David Li

CS 4641, C. Isbell

10 February 2019

Supervised Learning

Overview: The following is an analysis for the results of different surprised learning algorithms and their performances and trends based hyperparameters, data input, and the algorithms themselves. The algorithms in question are J48 for Decision Trees, MultilayerPerceptron for Neural Network, IBK for KNN, AdaBoostM1 for Boosting, and SVM with kernels of PUK and Normalize PolyKernel. The data are as stated below.

Data:

1. **Adult:** The data looks at over 30,000 adults during the 1990s and based on their background information, such as age, working class, education, gender, and native country, tries to predict whether an individual will be able to have an income of over %50k/year. 15 different attributes are looked at during this experiment, but many were removed personally for this analysis. FNLWGT, capital- gain, and captain- loss were removed due to a lot of missing data and a fundamental lacking in their understand; we also wanted to make the attributes be more based on their current situation more than some calculated weight, which FNLWGT takes into account many the attributes in consideration already and double weighing some things. Manually, many categories were discretized, such as marriage status, native- region, and education level.

   This data is interesting as at the surface, it seems that KNN and SVM would excel at predicting this because similar people especially in education background or work class would have similar pay. But then noise becomes a huge factor as gender or racial roles might a huge factor or act as noise in this data, as well as life events such as family size and marriage which may conflict or lead to very different results.

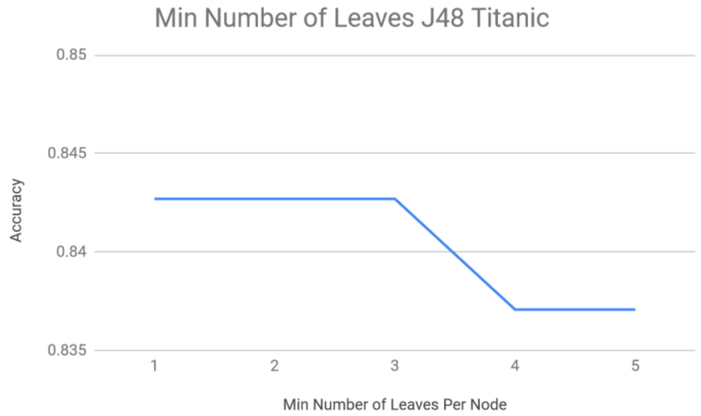
   The data was taken from http://archive.ics.uci.edu, exact file in README.txt
2. **Titanic:** This data looks at the majority of the passengers during the Titanic tragedy and based on whether attributes such as if they are alone, family size, gender, and indirectly, income, can be used to predict their survival. 12 attributes were given; I removed ID for lack of input, where they boarded the ship or embarked from which did not matter, and number of siblings and parents as that can be covered through size of family and if they are alone.

   The titanic data set is interesting as it looks at a very dividing factor, family size. Some people benefit from surviving by having family members help them, but on the flip side, those who help or those with families can be hindered while taking care of others. This brings an unclassified set of noise into the data, and seeing how well the algorithms can handle it will be interesting. This also provides a smaller dataset to juxtapose against the adult data set, by about 40 times difference in size.

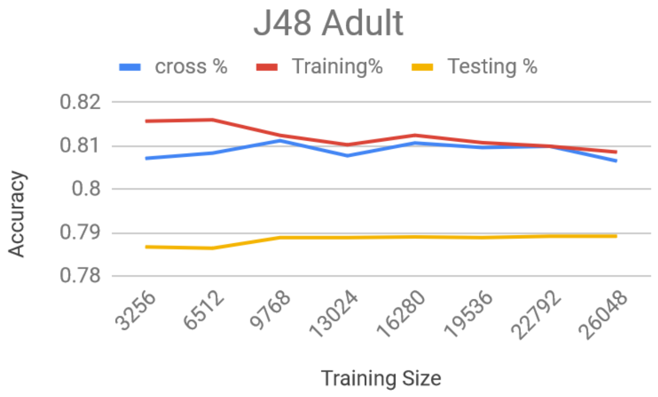
   The data was taken from a Kaggle competition, link given in the README.txt

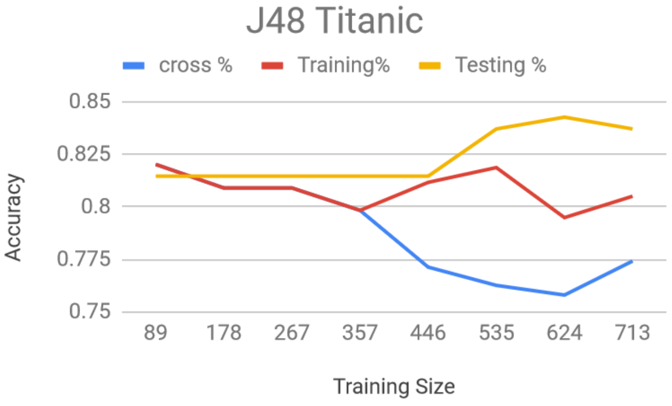
Decision Trees

The algorithm chosen is J48, which has additional features over a basic tree such as accounting for missing values, decision trees pruning, continuous value ranges, derivation of rules and so on. Since we needed a tree with pruning and many of our data set had missing values (dead man tells no tales from the Titanic, and some adults omitted information) this tree covered much of our bases. For both data set, as we turned hyper parameters using cross validation, one of the biggest ones we looked at was the minimum amount of leaves per node and how it affects accuracy.



This became a factor during Titanic as there were some attributes had very little few values especially after discretization and pushing the minimum leaves would end up discounting them altogether. This was not as big of an issue for adult. The other parameters looked at was confidence factor and the default was used after testing it using cross validation.





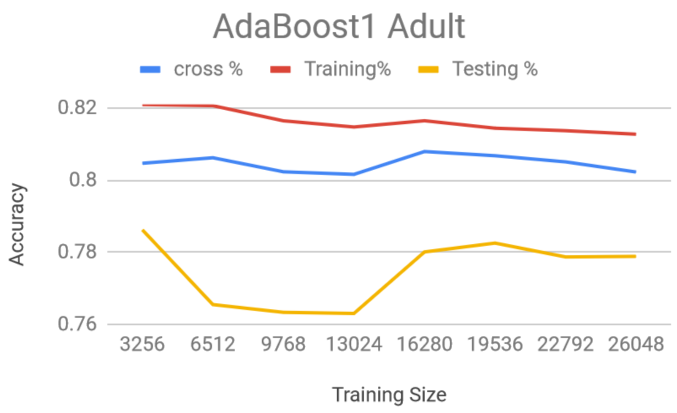
For reference, each of the number on the bottom represents a 10% increment of the data training size, from 10 to 80%.

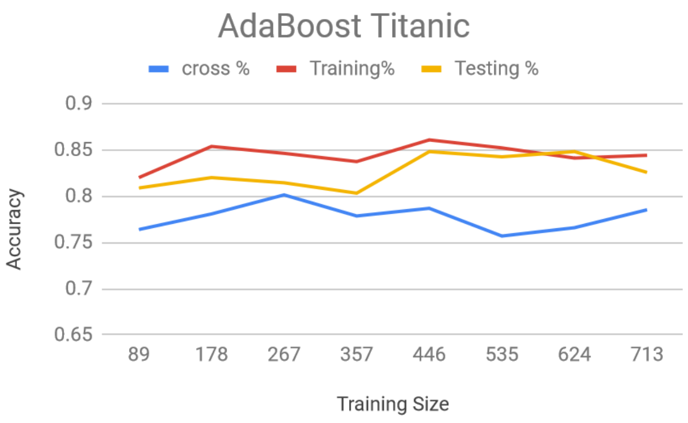
In the J48 graphs, we can see that over time, the training accuracy was decreasing while testing accuracy increased, this occurs more slowly in the adult data size with testing increasing at a much slower rate eventually testing should cross with training as training continues to drop near. In Titanic, the testing surpassed the training at 20%, and continue to increase while training overall decreased or stayed the same. This makes sense as at lower input data, a smaller training data is easy to over fit and predict itself more accurately, as demonstrated by both training and cross validation, while the small data sets may not be representative of the entire data set nor the testing data. At higher percentages the training data becomes more representative of the whole entire data set and thus can predict the testing data more accurately, but the training set loses the edge of overfitting.

The effects of missing the missing data is not to be ignored and addressed in boosting of the J48 tree.

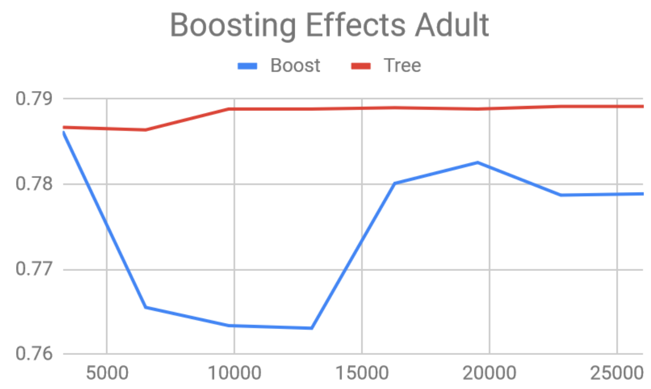
Boosting

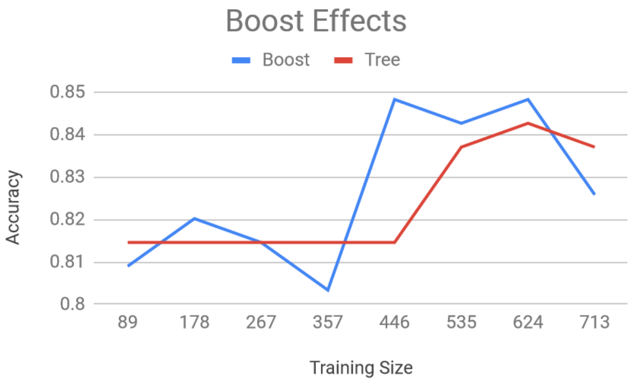
AdaBoost M1 was used as the algorithm for boosting the J48, which has built in pruning already as we can pick the settings for it in Weka. Weka offered little tuning for hyper parameters and so for the sake of consistency, we used the same hyper parameters for the J48 algorithm.





As displayed here, one of the common trends for training is that as data size increases, the accuracy decreases as overfitting becomes less of an issue. In adult, however, the testing had a really large drop , or a really high start on the smallest data set, which may be the size of the tree and leaves that was affected by the smaller dataset, despite the set being randomized a 10% sample maybe not be normally distributed relative to the entire set, especially to the overwhelming number of Canadians and Americas in the adult data set skewing everything. Titanic the data was much more consistent, a trend we will continue to see, possibly due to the large amount of noise from the adult set.



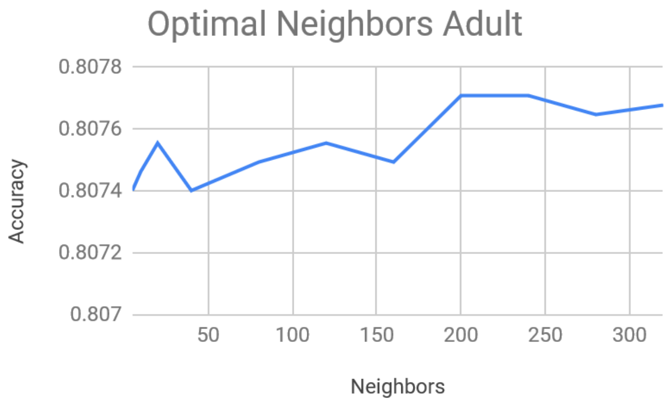


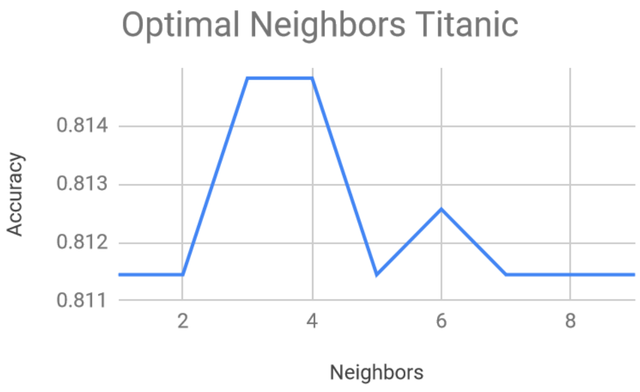
Here are graphs comparing J48's result with and without boosting. In the adult data set, boosting overall lowered the results of the prediction, while in Titanic, it brought on average a higher accuracy but higher variance. The huge drop in adult might be due to AdaBoost M1's high sensitivity to noise, which is very prevalent in the adult data as discussed in the data introduction slide. In the Titanic testing set, AdaBoost M1 targets the incorrect samples from the before and focuses on them, and as we increase size of training set, it is possible newer unknowns are introduced since it is such a small training sample that each 10% is only around 70 instances, thus large amounts of variance are brought in.

Another source of noise for adult is the large amounts of missing data, which can be very deceive when two things are similar in a tree, but the missing data might make a huge distinction between the two. When talking about boosting it may add noise but when only running trees, J48 has mechanisms to account for missing values.

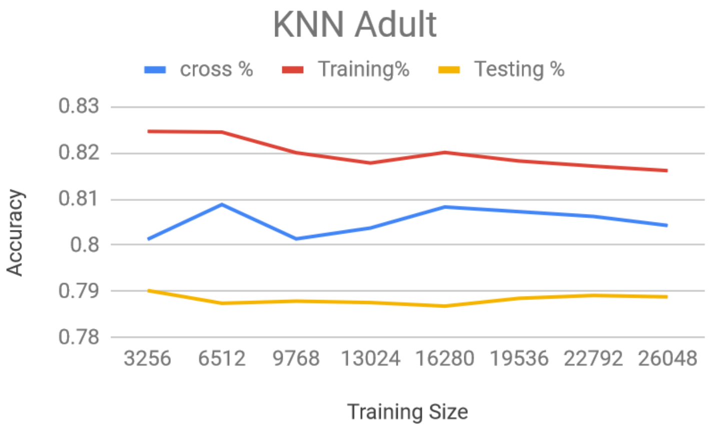
KNN

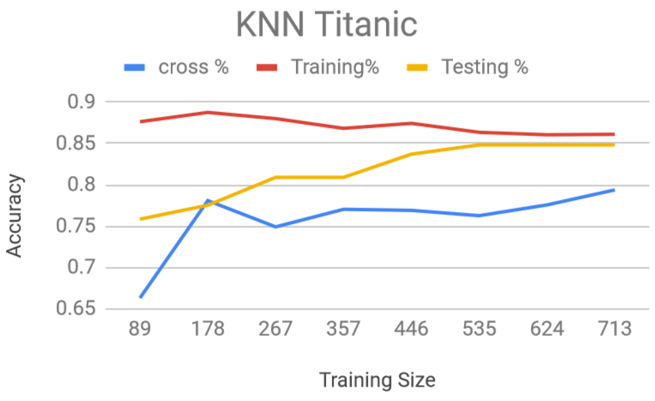
For KNN we chose IBK in the Weka library. As far as we can tell it runs like any other KNN algorithm. During cross validation tuning for our hyperparameters, some basic things we decided was using cover search for faster finding of neighbors, and using distance weighting of 1/ distance, which was important as many data near a certain point maybe be similar, but if we take a lot of neighbors and the radius increases, it is imperative that the further instances are weighted less the further they are. Lastly was the number of neighbors.





Running cross validation with different neighbors, despite adults slow rising, the difference is miniscule at about .001 per 50 neighbors, capping out at around 200, which became our neighbors for adults. For titanic, 3 or 4 seemed to fit the data the best and was chosen as our hyper parameter. The increments of K to test was decided by me to use only up to 5 % of the testing data, which is 20% of the over all data, as I did not want every single point of the testing data to be considered.



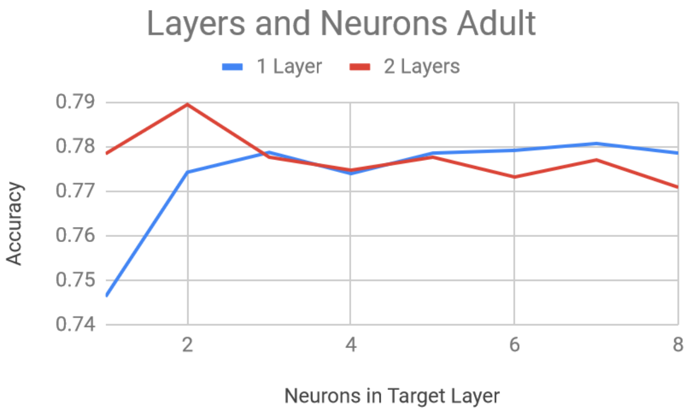


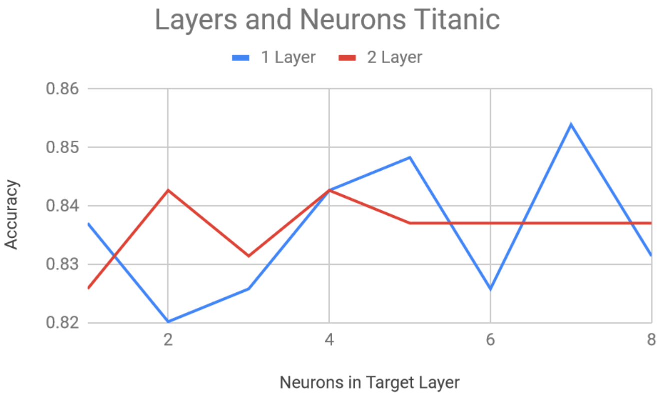
For the adult training data, increasing the training size did not affect it greatly, this may be because of the overwhelming number of similar data points due to the large data set and the large number of attributes that the nearest neighbors regardless of 10% or 80% of the data gave the same results. With a large K to data size ratio for Titanic, the increase in testing result was more evident, as running 4 neighbors as the training size increased was enough to give a better representation of the 178 testing instances. As per usual training accuracy fell as data increased most likely due to over fitting, and eventually, the testing curve will cross with the training curve, especially evident in the Titanic data.

Another thing to note is that due to the large amounts of missing data for adult, many unrelated instances may be close by each other due to algorithms placing same value for missing values of an attribute, bringing different adults of different backgrounds closer as neighbors.

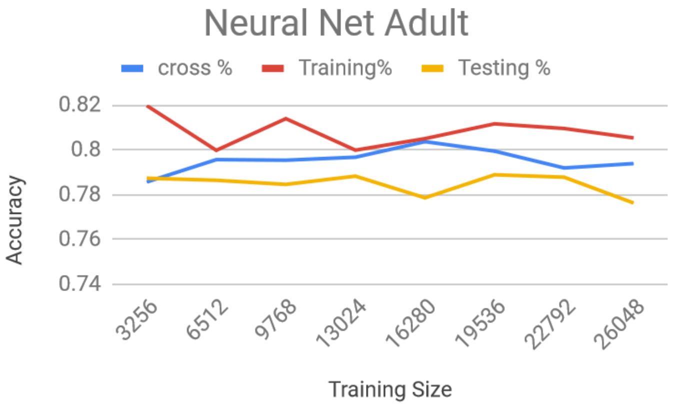
Neural Network

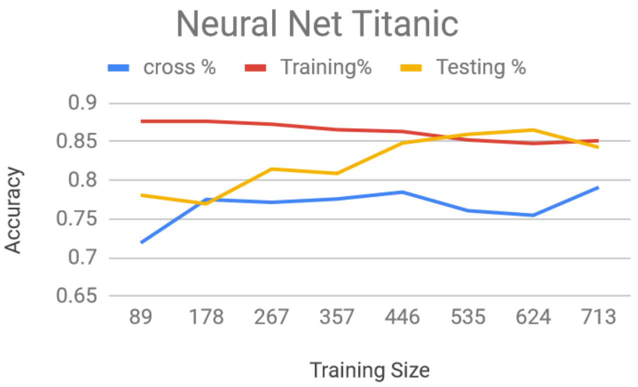
We used multilayer perceptron, or MLP, for neural networks. The idea of MLP is that given enough neurons per layer, anything can be modeled using two layers. With more layers it can model more complex data, but simple data will be overfitted. During tuning, found the best learning rate of .5 and momentum of .2, both of which were higher than the default values, allowing for our data to have a bigger affect on the model as it is fed newer features, especially given large amounts of attribute.





For layers, we first cross validated testing the first layer, from 1 to attributes / 2, and then took the best one as the first layer and ran 1 to attributes / 2 for the second layer. With adults being a more complex data with more attributes and instances, two layers with 7 and 2 gave the optimal results, while a simple Titanic only required a single later of 7, which is a, attributes / 2, a default value.





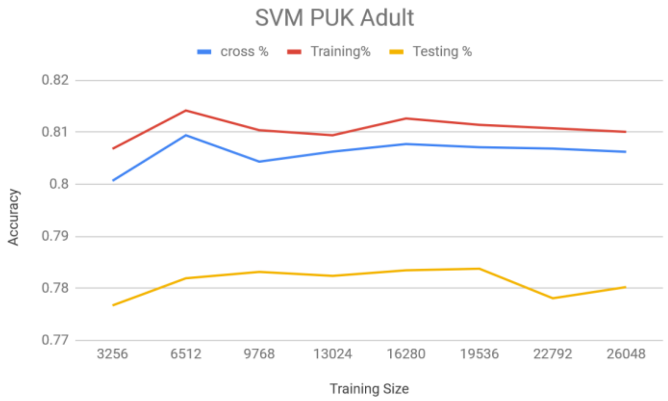
A trend appearing especially at this point and going onwards is the adult data set becomes less accurate for testing. This may be because for neural nets, we had too high of a learning rate and thus anything new added towards it affects the prediction in each model. At 80% used as training data, the neural net for adult takes an especially hard hit, probably du to all the different attributes being used compared to 10%, many of them are not as defining for the overall prediction as the algorithm might have expected. For Titanic, still going strong, at about 50% the testing surpasses the training, which has started to fall from a high start due to overfitting. Titanic's neural had a huge increase as sample size increased, attributed to the single layer of hidden layers used that does not overfit a much simpler data set.

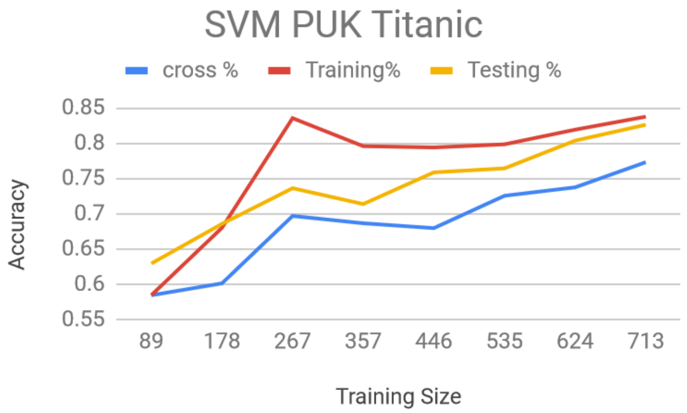
SVM

When deciding parameters, we once again tested multiple values using cross validation. But due to the large amounts of time to run each of the SVM kernels multiple folds, we had to run them 5 folds with limited hyperparameter testing, but each one is tested to some degree. We decided on a tolerance parameter of .1, which is much higher than the default, forcing the program to keep searching for a maximum until that point is reached. C is .25 for the Titanic set for both kernels while .5 for Adult, as it is harder to draw a distinction and create a larger gap of separation for the adult set, likely because of how close some attributes might be while still yielding different results. Epsilon is maintained and little punishment is given to wrong answers as this also yielded the best results during testing, although many of the epsilon values made little to no difference.

Puk

Not much is known found about Puk in both Weka and online, but it yielded some of the better results during cross validation. The hyper parameters for both SVM are already discussed above.





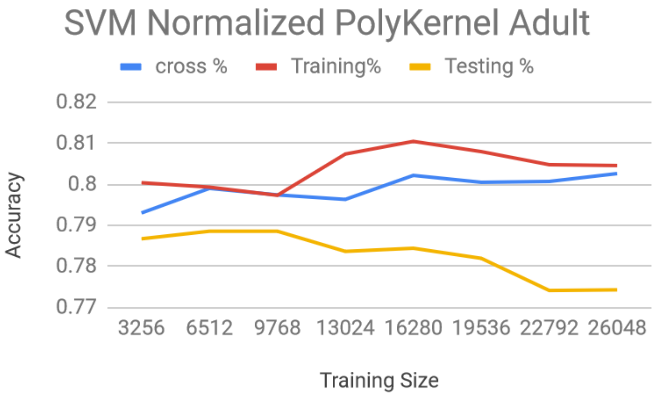
The adult data set had a much lower testing accuracy and it did not increase much or change on average except for the boost at 20% and 70% which were offset. This may be due to the large amounts of missing attributes and attributes in general, as well as a low punishing rate. These two combined made it extremely difficult to separate the instances into separate groups, especially at 70% and 80 of the data. The missing values acts as a bridge connecting the above and below $50k income individuals making it very hard to draw the support vector.

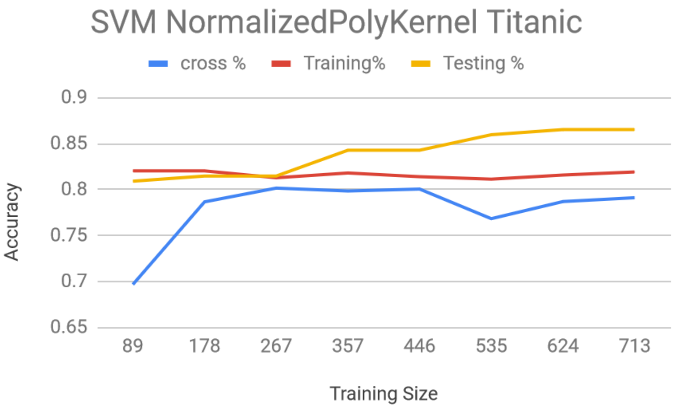
For Titanic, the training as well as the testing accuracy increased as time wore on, and towards the end they will cross each other's learning curve. Titanic has fewer missing data and more clear distinctions which may have made the separation easier to make using traits like is alone, and cabin location.

Normalized PolyKernel

Each instances of the data put through this kernel is normalized using the instance value divided by the magnitude of the instance making it fall between -1 and 1. The parameters are discussed above and the two kernels required very little difference in parameters, but more so based on the data set and size used.

A common theme amongst adult, the training can overfit on itself and consistently have a higher accuracy than testing. While testing keeps on dropping as more and more missing values are introduced into the data set and less of a distinction can be made. By normalizing it, rather than run puk, the testing overall became lower. The normalization process might have given weight to the missing values in negative ways that not only did not contribute but heard the algorithm, and this algorithm does not specify in Weka it processes missing values like J48 did. Also normalization may end up grouping and reducing the distance as it is normalized by magnitude between -1 and 1, and since it is divided by its own magnitude it becomes more like a direction vector, many of which can just be 0s due to missing values and in the multi-dimensional plane where they should have been going on in all N directions it only went off in N/2 and a bunch of data points all share the same plane for one or more attribute.

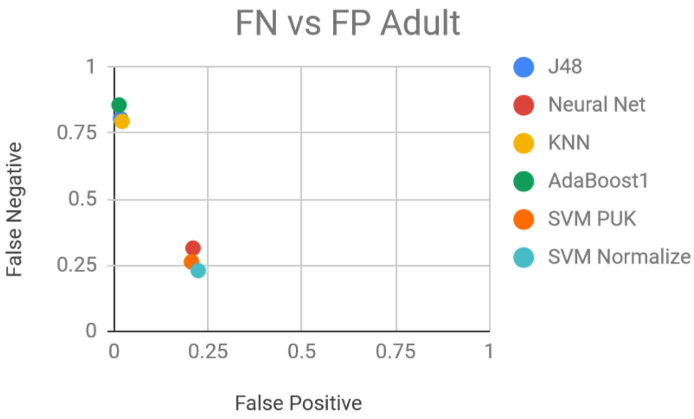


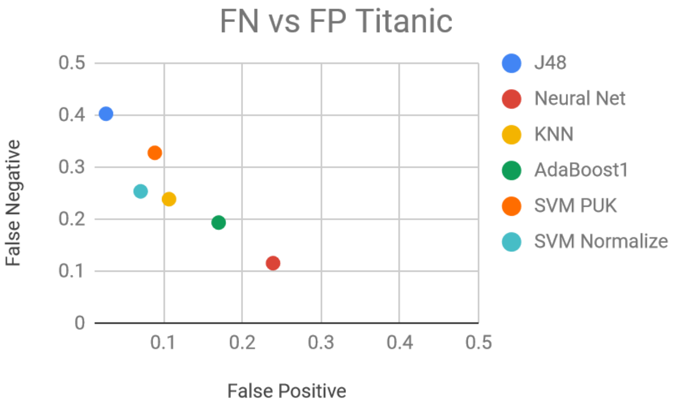


With Titanic, the crossing between training and testing occurs relatively early at 30%, and testing goes onto be one of the best predictors amongst all algorithms. The normalization may have helped in this case as they all become direction vectors but if each vector has a value it will not overlap in plan with other vectors when it shouldn't and really create distinction between the instances.

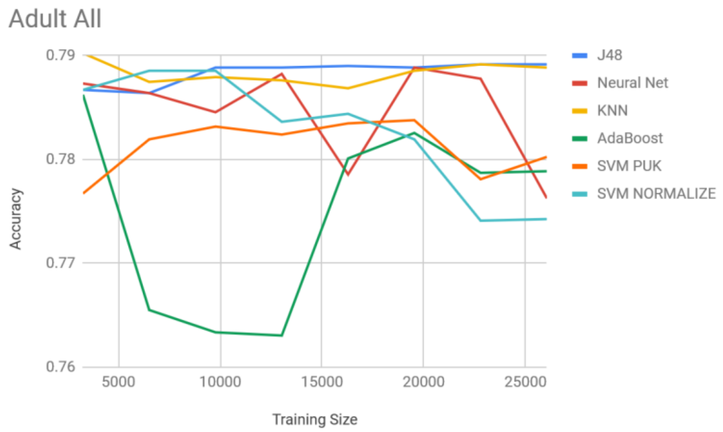
Analysis

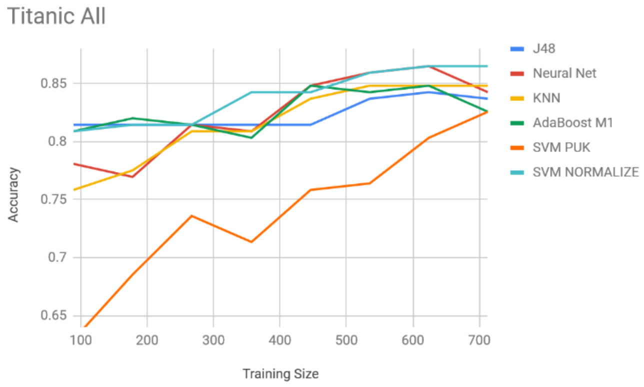
FP vs FN





As seen in each of these graphs, J48 had high false negative for both data set, while in Titanic boosting was able to have the closest and lowest scores in both false categories, while it remained high in adults. This is most likely due to the vast amount of data in adults and the large number of missing values that created noise for boosting, where Titanic was able to improve its J48 tree drastically, it prevented boosting from working correctly in adults. Another trend over all is that neural net had relatively similar FP vs FN rates and are relatively low giving it a relatively high F1 score. Neural net was able to distinguish between close or similar values than the rest due to its forward and backward propagation.



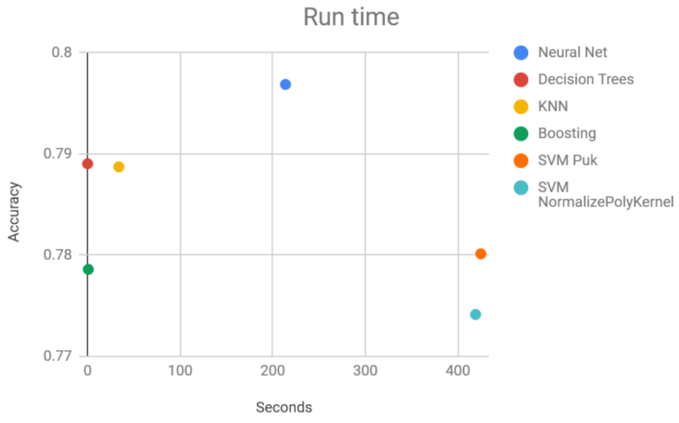


Within Titanic, SVM normalized performed the best while SVM puk improved the best as data input increased, as more data is taken in Titanic's data allowed for better separation helping both the SVMs. For both the SVMs and KNN, which also did well, demonstrating that Titanic having a full data discretized correctly allowed instances to be group and distinguished effectively.

With Adult, both SVMs performed terribly while KNN also did not improve with increased data input. The missing values hindered these algorithms.

Out side of that, Neural net performed terribly towards the end relative to everything else, most likely due to overfitting from the double layer multiple neurons per layer.

A key thing worth mentioning regarding the adult data set is my personal discretization might have been biased and aggressive. The discretization for the education portion, for example, had many grade levels, which I placed into those who didn't finish high school. Associate school, college, bachelor was all placed under college, and masters, prof school, and PHD into higher education. Occupation was entirely deleted for having too many different categories that were too specific. Marriage and work class was also discretized in a similar fashion as education. This discretization might have hidden some valuable data and taken away distinctions that were already blurred due to missing data, both acted as huge hindrances as to why the adult data had poor performances in general and hurt certain algorithms so much.



Lastly, given a time constraint, Decision Tree and KNN were able to get relatively good results in short time, while Neural Net performed the best while taking the medium time, so given unlimited time neural net is preferred. SVM in general did poorly time wise and performance wise. This is done by adding the run time for both data sets. Svm is heavily affected by the size of the data as well, more than any other algorithm.

Conclusion

Overall, one of the most important things in terms of comparing these algorithms is complete data sets. The overall analysis between the two was inconclusive in terms of data size vs performance, but it does show with more missing values and aggressive discretization's, it decreases the performance and accuracy. This contradicted from what I hoped for as I thought a larger data set would give more accurate results, but it was offset but the many holes left in it. Titanic's algorithms performed way above expectations in terms of accuracy vs data size, and it begs to question was my personal decisions of discretization too biased for adult while the Kaggle editor's discretization was just more well thought out and experienced.

Disregarding the holes in the data, it seems that larger data sets benefit better from J48 and KNN, while the smaller ones benefit from SVMs, it maybe be that with more attributes it forces SVM to take too many dimensions into consideration, not only slowing down the process but also making it difficult to separate.

If I was able to next time, I would spend more time into finding a better data set, but also one with multiclass classification as opposed to binary. I would also put more time into testing and tuning the hyperparameters as some combinations might be better optimized for different data sets and when adults got too large it took too long to test everything.

Acknowledgements

Ronny Kohavi and Barry Becker (1996). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Data Mining and Visualization ,Silicon Graphics.

Diego Milla (2018Kaggle[Kaggle.com]) Introduction to Decision Trees(Titanic dataset) for dataset and python script

Weka Data Mining and Machine Learning Software, Waikato University [https://www.cs.waikato.ac.nz/ml/index.html]