Assignment 1 – CS 4641

Keertana Subramani

Supervised Learning

I. Description:

The two datasets I chose to work with are: 1) The Blood Transfusion[1] and 2) The breast Cancer[2] datasets. Both datasets tackle very interesting prediction questions.

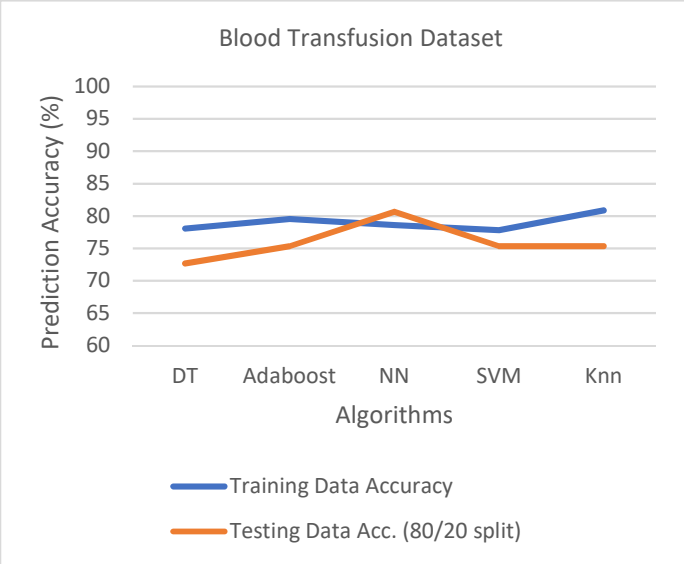
Dataset 1: Blood Transfusion Dataset: This dataset has 4 attributes (with 748 instances) that relate to a person's blood donation history and predicts using these whether or not they donated blood in March 2007. This data is interesting because there is a much higher probability overall that a person didn't donate blood than that they did, and so the error rates of predicting a person as a non-donor even though they did, in fact, donate is high. It was relatively clean, nominal dataset with no missing data, and large enough to allow meaningful comparisons across different amounts of training input and across different algorithms. Overall, predictive accuracy ranged from 70.56% to 80.67% (neural nets).

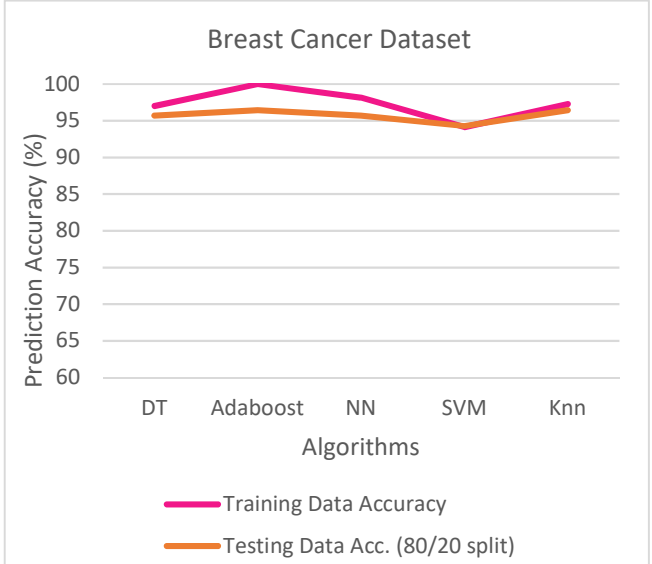
Dataset 2: Breast Cancer Dataset: This dataset uses a number of features in a breast-cell to predict if a tumor is benign or malignant. The data has 699 instances and 10 attributes- higher than in dataset 1 - allowing for meaningful difference. Unlike the previous dataset, this dataset has perhaps fewer outliers and less skewed distribution, for across all algorithms used, it allows for higher predictive accuracies (94-100%), and in real life, is shown to have predicted correct results for 174 consecutive patients.

II. Performance Observations across models:

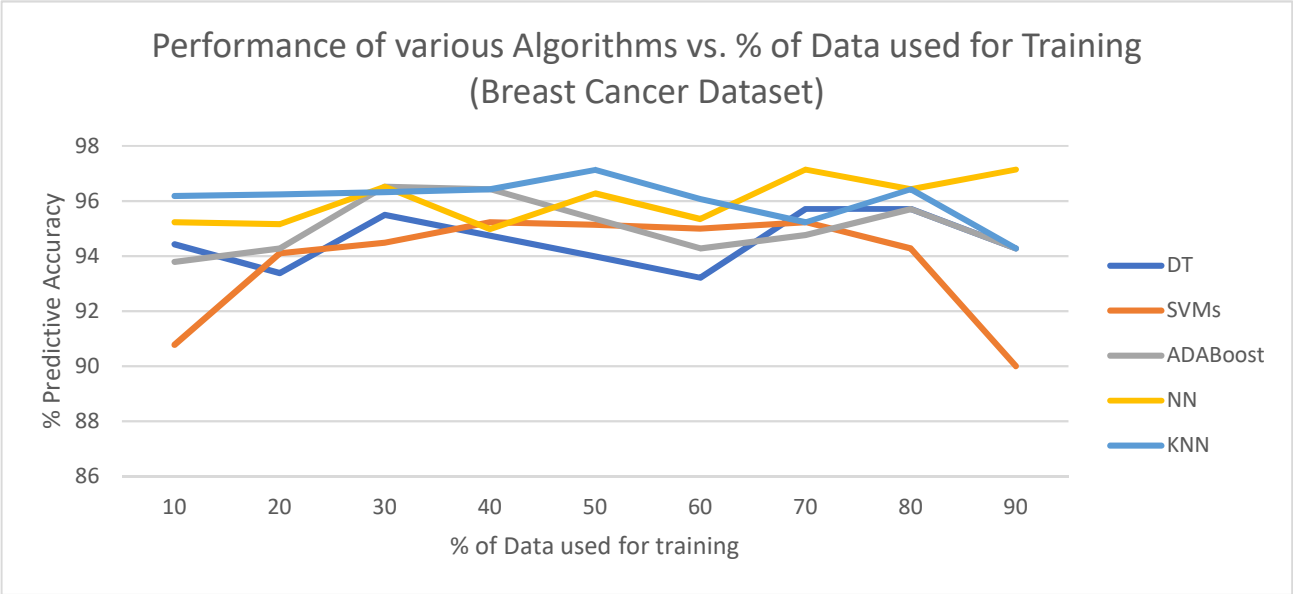
A. Prediction Accuracy of Various Algorithms on % of Training vs test data

We see in both datasets, that overall, the training accuracy is significantly higher than prediction accuracy on testing data. This is because the classification model is built on the training data, so will be easily able to make better predictions on the exact same data rather than on potentially different, unseen test data.

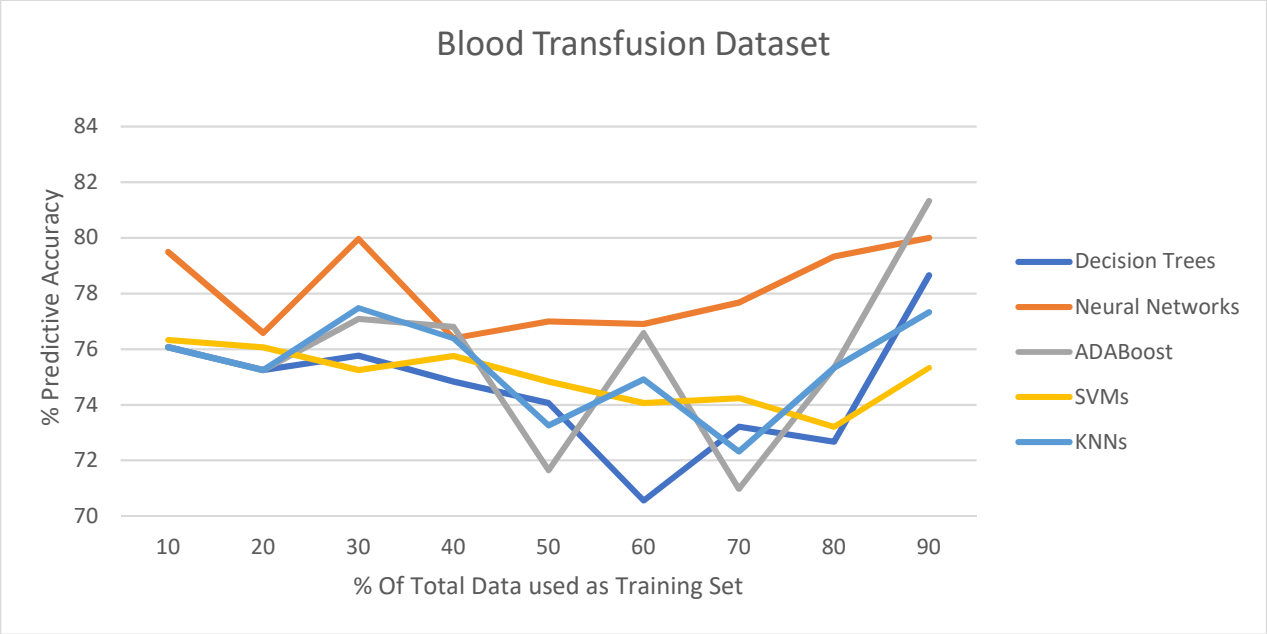




B. Predictive Accuracy of Different Algorithms vs % of data used for Training



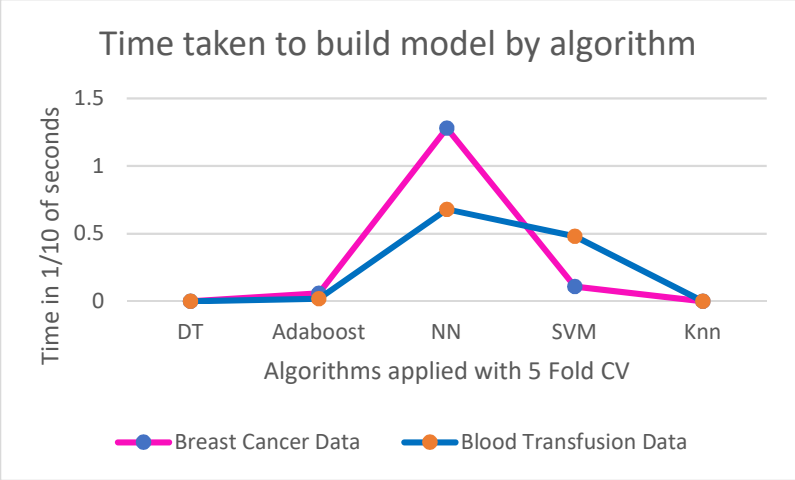
In the first dataset, we see that as the percent of training data is very low or very high , the predictive accuracies are at their highest. This could be because, with very little training data, the model is generalizable enough to predict new instances without overfitting, and with lots of training data, the model is well trained enough to be able to make suitable predictions on the small test sets it's given. Overall, for both datasets, **neural networks** seem to produce best prediction accuracies on test-data, which is plausible since this algorithm optimizes model performance through mapping with hidden layers (the number of which has been optimized here). Still, one would also expect Adaboost to produce the best results still it combines multiple models- and while doesn't consistently come out top, it does give the highest predictive accuracies for particular splits of the training-test datasets (90% in set 1 and 40% on set 2).



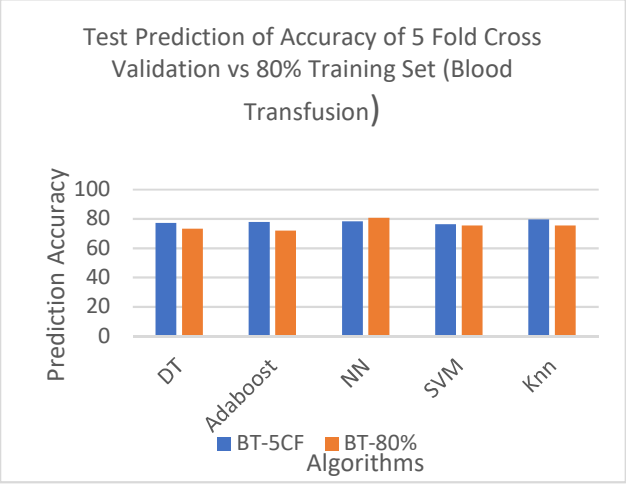
On the other hand for the breast cancer dataset, predictive accuracy peaks somewhere in the middle for most algorithms, between the 50 or 70% training-test splits. This could be because too little training data results in an insufficiently studied (underfitting) model and too much training data results in an overfitting classifier.

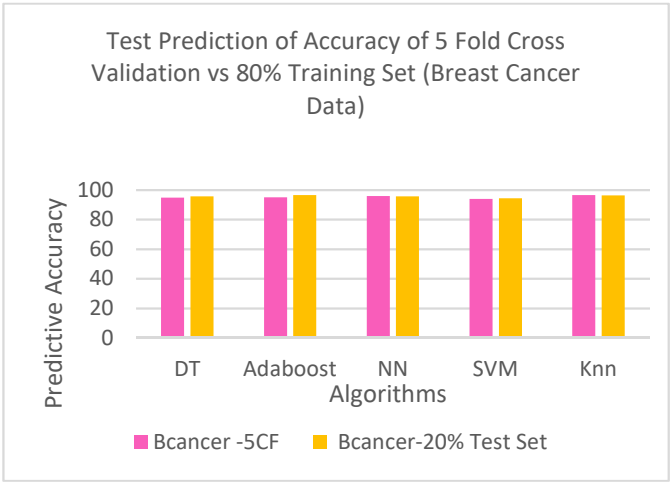
C. Time Taken to Build Classification Model by Algorithm Used

For both datasets, time taken to build the classification model is lowest for decision trees and k-NN algorithms (~0 Seconds). For both sets, neural networks take the highest amount of time – which makes sense because there are 3 hidden layers set for both NNs to give optimal predictive output, and it takes time to build a build with 5 layers. For breast cancer data, there are 5 nodes in each hidden layer (more than in the first dataset); hence higher model-build time. SVMs take the second highest time, perhaps because it is also a complex algorithm that performs optimization to result in the best dividing hyperplane.



D. Performance of data on 80-20 test splits vs using a 5-Fold Cross Validation model across algorithms:



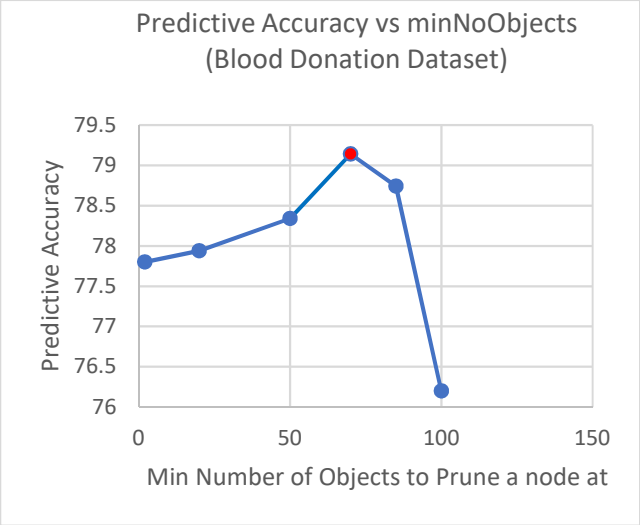


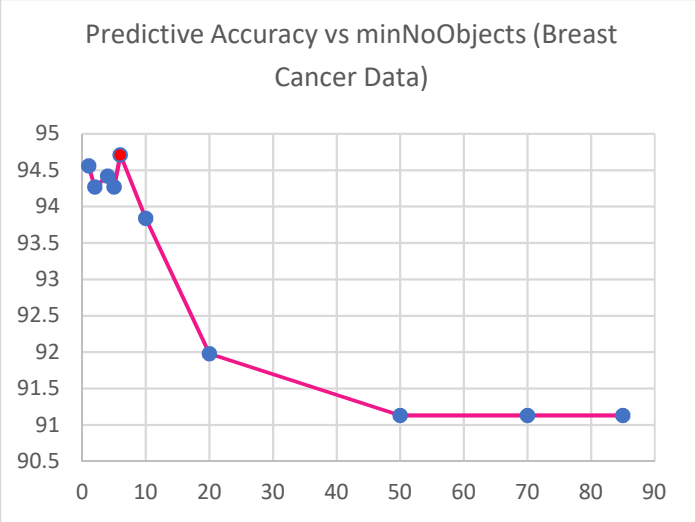
Overall, the 5-Fold Cross Validation model produced better results on test data than a single 80-20 training-test split of the dataset. The purpose of the cross validation model is to reduce overfitting and produce a more robust, generalizable model. This was the case, especially in the blood transfusion data set. However, the results of both kinds of data testing are more similar in the breast cancer data set, perhaps because the data is already random and robust enough that it produces a well-generalizing model with just one test-training split.

III. Analysis by Algorithm:

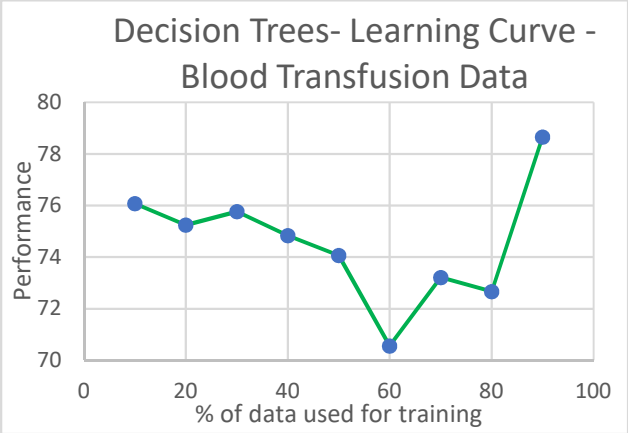
1. Decision Tree: I used Weka's J-48 algorithm and ran the algorithm with the optimal minNoOfObjects for pruning (n = 70 for dataset1, and n = 6 for dataset 2). I chose post-pruning as the means of optimizing the algorithm performance and chose minimum no. of instances to prune the leaf node on as the most important parameter that affects performance. This is because while post-pruning, allowing for leaf nodes with too few instances can result in the model overfitting the data, and having too many instances per leaf can result in over-generalization of the model, reducing its robustness to new test data.

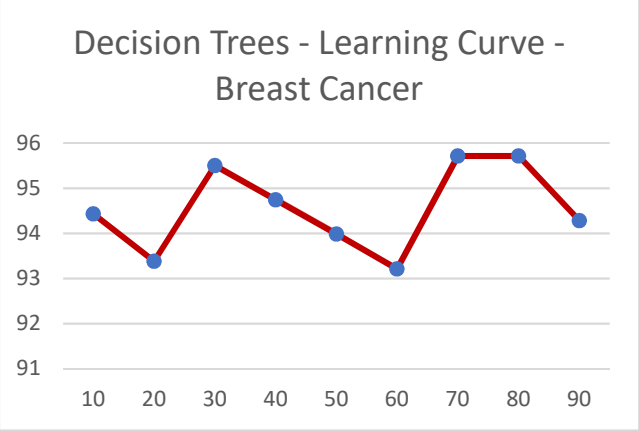
Min No of Objects: For the Breast cancer dataset, the optimal min No of objects is 6 and for the blood donation set, it is 70. As we increase the robustness of pruning on the blood donation dataset, one of the attributes (V3) is completely pruned away as not relevant to a sound prediction – which shows that simpler decision trees can also make good models with sound performance. Below are the charts showing predictive accuracy as we increase min no of objects.





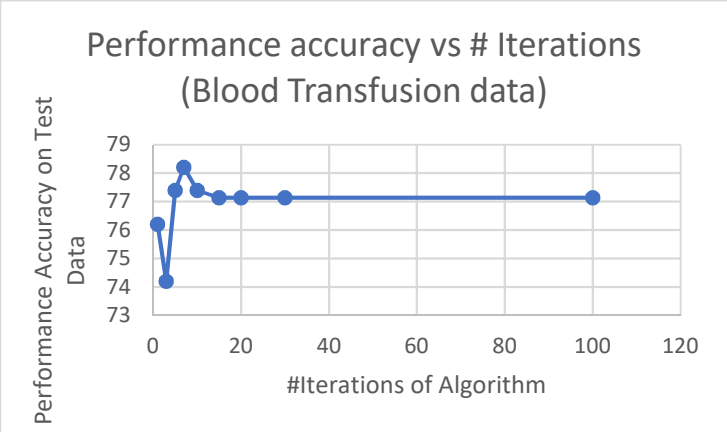
Learning Curves: The learning curve of the algorithm varies for both datasets, most likely because of the specific features that vary in both datasets, differing distribution and variability of data etc. Too little training data may result in suboptimal performance of model and too much training data, result in overfitting.

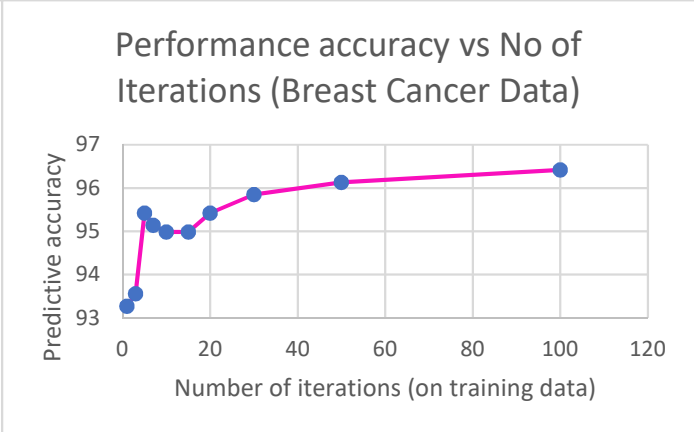




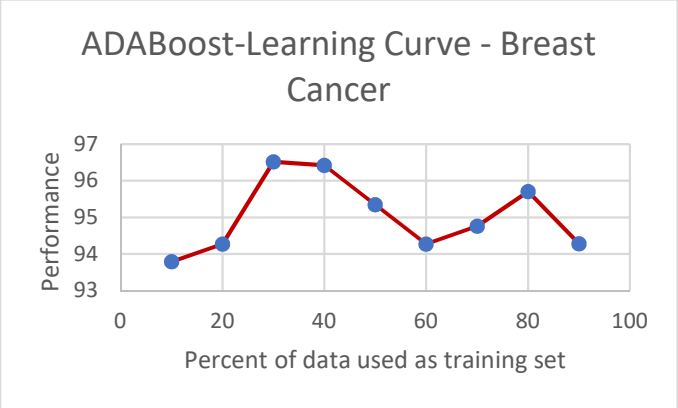
2. **Boosting**: I used Adaboost1 Ensemble Learning algorithm within Weka for this purpose. I used J48 decision trees as base algorithms for this ensemble methods. Results were expected to be much more robust with higher

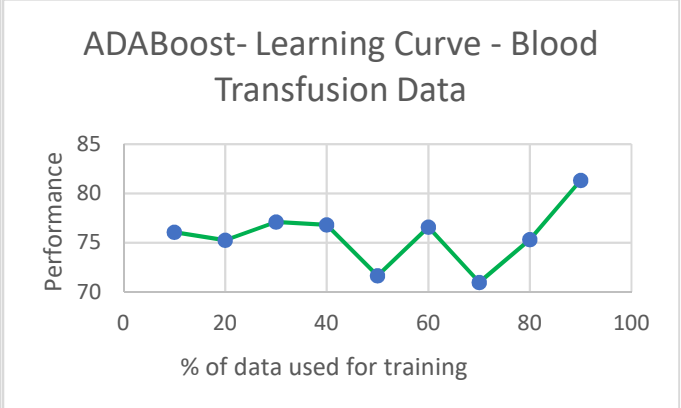
predictive accuracy for this algorithm since there this algorithm combines several component DTs to produce the optimal model. I used resampling of my data points while running the algorithm as this has shown[3] to train the data better. I also tested the performance of this data by increasing the number of iterations.





Number of iterations: As we see above, as the number of iterations increase, in both datasets, the predictive accuracy significantly increases. In the first dataset, it peaks at about 7 iterations whereas in the second, it goes up slowly after iterations increase beyond 30. This could be because ADAboost converges after a certain amount of iterations optimizing over different combinations of decision trees. As the number of iterations increase, ADAboost builds models that first reduce bias and then reduce variance using component decision trees. Below are the *learning curves* for Adaboost in both datasets.



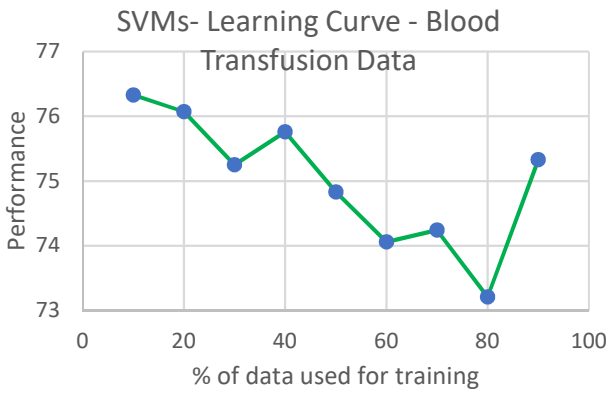


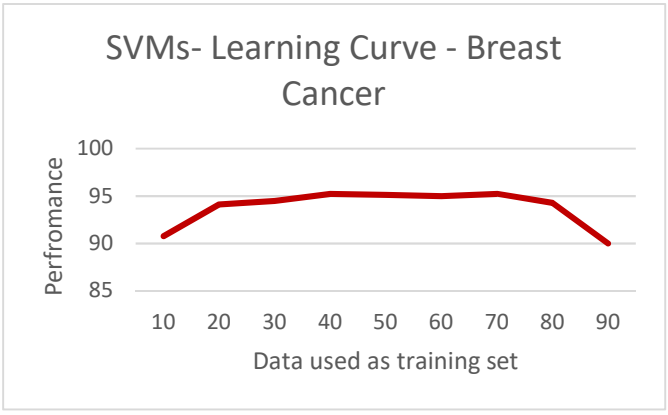
Learning Curve: In the learning curves above, we observe that with change in percent of data for training, the performance of the boosting algorithm improves, with best performance at 90% training data for set 2 and 30% training data for set 1 (perhaps bigger for set 1, more training data produces an overfitting model).

3.SVM: I used Weka's inbuilt SMO module that implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. Support Vector machines help classify data by constructing the best separating hyperplane for the data.

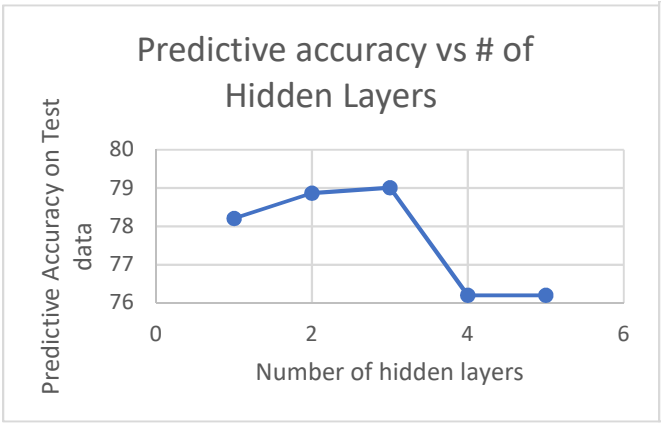
Learning Curves: Too much training data seems to generally produce an overfitting model for this algorithm as reflected in the decrease in the performance of the learning curve shown below. Overall, the LC differs for both datasets. Number

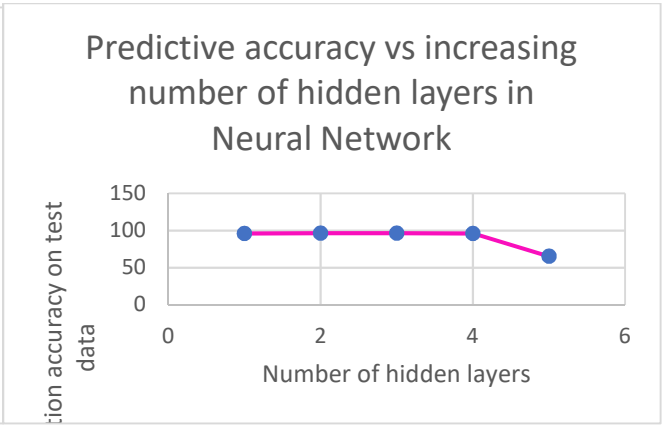
and type of kernels can be changed for this algorithm that may affect performance. Here, I use the Normalized poly-kernel. The regularization parameter can also be changed, which will control for overfitting of the model, especially when there is too much training data. I didn't test model performance on any of these parameters since I was unsure of the exact mechanism of kernel functioning (not covered in class thus far).



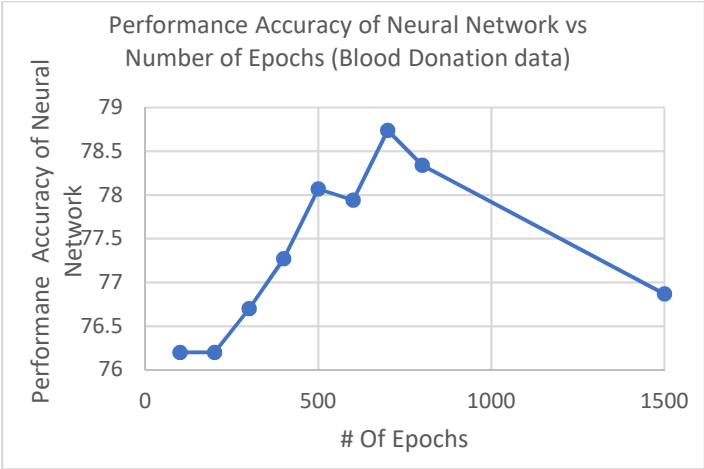


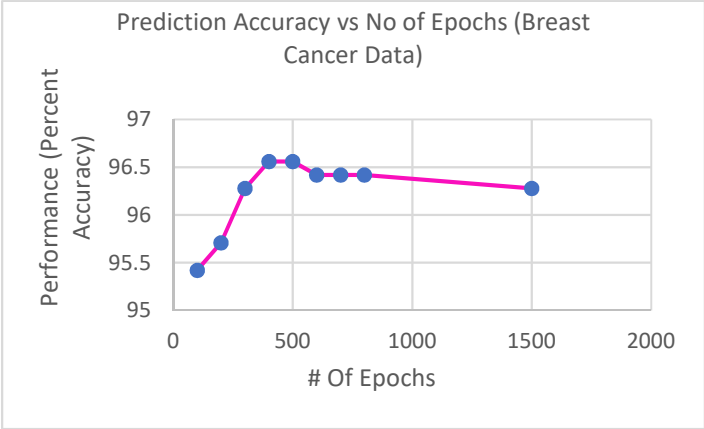
4. Neural Network: I used multilayer perceptron function in Weka for neural networks, and optimizes over the number of hidden layers which can affect model performance significantly. For the first dataset, performance peaked at 3 hidden layers of 3 nodes each and for the second dataset, at 3 hidden layers of 5 nodes each. As the number of hidden layers increases beyond, say the total number of attributes, model performance declined significantly, as shown below. This could be because the model could be overfitting data as the neural network complexity increases. Overall, of all algorithms used, neural networks seem to produce the best results on test data.



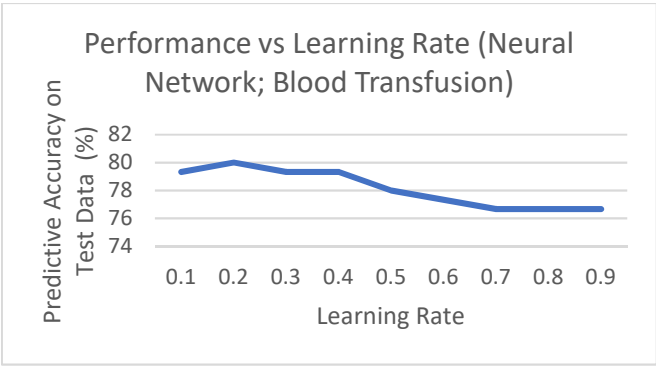


Epochs: As the number of epochs for training increases, the accuracy of the neural network increases to a maximum and then decreases with further increase in number of epochs. As the no. of epochs increases, the fit of the model first increases, and as the model is well-trained and begins to overfit the data, the performance accuracy decreases. This trend seems to be true in both datasets, as shown below.

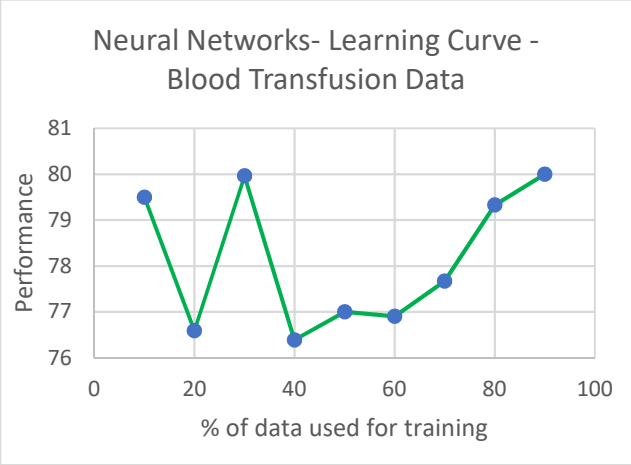


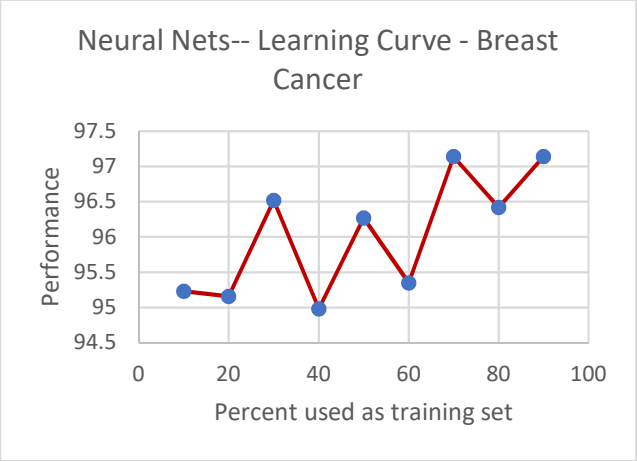


Learning rate : Below is the performance as a function of the learning rate parameter for the blood transfusion dataset. As learning rate increases, the predictive accuracy first increases then eventually decreases, since high learning rates could lead to the model overfitting data. Since neural networks, use the gradient descent algorithm and use the learning rate as a step size to reach the local minima, the size of this parameter is important. Too high a learning rate could lead to the model to skip over the optimal solution even after it finds it, and too low a learning right might lead to the neural network taking a lot of time in the gradient descent algorithm to eventually reach the local minima.



Learning Curves : As we can see below, increasing the amount of training data significantly improves performance of the neural network, steadily after 40% of training data in dataset 1, and in a steady fashion for dataset 2. This could be because larger amounts of training data allows the relationships between the nodes and the hidden layers to be refined to create a better fit for the classification model.

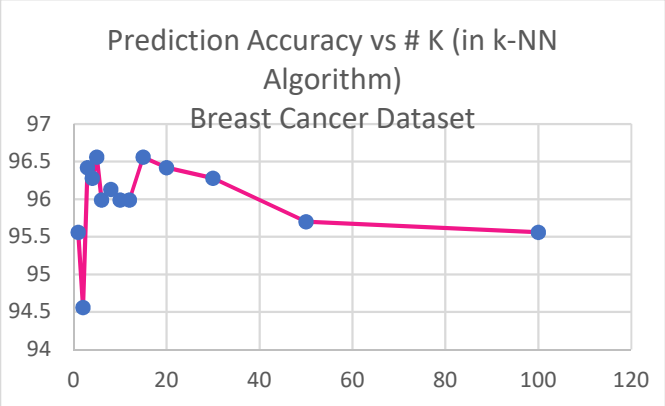


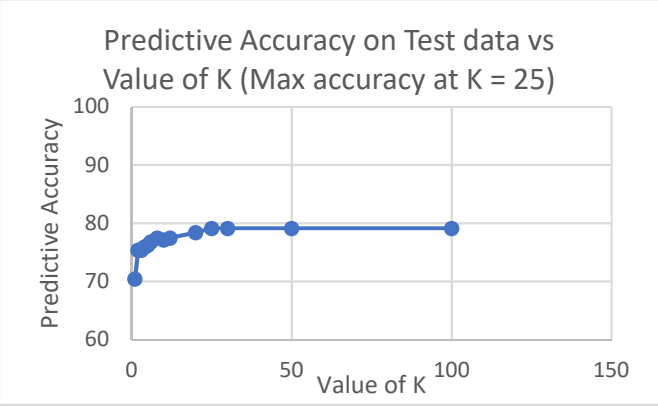


5.K-NN:

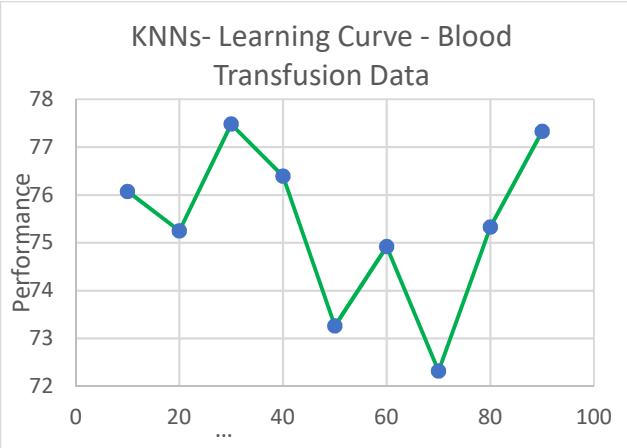
Number of Neighbors: I found that optimal number of neighbors is 6 by doing the analysis shown below.

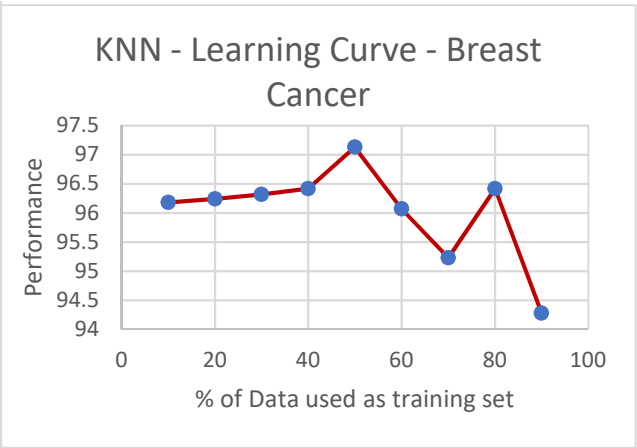
- Breast Cancer Data: As the no. of neighbors increases (in both datasets) the performance of the model first increases, reaches a peak at **K= 5,** and then decreases as the model begins to overfit on the training data and fails to generalize on test datasets.
- Blood Donation Dataset: As the no. of neighbors increases (in both datasets) the performance of the model first increases, reaches a peak at **K=25** and then increases only very slowly with further increase in number of neighbors.





Learning Curves: For K-NN is widely fluctuating with % training data. Specifically, we can see that performance peaks for a moderate amount of training data, just big enough to allow the model to generalize and fit the training data well, and just small enough to avoid overfitting.





IV. Conclusion:

Overall, all the models seem to have much higher accuracy with the breast cancer dataset than the blood transfusion dataset. The performance of the algorithms on training data as test dataset is obviously much higher than the models learning on new datasets. In this paper, the performance of the Neural networks and ADAboost algorithms were compared against the # of iterations and epochs. Similar analysis could not be performed on the performance of other algorithms due to parameter limitations in the software interface – Weka – that I used. We see differences in the way that different algorithms train and perform for both datasets due to intrinsic differences in the size, attributes, spread and outliers in both datasets. Further improvements in learning can be made by optimizing over more parameters, using more robust classification algorithms as function inputs into ADABoost, training over a larger input data size, and testing over newer datasets etc.

Citations

[1] Yeh, I-Cheng, Yang, King-Jang, and Ting, Tao-Ming, "Knowledge discovery on RFM model using Bernoulli sequence", Expert Systems with Applications, 2008.

[2] Dataset 1- http://tunedit.org/repo/UCI/breast-cancer.arff

[3] Dataset 2- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.486.6694&rep=rep1&type=pdf