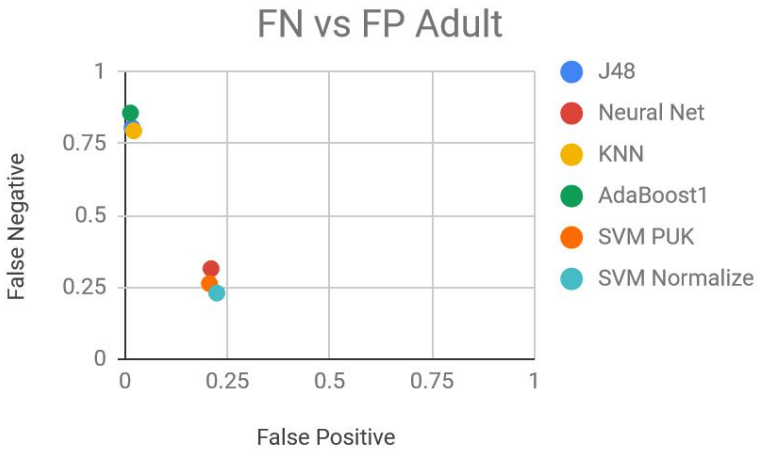


False Positive vs False Negative: Another metric for determining the ability of a classifier is the look at its false positive, false negative score. Overall, minimizing this value is the goal of any classifier. Though biases can be found and also preferred depending on the dataset. For example,

the neural would be the best classifier for the titanic if looking for minimizing the false negatives. Whereas the J48 would be the best if the goal is the minimize the number of false positives. Having a worse false positive and false negative rate than other classifiers, the classifier, SVM Puk, is dominated by the SVM Normalized classifier.

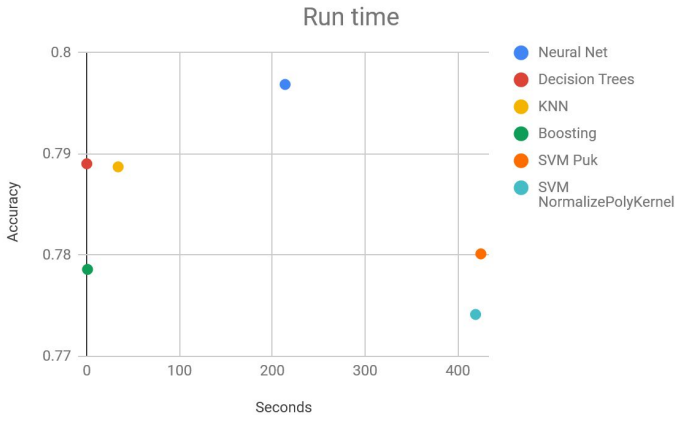


For the adult dataset, within the clustering of the false positive, false negative scores, the biases are less obvious. Specifically, the J48, Adaboost, and KNN have a large false negative rate with a minimal false positive, and the SVM Normalize, SVM Puk, and Neural Net have a more even distribution about the .25, .25 coordinate.



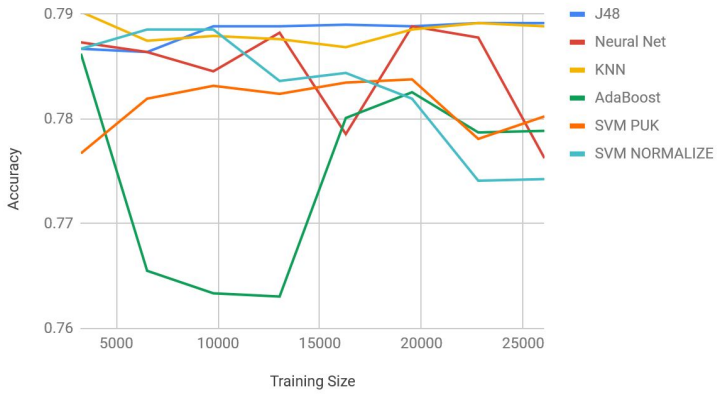
Time vs Accuracy: Another factor to take into consideration is choosing which supervised learning algorithm based on amount of resources needed. Given that some classifiers take much less time than others and the accuracy is relatively similar, there may be need to use the faster

algorithm. As can be seen from a 32,000 dataset, the SVM's take marginally more time than the Decision Tree and KNN.

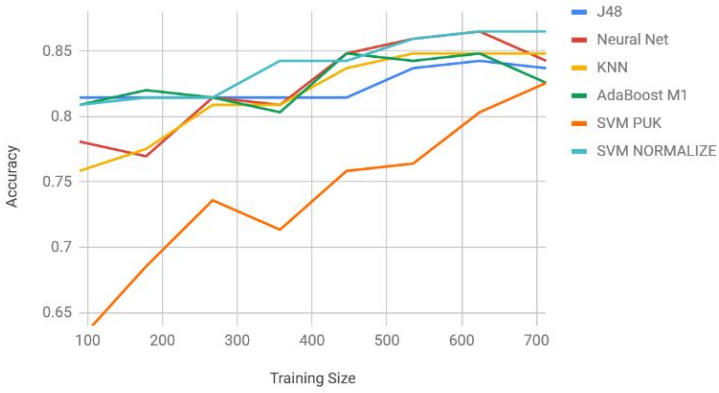


In conclusion, each dataset needs its own tool to best classify the data to maximize metrics. There is no one size fits all algorithm that will cover all datasets perfectly without overfitting. Given that these two datasets are extremely different in terms of attributes, size, and discrete vs continuous, their best classifiers are expected to also differ.

Adult All



Titanic All



Acknowledgements
All of the algorithms I used were taken from WEKA and I give fill credit to the creator for implementing all of these machine learning algorithms.
Datasets:

Titanic:
https://www.kaggle.com/dmilla/introduction-to-decision-trees-titanic-dataset?fbclid=IwAR0WZdBrq296WxXGUpWrSpdBHXXMN8b6Mp0uSK9C_rkd_aniSmsq1OCDiM4

Adult: http://archive.ics.uci.edu/ml/datasets/Adult