Arthur Shim
ashim6
CS 4641

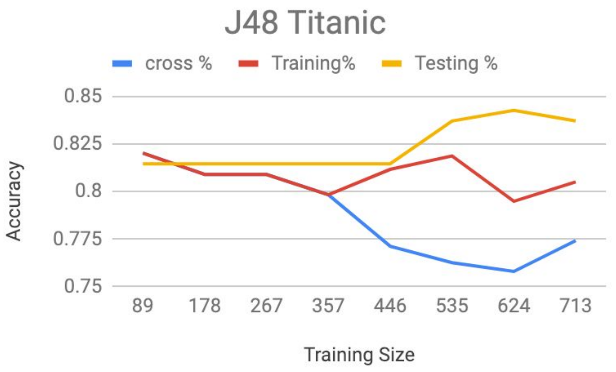
Analysis

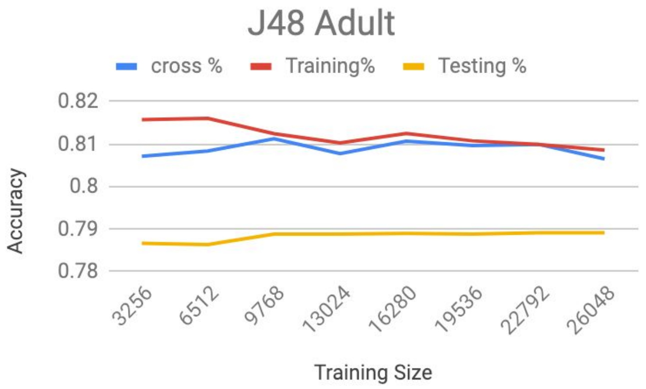
The two datasets I chose were the Titanic Dataset and the Adult Dataset. The Titanic Dataset holds information about passengers of the Titanic and can be used to solve the binary classification problem of whether the passengers survived or not. The dataset contains information like a passenger's family size, class status, and whether they had a cabin on the Titanic or not. It has 7 features and 891 instances. This dataset is interesting because it is composed solely of discrete features and is not an incredibly enormous dataset, so it will be intriguing to see how the different algorithms perform under these two conditions. The Adult Dataset contains information about adults around the world including age, education, residence, marriage-status, etc, and can be used to classify whether an adult has an income of above $50k. The Adult Dataset has 14 features and 48,842 instances, so it is a significantly larger dataset than the Titanic Dataset. It also contains a mixture of discrete and continuous features, so it'll be interesting to see how the learning algorithms perform on this dataset compared to the Titanic Dataset.

For each of the five learning algorithms, I assessed the performance of training, testing, and cross validation using different training sizes for each dataset. The performance of an algorithm was determined by its accuracy, i.e. the percentage of examples that it classified correctly. I used k-fold cross validation with k=10 for most of the algorithms, with the exception of the SVM algorithms for the Adult Dataset where I used k=5.

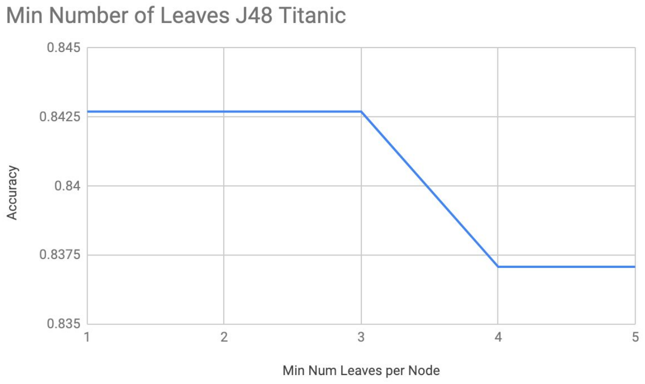
Decision Tree

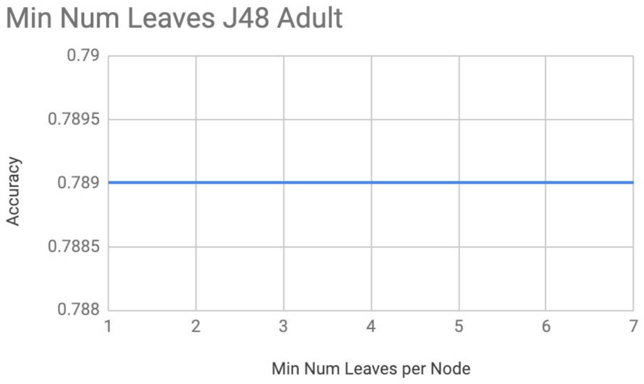
The decision tree algorithm I used is J48.





The performance of J48 on both datasets had similar trends. As the training size increases, the training accuracy decreases and the testing accuracy increases. This indicates that J48 overfits more with smaller training sizes, which is why its training accuracy is higher and its testing accuracy is lower for smaller training sizes. It is interesting to note that for the Adult dataset, increasing the training size does not dramatically increase the testing performance, leading me to believe that J48 does not greatly benefit from having more data. A more noticeable increase in the testing accuracy can be seen in the Titanic Dataset as the training size increases, however it is still only a slight improvement.

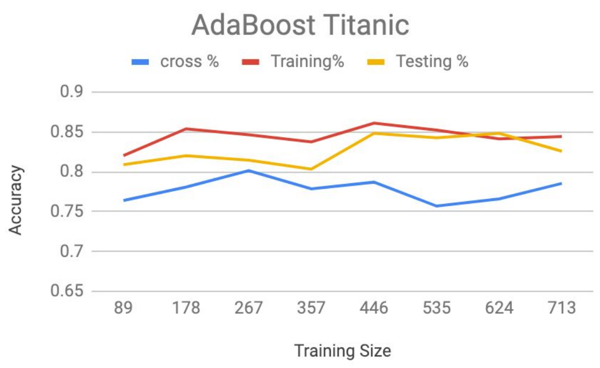


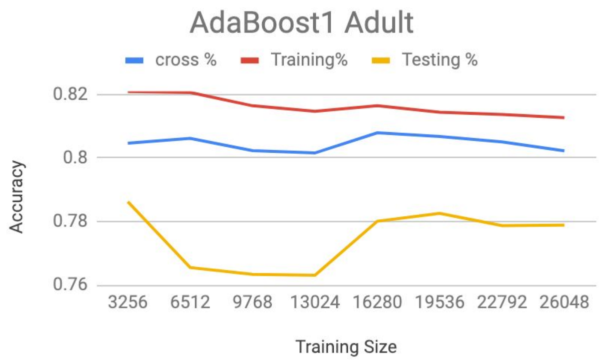


I measured the performance of J48 on both datasets using different minimum number of leaves per node as a form of pruning to create a more optimal decision tree to train on. For the Titanic Dataset, I found that using more than 3 minimum leaves per node decreased the accuracy of the tree, and for the Adult Dataset, there was no difference in performance with respect to the number of minimum leaves. Through more testing, I found that J48 performed best with a confidence factor of 0.3 on the Titanic Dataset and a confidence factor of 0.25 on the Adult Dataset. Since the Adult Dataset had a slightly smaller confidence factor, this indicates that J48 has slightly less faith in the data when it comes to the Adult Dataset than the Titanic Dataset. These results were used to tune the trees that resulted in the learning curves previously shown.

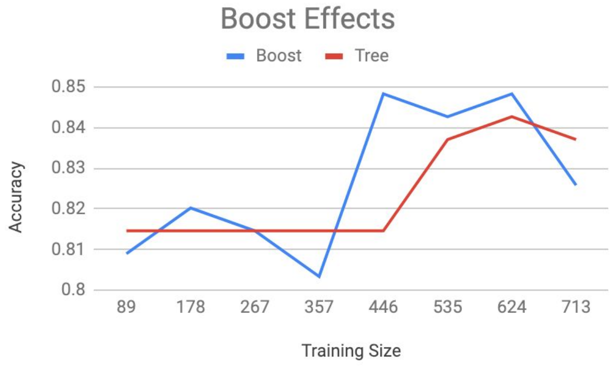
Boosting

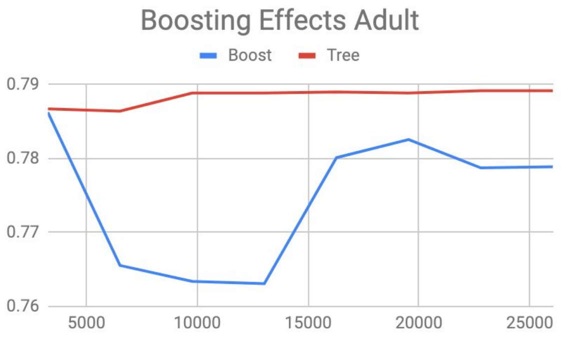
For boosting, I used the AdaBoost to boost J48. AdaBoost puts a greater weight on instances misclassified by a classifier, and is sensitive to noise and outliers.





For the Titanic Dataset, AdaBoost led to higher training and testing accuracy as training size increased. Both showed generally increasing trends, with less variation between training and testing accuracy for higher training sizes in the boosted version. For the Adult Dataset, AdaBoost's training accuracy was a little higher than the non-boosted version, maintaining the same downward trend as training size increases. AdaBoost's testing accuracy is noticeably lower than the non-boosted version. This is likely due to AdaBoost being sensitive towards noisy data. The Adult Dataset contains some instances with missing or unreported data. This likely is what caused AdaBoost to perform poorly in testing with the Adult Dataset.

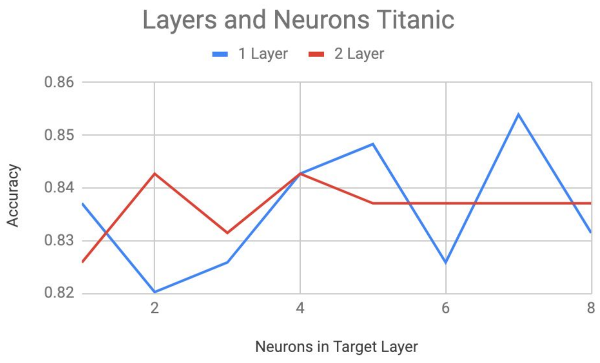


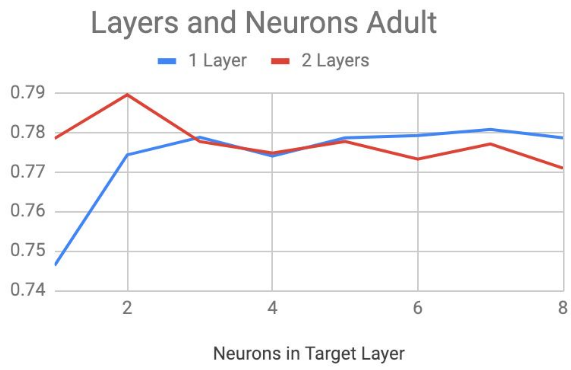


These graphs further illustrate the trend seen before. With the noiseless Titanic Dataset, AdaBoost generally has higher accuracy than the non-boosted J48. However with the noisy Adult Dataset, AdaBoost has a lower accuracy than the non-bosoted J48.

Neural Network

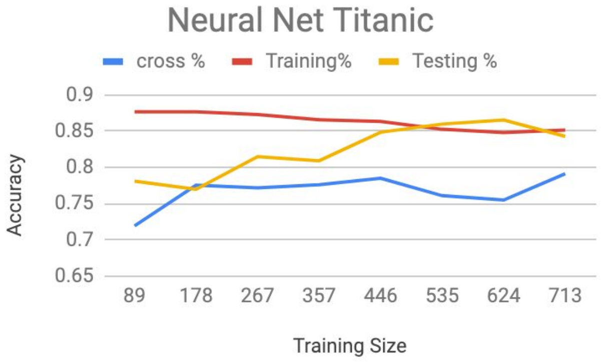
Initially the neural network used was tuned by comparing the performance using different numbers of neurons and layers. The results for each dataset are seen here:

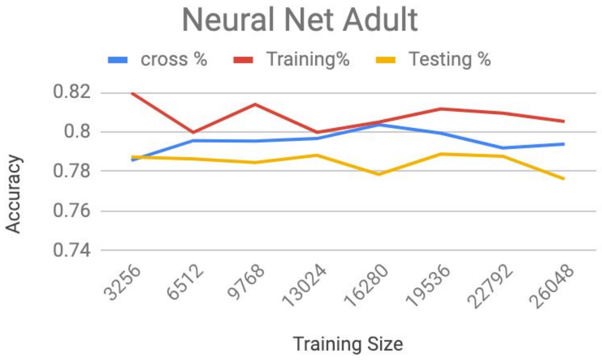




For the Titanic Dataset, the neural network with the best performance was a one layer network with 7 neurons in the target layer. For the Adult Dataset, it was a two layer network with 2 neurons in the target layer. Data for networks with three or more layers are not being shown because they were found to all have poorer performances than one and two layer networks.

I also tested the performance of different combinations of learning rates and momentum values for neural networks and found the highest accuracy for the Titanic Dataset was a network with a learning rate of 0.3 and a momentum value of 0.2. For the Adult Dataset, the network had the best performance with a learning rate of 0.5 and a momentum value of 0.3.

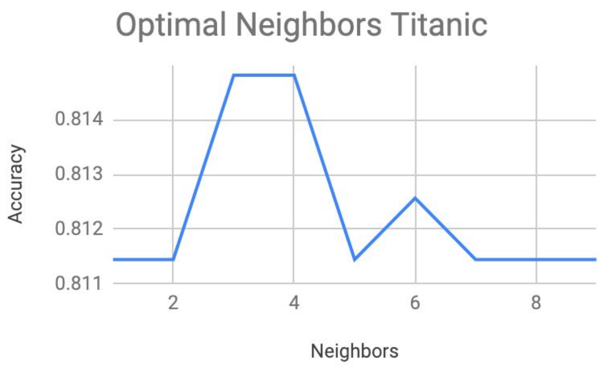


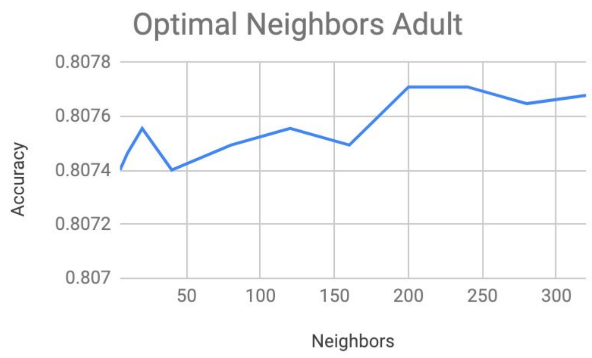


Using these acquired parameters, I found the performance of training, testing and 10-fold cross validation with respect to training size on the corresponding networks: Like the decision tree, both datasets displayed decreasing training accuracy as training size increased. This indicates a similar pattern of more overfitting with smaller amounts of data. The Titanic Dataset demonstrated increasing testing accuracy with larger training sizes, meaning the network performed better in terms of classifying data when more training examples were present. The Adult Dataset, however, demonstrated a slight decrease in the testing accuracy as training size increased. I took this as an indication that, although neural networks generally benefit from having more data, the Adult dataset had exceeded the network's optimal training size so any additional data decreased the network's performance. There were also instances in the Adult Dataset with incomplete data, such as certain features missing or unreported. Due to this, increased training size could have included more incomplete instances which decreased the overall performance of the network.

KNN

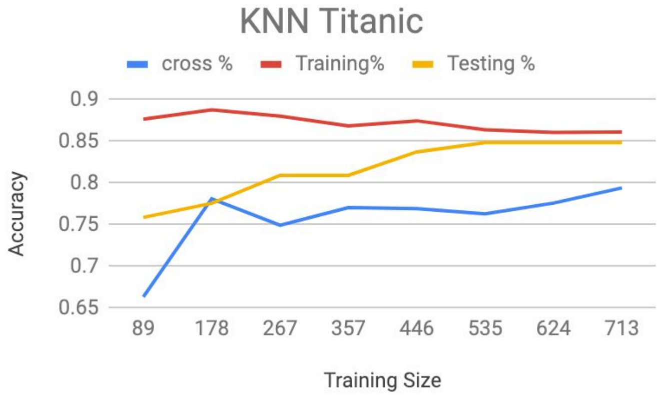
For KNN, the first step was finding the optimal number of nearest neighbors. I tested the accuracy of the algorithm using different values of k to get the following results:

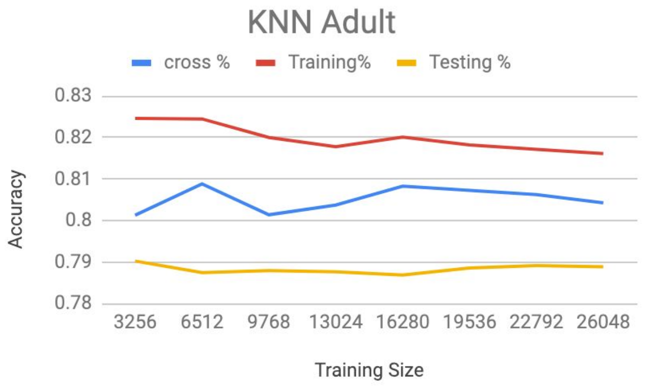




The optimal value for k for the Titanic Dataset is k=3 or k=4 and the optimal value for the Adult Dataset is k=320. For the remaining tests, I used k=3 for the Titanic Dataset.

I also weighted each example using 1/distance so that farther examples had less impact. For the nearest neighbor search algorithm, I used a cover tree.





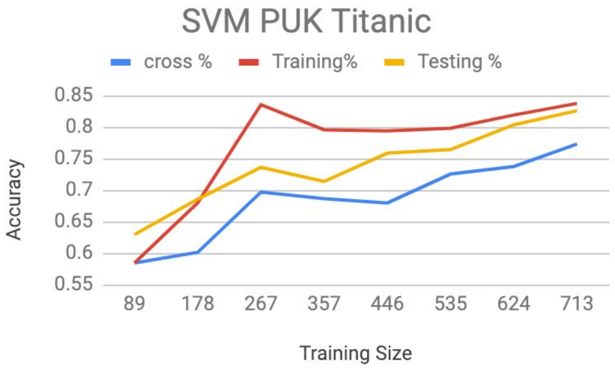
Similarly to the decision tree and neural network, as training size increases the training accuracy decreases indicating that the same trend of decreased overfitting holds for KNN. For the Titanic Dataset, the testing accuracy increased with training size meaning KNN's ability to correctly classify instances improved as more training data was provided. Like the neural network, KNN also demonstrated a slight decrease in testing accuracy as training size increased for the Adult Dataset. This could be attributed to the incomplete instances and missing data in the Adult Dataset, where adding more training data with incomplete instances groups certain examples together that should otherwise not be together.

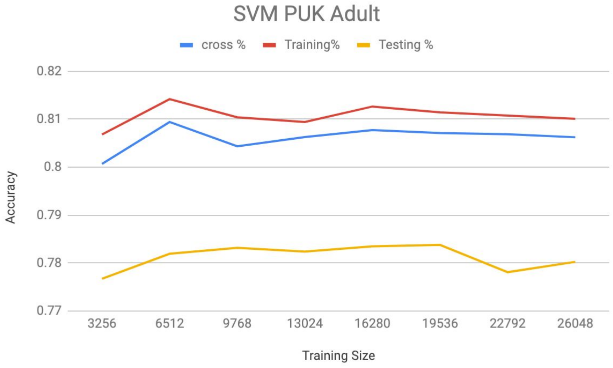
SVM

The two SVM kernels I used were Puk and NormalizedPolyKernel. This was after testing multiple kernels on both datasets and finding that these two kernels had the best performance.

SVM (Puk)

For the Puk kernel, I tested different filter types and values of c to find the combination that had the greatest performance. I found that the normalized filter type and c=0.25 were the highest performing values. 0.25 is a fairly small value for c, meaning the SVM didn't prioritize having the data points very close to the support vectors.

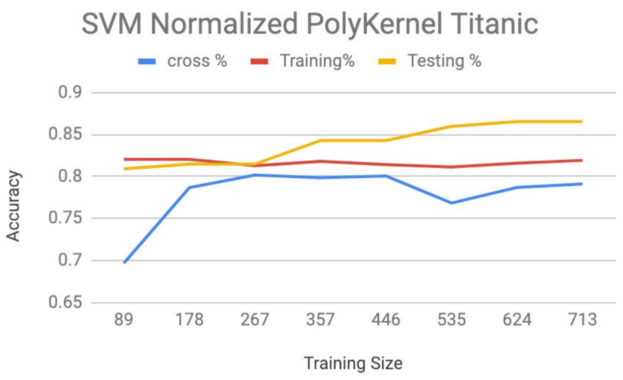


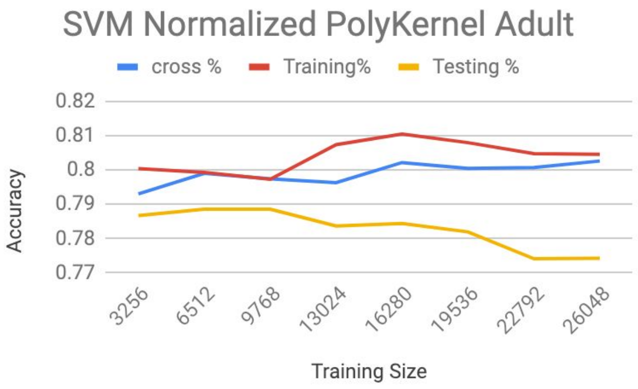


Unlike the previous algorithms, SVM Puk had increasing training accuracy as training size increased for the Titanic Dataset. This is consistent with the fact that SVMs increase in accuracy as more data is presented. Similarly, the testing accuracy increased as training size grew for the Titanic Dataset. However, for the Adult Dataset training accuracy did not show an upward trend with training size, and testing accuracy showed growth for increasing sizes of training data up to a point before dipping back down again. This contrast to the Titanic Dataset could once again be attributed to the incomplete data existing within the Adult Dataset. These results indicate that as more training examples were added, the addition of incomplete instances lowered the overall performance of the SVM by skewing the support vectors, making them less accurate.

SVM (NormalizedPolyKernel)

For the NormalizedPolyKernel, I found that the normalized filter and c=0.5 had the greatest performance. The higher c value indicates that the NormalizedPolyKernel puts a higher priority on having the data points closer to the support vectors than Puk. NormalizedPolyKernel normalizes its input by dividing it by its magnitude.





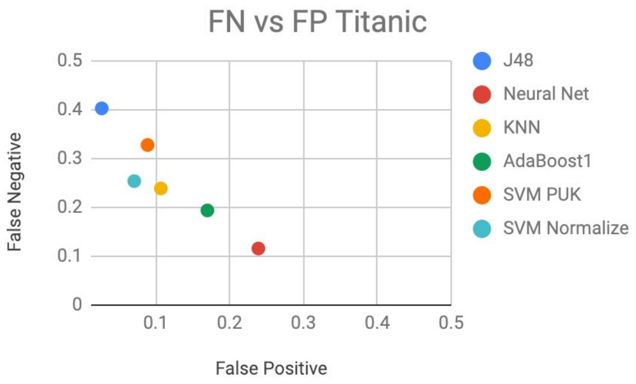
The NormalizedPolyKernel showed a more stagnant training accuracy than Puk for the Titanic Dataset. However, like Puk, its testing accuracy increased as training size increased. For every training size, NormalizedPolyKernel has a higher testing accuracy than Puk for the Titanic Dataset indicating it is a much better kernel to use to classify the Titanic Dataset.

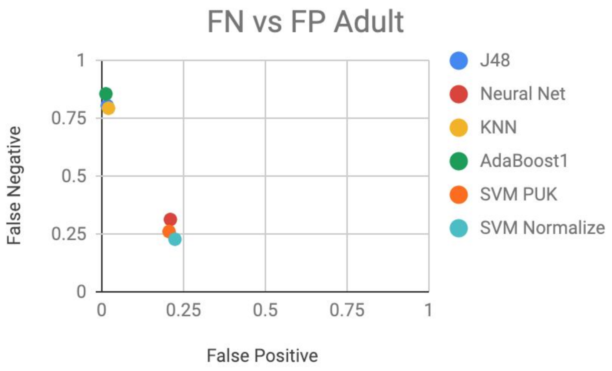
In terms of the Adult Dataset, NormalizedPolyKernel had a slight upward trend in the training data and a dramatic downward trend as training size increased. Its testing accuracy began higher than Puk's for smaller training sizes but then became lower than Puk's for larger training sizes. I took this to mean that NormalizedPolyKernel was more severely affected by the missing data in the Adult Dataset, so as more incomplete instances were added, the worse the SVM performed. This indicates that NormalizedPolyKernel is much more sensitive to noise than Puk, since it performed better than Puk on the Titanic Dataset which had no noise, but performed worse on the Adult Dataset which had incomplete data.

Conclusion

Overall, for the Titanic Dataset, SVM NormalizedPolyKernel had the best performance in terms of testing accuracy. For the Adult Dataset, SVM Puk had the overall best performance for all training sizes. It is likely that if the Adult Dataset had less noise and missing data, SVM NormalizedPolyKernel would also have had the best performance for the Adult Dataset (or at least not the worst). Likewise, if the Titanic Dataset had more noise, then SVM NormalizedPolyKernel would likely not be have the highest testing accuracy.

The testing accuracy is not the only measure that can be used to determine the effectiveness of each algorithm. The false positive and false negative rates can also be used. An algorithm with a high false positive rate indicates that a high portion of its true classifications are false. Likewise, an algorithm with a high false negative rate indicates that a high portion of its false classifications are true.





For the Titanic Dataset, J48 has the lowest false positive. So if most of the passengers on the Titanic survived with only a few dying, then J48 could be considered the best performing algorithm for this dataset. If most of the passengers on the Titanic didn't survive, then we could consider neural networks, with the lowest false negative rate, to be the optimal algorithm for this classification problem. If we expect an equal number of survivors and nonsurvivors, it's possible we desire an algorithm with the minimal false positive and false negative rates. In this case, that would be KNN. In the case of the Titanic, since the majority of passengers died, it could be fair to say that neural networks are the optimal algorithm for this problem.

For the Adult Dataset, the same line of thinking could be applied to the problem of whether to prioritize algorithms that are very accurate in their true classifications, very accurate in their false classifications, or both. In this case, if there are many more people with incomes above 50k then it could be said that SVM Puk is the optimal algorithm since it has the lowest false positive rate. If most adults have incomes less than 50k, then SVM NormalizedPolyKernel could be said to be the optimal since it has the lowest false negative rate. Both SVM Puk and SVM NormalizedPolyKernel have the lowest false positive and false negative rates of any of the algorithms, so if both are important to consider then either algorithm could be argued to be the optimal in terms of solving this problem.

Acknowledgements

All algorithms were run using Weka.

My datasets came from the following links:

Titanic:
https://www.kaggle.com/dmilla/introduction-to-decision-trees-titanic-dataset?fbclid=IwAR2wDL2sOlU1vRf4QY8h-FmWbXe0fb_S_dKrLDzzPrGeOJVjg4LvANazzLc

Adult:
http://archive.ics.uci.edu/ml/datasets/Adult?fbclid=IwAR0lRjGu57NmSSrwT2F4UqH-9hyvtJrq9yGdzoUo8QsawnKgo7660AqD5Uw