1. Datasets

I wanted to explore the differences between a medium-sized dataset and a large dataset when implementing the various machine learning algorithms. Both datasets were discovered in the UCI Machine Learning Repository [1].

The first one chosen is based off of mammography data that is used to identify whether a tumour is malignant or benign [2]. This dataset had 6 attributes with a binary output and consisted of 961 samples. I chose this dataset because of my interest in healthcare and because I believe the application of machine learning has the potential to play a massive role in improving patient care. The second sample chosen was of identifying whether a particular RGB value corresponds to a human skin colour [3]. This dataset only had 3 attributes and the output was also binary, however the number of samples totalled 245,057. 50,859 were valid skin samples while 194,198 were not. My main reason for choosing this dataset was to contrast its large size to the mammography data. Its binary output also helped maintain consistency between the datasets.

Only the mammography data required pre-processing. It was reduced to 830 total samples due to missing values for some of the attributes. 427 of these were benign while 403 were malignant. Some of the attributes, such as shape, were nominal and needed to be broken into multiple attributes to prevent some algorithms, like neural networks, from weighting one "better" than the other. I broke shape down into 4 mutually exclusive binary attributes: round, oval, lobular, and irregular. This way, "round" won't be valued as a lower weight than "oval" if its index value was simply less in the mapping. I followed the same procedure for margin and density. Finally, in order to prevent the instances from being far too specific, I transformed the age attribute into a decade attribute. This would place, for example, ages 60-69 into the 60 decade. The total number of attributes came out to 15 after the pre-processing.

2. Experiments

The data was processed using the Scikit-learn library for Python 3 [4]. In each run, the training set made use of a 4-folded cross-validation procedure and the different scores were recorded. Of the data available the training sizes/portions used were 20%, 40%, 60%, and 80% of the total number of instances available.

The decision tree algorithm used information gain to split the data because the difference in performance between it and Gini is marginal when the two are compared over a wide range of practical applications [5]. The swaying factor was the abundance of documentation on information gain available. Pruning was applied by preventing further splits if the level of impurity that may result is less than some threshold. The results were as follows.

Mammography				
Training Portion	Impurity	Train Score	Average X-Val Score	Test Score
0.2	0	0.903614458	0.782738095	0.805722892
	0.1	0.843373494	0.84375	0.823795181
	0.2	0.843373494	0.84375	0.823795181
	0.4	0.524096386	0.58958	0.512048193
0.4	0	0.903614458	0.786144578	0.799196787
	0.1	0.843373494	0.837349398	0.817269076
	0.2	0.843373494	0.843373494	0.817269076
	0.4	0.506024096	0.584337349	0.47188755
0.6	0	0.897590361	0.807186992	0.828313253
	0.1	0.843373494	0.843317073	0.804216867
	0.2	0.843373494	0.843317073	0.804216867
	0.4	0.504016064	0.50401626	0.530120482
0.8	0	0.893072289	0.813300671	0.777108434
	0.1	0.843373494	0.84336781	0.765060241
	0.2	0.843373494	0.84336781	0.765060241
	0.4	0.515060241	0.515060788	0.512048193

Skin				
Training Portion	Impurity	Train Score	Average X-Val Score	Test Score
0.2	0	0.999979596	0.998612561	0.998775798
	0.1	0.881863255	0.881863335	0.882512268
	0.2	0.881863255	0.881863335	0.882512268
	0.4	0.791924262	0.791924263	0.792594595
0.4	0	0.999969395	0.998908411	0.999041045
	0.1	0.882046887	0.88204689	0.882606182
	0.2	0.882046887	0.88204689	0.882606182
	0.4	0.79299545	0.79299545	0.792103921
0.6	0	0.999959193	0.999075045	0.999081848
	0.1	0.882204116	0.882204135	0.88264999
	0.2	0.882204116	0.882204135	0.88264999
	0.4	0.793211094	0.793211094	0.791334687
0.8	0	0.999959193	0.999137952	0.999265486
	0.1	0.882552475	0.882552486	0.88170244
	0.2	0.882552475	0.882552486	0.88170244
	0.4	0.792807774	0.792807774	0.791071574

The boosted tree version of this made use of the AdaBoost algorithm to focus in on attributes that have more predictive power. By boosting the weights, I expected to be able to get away with a larger impurity allowance without sacrificing too much accuracy on the test scores. The results below mostly confirmed my hypothesis.

Mammography				
Training Portion	Impurity	Train Score	Average X-Val Score	Test Score
0.2	0.18	0.843373494	0.84375	0.823795181
	0.36	0.843373494	0.75625	0.823795181
	0.54	0.524096386	0.524107143	0.512048193
	0.72	0.524096386	0.524107143	0.512048193
0.4	0.18	0.843373494	0.843373494	0.817269076
	0.36	0.843373494	0.843373494	0.817269076
	0.54	0.506024096	0.506024096	0.47188755
	0.72	0.506024096	0.506024096	0.47188755
0.6	0.18	0.843373494	0.843317073	0.804216867
	0.36	0.843373494	0.843317073	0.804216867
	0.54	0.504016064	0.50401626	0.530120482
	0.72	0.504016064	0.50401626	0.530120482
0.8	0.18	0.843373494	0.84336781	0.765060241
	0.36	0.843373494	0.84336781	0.765060241
	0.54	0.515060241	0.515060788	0.512048193
	0.72	0.515060241	0.515060788	0.512048193

Skin				
Training Portion	Impurity	Train Score	Average X-Val Score	Test Score
0.2	0.18	0.895084777	0.894574806	0.896422268
	0.36	0.881863255	0.879659614	0.882512268
	0.54	0.791924262	0.791924263	0.792594595
	0.72	0.791924262	0.791924263	0.792594595
0.4	0.18	0.896237579	0.896237601	0.896099568
	0.36	0.875670768	0.877037913	0.876192743
	0.54	0.79299545	0.79299545	0.792103921
	0.72	0.79299545	0.79299545	0.792103921
0.6	0.18	0.896391311	0.896391334	0.895799965
	0.36	0.875865446	0.875865473	0.876161717
	0.54	0.793211094	0.793211094	0.791334687
	0.72	0.793211094	0.793211094	0.791334687
0.8	0.18	0.896391311	0.896360532	0.895331755
	0.36	0.875865446	0.8761866	0.875173427
	0.54	0.793211094	0.792807774	0.791071574
	0.72	0.793211094	0.792807774	0.791071574

When choosing the implementation for the neural network experiments, I decided to use the logistic sigmoid activation function paired with a stochastic gradient descent solver. ReLU activation was also considered but it was ultimately not chosen because it has the possibility to cause a detriment to learning when paired with this solver in that gradients cannot flow backwards [6]. The results over several different iterations are below.

Mammography				
Training Portion	Iterations	Train Score	Average X-Val Score	Test Score
0.2	1	0.439759036	0.439583333	0.414156627
	5	0.451807229	0.451488095	0.421686747
	10	0.487951807	0.488095238	0.475903614
	15	0.506024096	0.50625	0.483433735
0.4	1	0.34939759	0.361445783	0.369477912
	5	0.478915663	0.503012048	0.413654618
	10	0.503012048	0.503012048	0.47188755
	15	0.503012048	0.503012048	0.47188755
0.6	1	0.33935743	0.385398374	0.388554217
	5	0.493975904	0.371235772	0.469879518
	10	0.497991968	0.49398374	0.469879518
	15	0.670682731	0.510146341	0.63253012
0.8	1	0.414156627	0.400435493	0.427710843
	5	0.353915663	0.386871711	0.415662651
	10	0.623493976	0.555625113	0.602409639
	15	0.676204819	0.692741789	0.704819277

Skin				
Training Portion	Iterations	Train Score	Average X-Val Score	Test Score
0.2	1	0.983575116	0.985186993	0.982810157
	5	0.987798657	0.987288579	0.987268294
	10	0.990675563	0.99045118	0.990048254
	15	0.991654935	0.990757237	0.991083725
0.4	1	0.985431842	0.983717938	0.98506478
	5	0.991093836	0.990083843	0.9907437
	10	0.996031503	0.993736088	0.995817322
	15	0.996837445	0.996153912	0.996946305
0.6	1	0.986044044	0.985261916	0.985942075
	5	0.99549764	0.990893263	0.995480652
	10	0.996674239	0.996626631	0.99661304
	15	0.997007495	0.996857877	0.997072116
0.8	1	0.986865261	0.986059318	0.986839958
	5	0.99636308	0.994332914	0.996531462
	10	0.996817057	0.996740543	0.997061944
	15	0.997194522	0.99702109	0.99755162

For the KNN algorithm, I chose a uniform weight distribution because the spread of values was large in both datasets and there was a large potential for overfitting otherwise (especially on the skin data). For example, just because a specific RGB in the brown range isn't a skin sample shouldn't mean that a close RGB should be classified as a non-skin sample as well, especially since we know that there are a wide variety of brown tinges that do correspond to common skin colours. The results when varying the number of neighbours are below.

Mammography				
Training Portion	Neighbours	Train Score	Average X-Val Score	Test Score
0.2	1	0.86746988	0.662797619	0.754518072
	2	0.84939759	0.73452381	0.760542169
	3	0.861445783	0.747321429	0.823795181
	4	0.819277108	0.729166667	0.789156627
0.4	1	0.86746988	0.756024096	0.708835341
	2	0.84939759	0.737951807	0.724899598
	3	0.861445783	0.804216867	0.787148594
	4	0.870481928	0.807228916	0.821285141
0.6	1	0.893574297	0.763056911	0.768072289
	2	0.853413655	0.740796748	0.774096386
	3	0.857429719	0.817317073	0.813253012
	4	0.851405622	0.811317073	0.807228916
0.8	1	0.871987952	0.763645436	0.777108434
	2	0.834337349	0.772609327	0.753012048
	3	0.861445783	0.811930684	0.753012048
	4	0.856927711	0.810361096	0.734939759

Skin				
Training Portion	Neighbours	Train Score	Average X-Val Score	Test Score
0.2	1	0.999979596	0.999510309	0.999444008
	2	0.999653139	0.999326672	0.999239974
	3	0.999673543	0.999408285	0.999301184
	4	0.99948991	0.999020613	0.999117554
0.4	1	0.999959193	0.999459306	0.999428707
	2	0.99960213	0.999398094	0.999367498
	3	0.999632736	0.999449101	0.999415105
	4	0.999540919	0.99931648	0.99931989
0.6	1	0.999952392	0.999435503	0.999510319
	2	0.999619136	0.999469509	0.999387899
	3	0.99964634	0.999496714	0.999469512
	4	0.999591931	0.999449106	0.99933689
0.8	1	0.999948991	0.999489913	0.999510324
	2	0.999668443	0.99947971	0.99955113
	3	0.99970415	0.999540921	0.999591937
	4	0.999617435	0.999438903	0.999449115

For SVM learning, the kernels chosen were linear and polynomial of degree-2. Since linear is simply a degree-1 polynomial, I wanted to compare it with the next logical progression and analyze the results. For each training rate, the different kernels were applied against a single iteration and 100 iterations to record the effect of repetition on the datasets. The results are as follows.

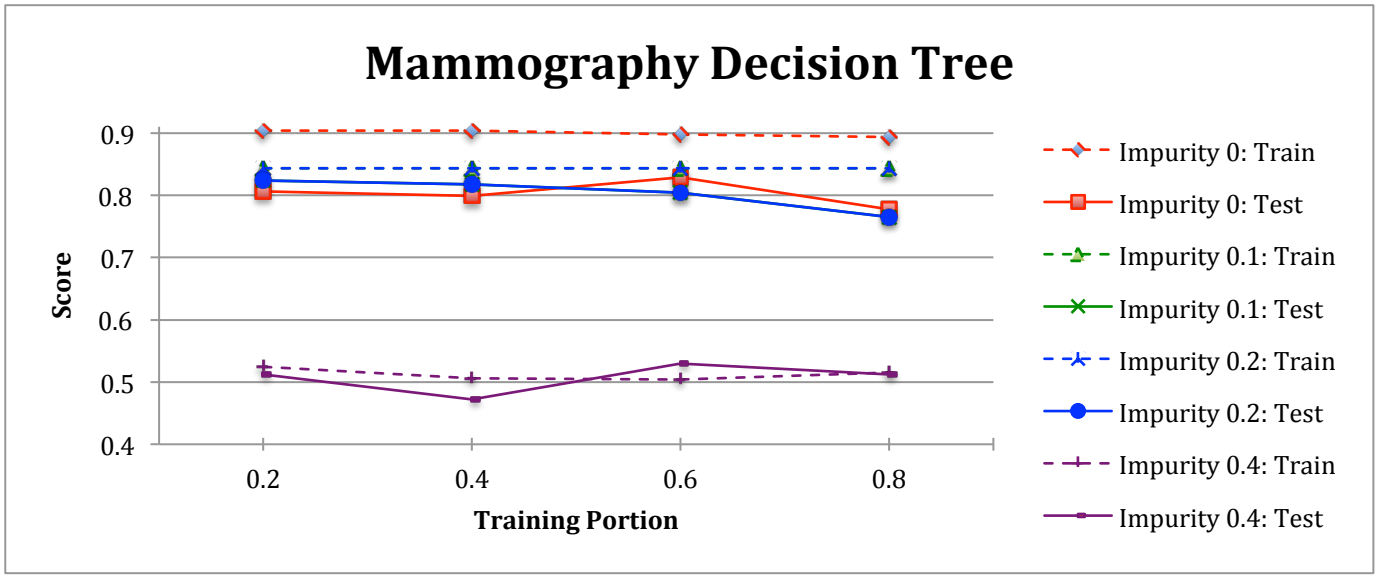
Mammography					
Training Portion	Iterations	Kernel	Train Score	Average X-Val Score	Test Score
0.2	1	Linear	0.518072289	0.523511905	0.53313253
		Polynomial	0.518072289	0.523511905	0.53313253
	100	Linear	0.84939759	0.753571429	0.817771084
		Polynomial	0.704819277	0.480059524	0.733433735
0.4	1	Linear	0.704819277	0.647590361	0.638554217
		Polynomial	0.659638554	0.608433735	0.604417671
	100	Linear	0.626506024	0.798192771	0.590361446
		Polynomial	0.728915663	0.692771084	0.742971888
0.6	1	Linear	0.495983936	0.53198374	0.469879518
		Polynomial	0.495983936	0.53198374	0.469879518
	100	Linear	0.457831325	0.63195122	0.439759036
		Polynomial	0.704819277	0.664731707	0.737951807
0.8	1	Linear	0.493975904	0.577309018	0.487951807
		Polynomial	0.763554217	0.62192887	0.710843373
	100	Linear	0.546686747	0.753075667	0.403614458
		Polynomial	0.573795181	0.641843586	0.524096386

Skin					
Training Portion	Iterations	Kernel	Train Score	Average X-Val Score	Test Score
0.2	1	Linear	0.49197119	0.459389898	0.493557634
		Polynomial	0.553365571	0.445960965	0.554930986
	100	Linear	0.841770215	0.818225064	0.841843241
		Polynomial	0.181224623	0.670671326	0.180865715
0.4	1	Linear	0.488533186	0.430022351	0.486938484
		Polynomial	0.362377834	0.383078527	0.359764682
	100	Linear	0.506416927	0.528663644	0.504349305
		Polynomial	0.898971659	0.530292883	0.899316489
0.6	1	Linear	0.811778228	0.742626406	0.810962733
		Polynomial	0.689935661	0.725165333	0.687634535
	100	Linear	0.608335487	0.338723622	0.605602767
		Polynomial	0.770576873	0.531902781	0.770258001
0.8	1	Linear	0.680037746	0.432726287	0.679955929
		Polynomial	0.577683695	0.674974542	0.574287929
	100	Linear	0.536810426	0.676889258	0.534481352
		Polynomial	0.875457165	0.598673211	0.874234881

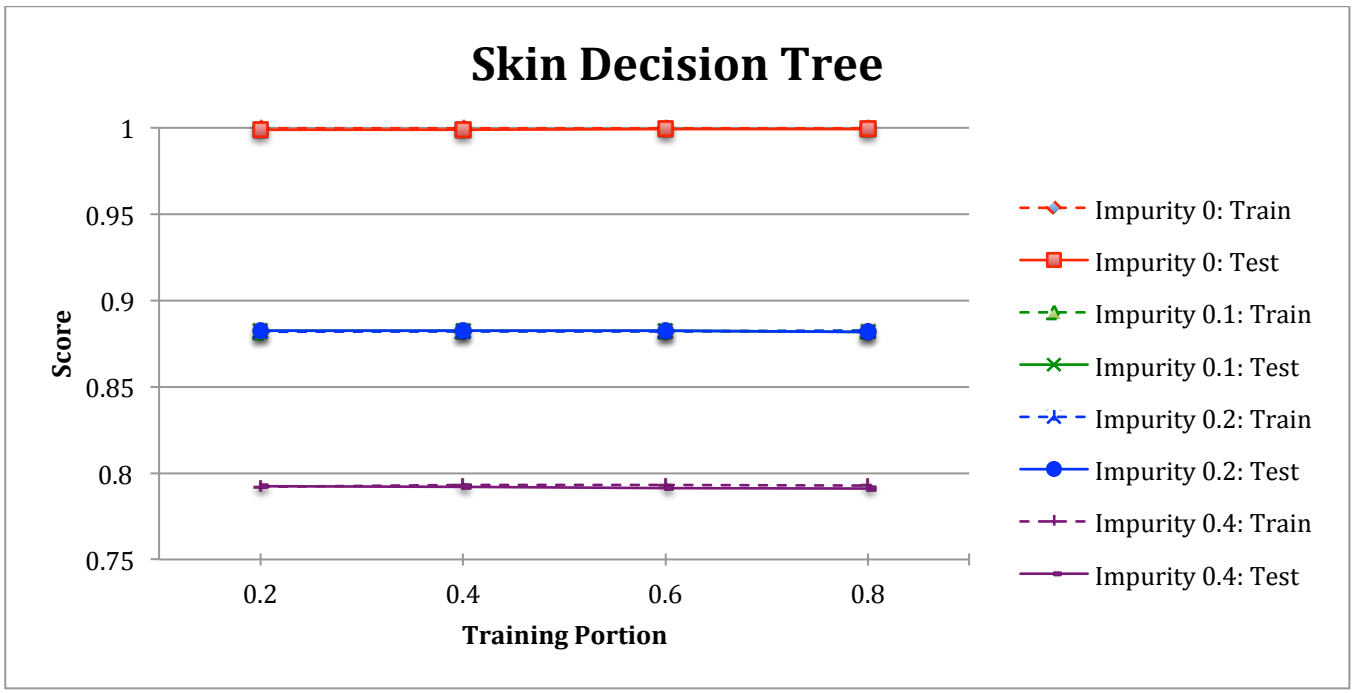
3. Analysis

To ease readability in the graphs in this section, dashed lines join the training scores, while the solid lines join test scores.

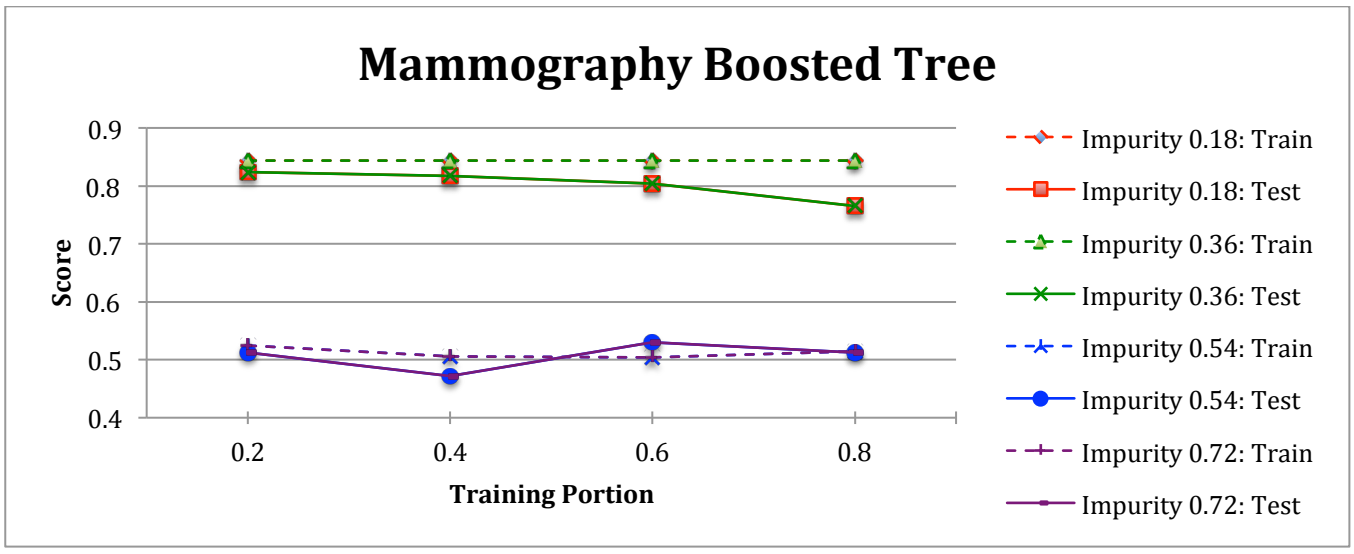
The decision trees for the mammography data had the least error on the testing data when the training rate was 60% for impurities 0 and 0.4. This implies that beyond this rate tended to overfit the data for these pruning levels. There is further evidence of this fact by the trend of the error rate increasing on the testing data as the training portion rises while the training error rate was relatively constant. In contrast, the best testing scores for impurities 0.1 and 0.2 was only 20%. So while the best single test run was on impurity level 0 with a testing error rate of 18.2%, both impurity levels 0.1 and 0.2 only needed a 20% training portion to accomplish a very close testing error rate of 18.6%. This leads me to believe that very little score is sacrificed by pruning and the gains in performance can far outweigh the deficit. Although, this would not be a worthwhile trade-off for this problem domain due to the repercussions of misclassifying cancer tissue.



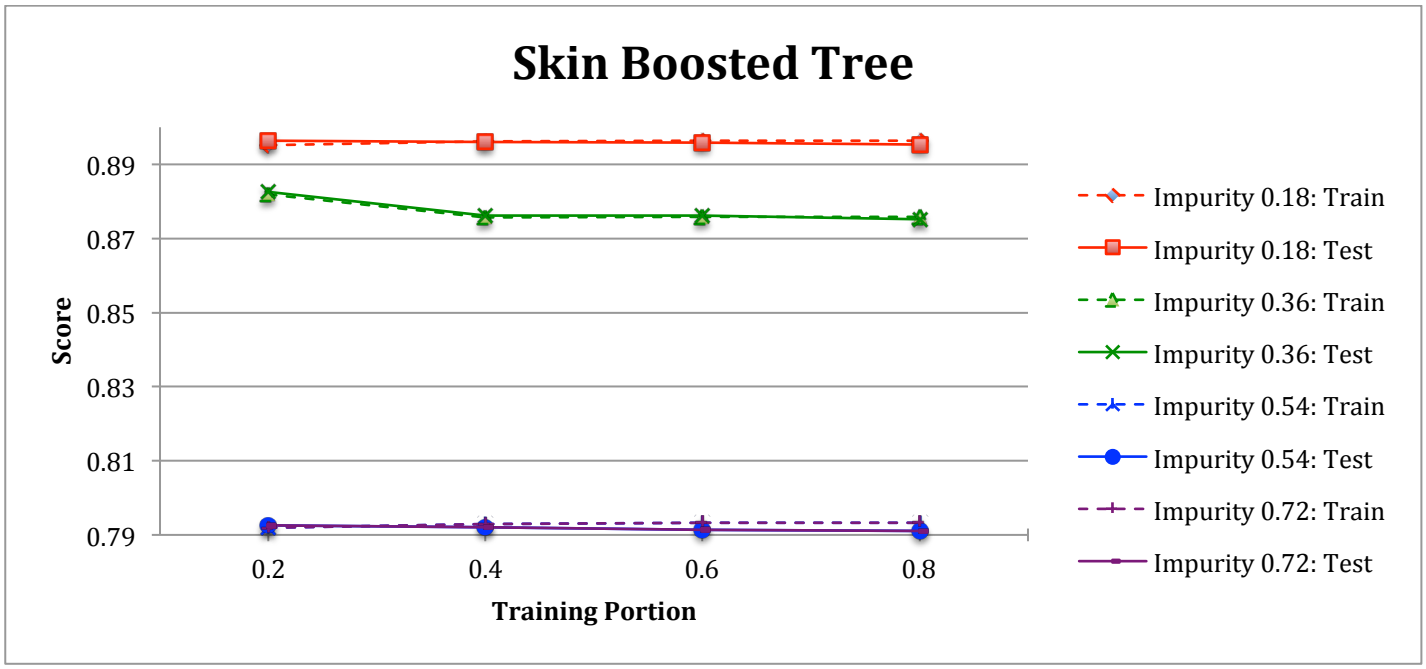
As opposed to the mammography output, the training and testing scores were very similar in all runs on the skin data. The un-pruned tree was the significantly better structure in this case with a <0.1% error rate on both the training data and the testing data as it gets a higher training portion. Impurities 0.1 and 0.2 are tied in scores once again across all training rates. Their lowest testing error rate is at 11.8%, however it's very close for all training portions. Very little is sacrificed if only 20% of the data is used to train for all impurity levels. This may be attributed to the fact that there are only 3 features and that it's very easy to tell apart most colours as not being representative of skin pigment. Colours that are green, purple, and pink fit such colours and the decision tree algorithm does a fantastic job of identifying this with a small training portion.



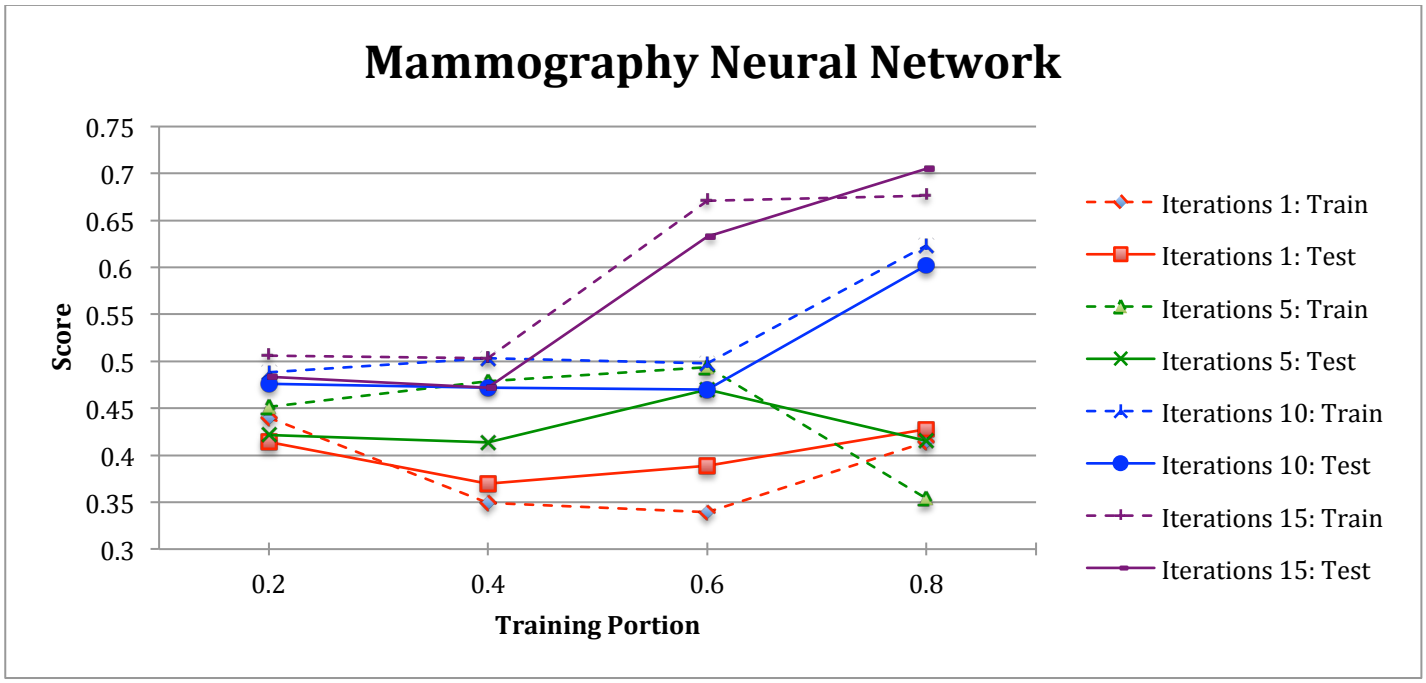
The most noteworthy change in the results when using AdaBoost on the mammography data was that even with a high impurity level like 0.36, the algorithm was able to have an error rate of 18.6% on the testing set at a 20% training portion. This is the same as the error rates at impurity levels 0.1 and 0.2 before. The boosting seemed to have allowed additional impurities to creep in to the splits without forgoing any accuracy at all in this case. However, as the boosting is stressed even more with impurity levels 0.54 and 0.72, the excessive pruning hugely detriments score, even dipping below 50% at its worst.



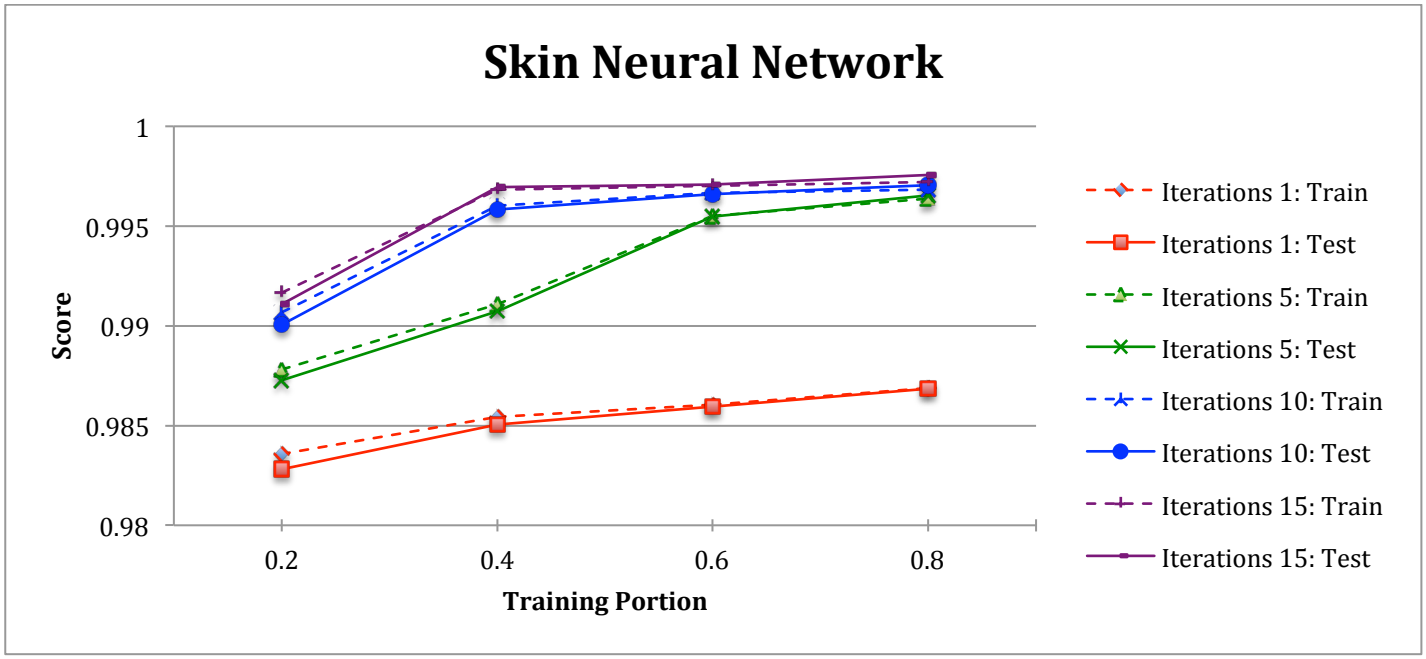
Boosting on the skin data was ineffective for both scores and, in the case of no pruning, yielded much worse results if anything. This may be due to the fact there's no one important attribute out of red, green, or blue, and so there would be nothing to gain out of weighting these. Weighting them would be counter-intuitive in this regard, which would result in inaccurate labels.



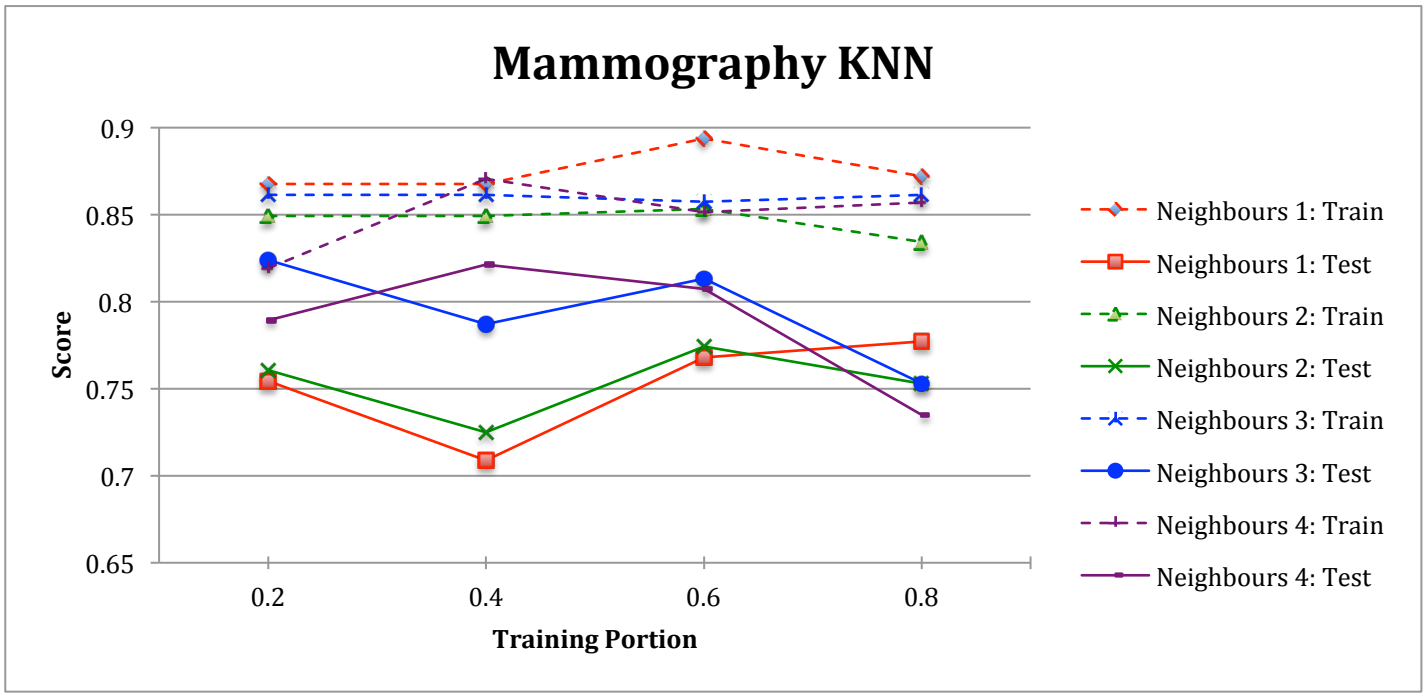
The neural network results on the mammography dataset are unsurprising in that the error rate on the testing set decreases as more iterations are added, with the exception of 5 iterations near the end. What did surprise me was how poorly it did in general. The lowest error rate was 29.5% between both training error and testing error.



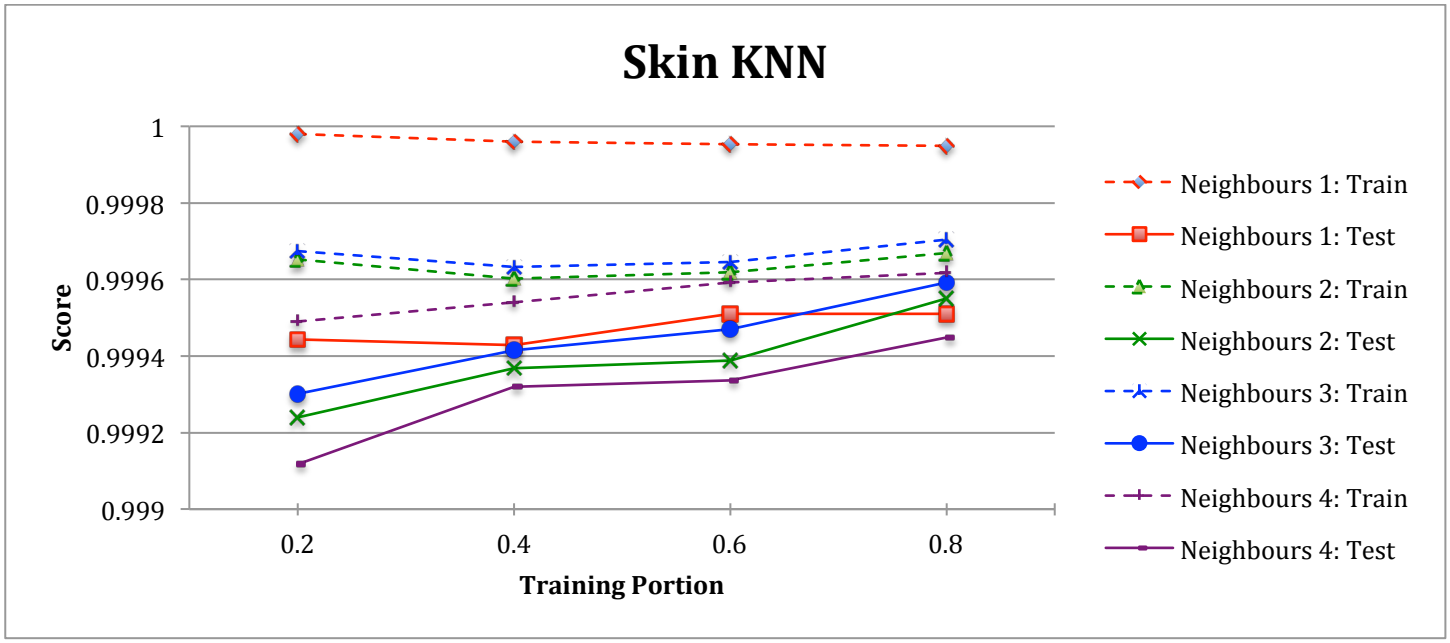
In contrast, the neural network scored extremely well on the skin dataset. The error rates were as low as 0.3% on the training data and 0.2% on the testing data. There was also very little overfitting anywhere, the scores simply improved as more training was provided and as more iterations were added.



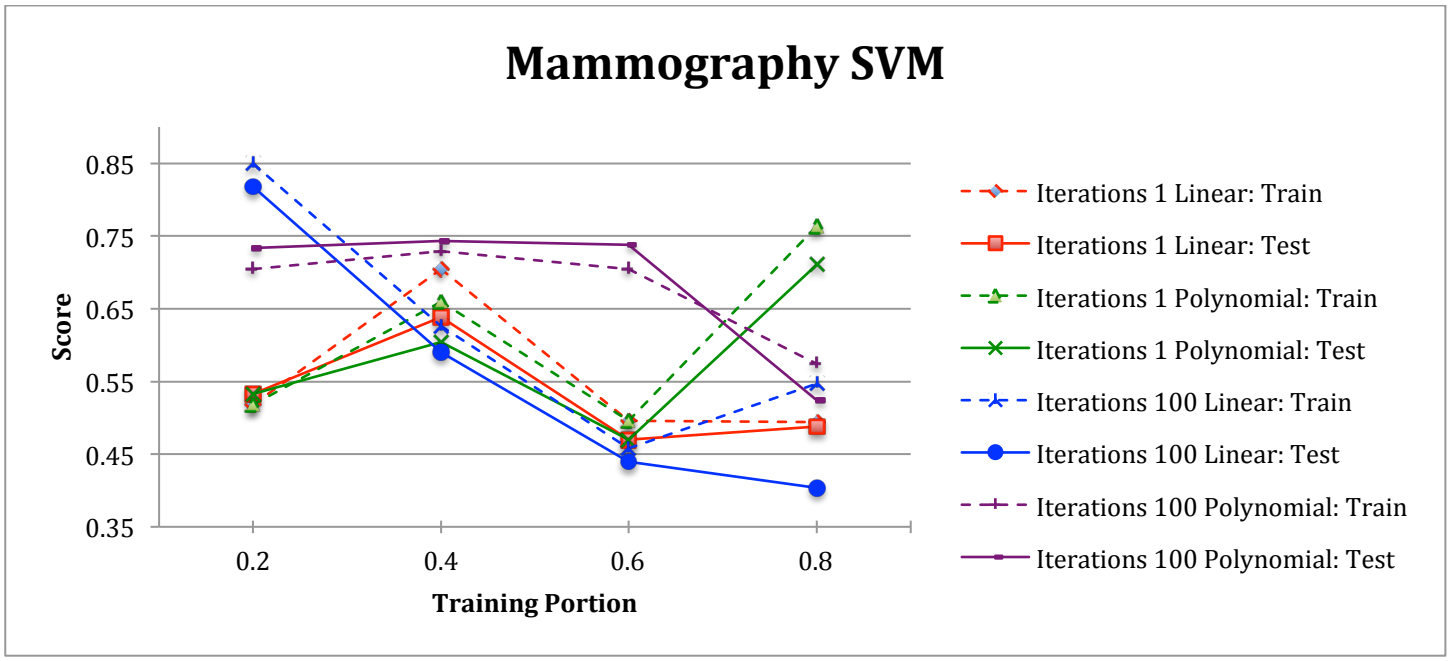
KNN on mammography data was the most successful on the test score with 20% of the samples used for training with K = 3. Aside from K = 4, the other runs consistently produced a testing error increase between training portions 20% and 40% while the training error was mostly stable. This indicates overfitting early on and that for a low K-value, this algorithm performs well with a smaller training size.



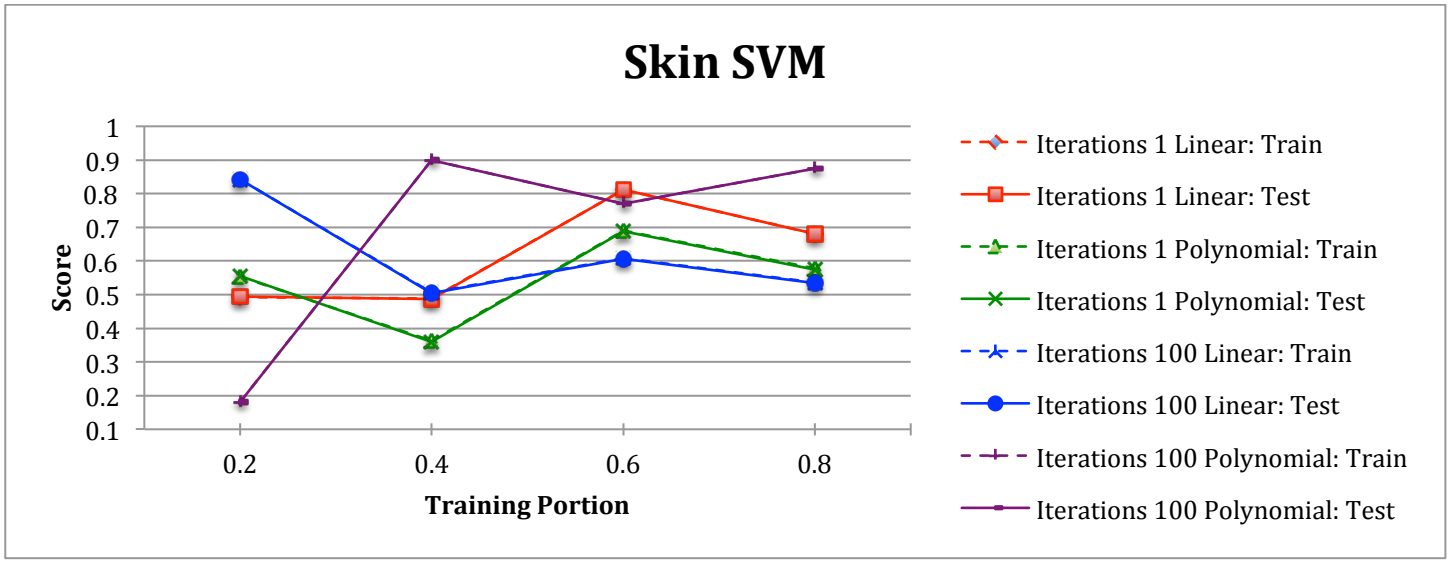
KNN on the skin data was, simply put, the most successful of all the experiments. The error rate in the worst case was still <0.1%. While the best score was for the K = 1 training data and K = 3 training data, the difference was marginal. The success of this experiment could be attributed to the size and spread of the dataset such that there was always a close neighbour nearby. Interestingly, KNN was the only algorithm to have the lowest training score still be higher than the greatest testing score for this dataset. This could indicate the challenge of generalizing based on the K-values used, but again, to reiterate, the results on the skin data were still remarkably close to perfect.



The SVM approach yielded a great deal of variability in the results on the mammography dataset. The polynomial kernel at 100 iterations was the most consistent, hovering at around a 25% testing error rate. However, the 100 iteration linear kernel is the most interesting in that it held both of the highest scores and the lowest test score. The training error rate did drop between 60% and 80% training portions though, indicating the overfitting in that segment.



This approach on the skin data had even more variability, the most seen in all the experiments. However, both scores remained in sync throughout every run. The most immediately noticeable item of interest is the errors dropping close to 70% when increasing the training portion from 20% to 40% for the 100 iteration polynomial run of the algorithm. This experiment demonstrated very well how important the training portion can be and how quickly scores can vary based off of this parameter. At its worst, the errors exceeded 80%, that's impressively inept considering that random chance at labelling correctly is 50%.



4. Discussion

The overall best performer for both datasets was the KNN algorithm. This is easier to see in the case of the skin data because the spread of scores was incredibly low while always remaining closer to perfect scores than any other algorithm tested. While KNN did not hold the highest testing score for the mammography data, it was close, and the lowest one was well above that of the decision tree (19.1% higher). I chose testing scores to determine the best performing algorithm mainly because generalizability is the ultimate decider in how well the algorithm learned from the input instances.

The worst performer on the mammography data was the neural network. The variability in testing scores was 34.4% but the real issue was that most of the runs scored less than 50% on a binary label. It would be interesting to see how greatly increasing the iteration count would affect the scores because they did have some positive impact. Unfortunately, this was not done because increasing it on the skin data was far too slow to be feasible and consistency between the input parameters for both datasets was a key component to the experiments.

While the SVM algorithm had some low error rates, it also had some of the highest for the skin data, including one run that held a testing error rate of 81.9%. The huge range was the primary reason for it being the worst performer for this dataset. I attempted higher order polynomials but the execution could not complete. However, it may not have actually helped in this case due to the sporadic scoring seen. At some training sizes, the linear kernel actually performed better.

In general, the graphs showed more consistent scoring between training and testing sets for the skin data. One reason to explain this is that certain colours, such as purple, can be very easily determined as non-skin. In contrast, the determination of malignant breast tissue can be very challenging. For example, while it may be true that breast cancer is more common as age increases, it's certainly not the case that younger individuals cannot acquire it as well [7]. Furthermore, there is a strong possibility of hidden variables obscuring algorithmic scores. One such variable would be the impact of the age when a woman had her first child [8].

There is very little room for improvement on the skin data when using KNN, however there are several enhancements that can be made for KNN on the mammography data. First would be to identify other hidden variables and to account for them in the learning process. Another suggestion would be to greatly increase the number of neighbours. Generally, increasing the K-value for this dataset increased the scores. This may have been because more neighbours could have an easier time masking outliers. Finally, perhaps a uniform weight distribution isn't the best approach for the data. It would be worthwhile testing a different distribution technique such as distance-based tactics to see their impact.

Finally, something that wasn't discussed was the use of cross-validation. The average cross-validation scores were also recorded in section 2 and, aside from the SVM algorithm, these scores were a great indicator of the generalizability of the algorithms when comparing them to the test scores. They did however give quite a poor indication of applicability to test data in the SVM runs and this may have been because the polynomial orders weren't high enough to allow for better separability.

There was a great deal to learn from the experiments. While, the skin data did produce more consistent scores between the training and testing runs, it may not only have been because of the advantage it had in instance count. The type of data available for the mammography learning process could've been the larger difference. Also, the power of cross-validation was on full display in the experiment output. Its predictive power can be a very useful tool to approximate generalizability.

References

[1] D. Dua and E. Karra Taniskidou, *UCI Machine Learning Repository*, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/index.php. [Accessed: 09- Jan- 2019].

[2] M. Elter, R. Schulz-Wendtland and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process", *Medical Physics*, vol. 34, no. 11, pp. 4164-4172, 2007. Available: 10.1118/1.2786864 [Accessed 13 January 2019].

[3] R. Bhatt and A. Dhall, "Skin Segmentation Data Set", *UCI Machine Learning Repository*. [Online]. Available: https://web.archive.org/web/20181222161344/https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation. [Accessed: 13- Jan- 2019].

[4] "scikit-learn: machine learning in Python — scikit-learn 0.20.2 documentation", *Scikit-learn.org*. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 10- Jan- 2019].

[5] L. Raileanu and K. Stoffel, "Theoretical Comparison between the Gini Index and Information Gain Criteria", *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77-93, 2004. Available: https://link.springer.com/article/10.1023/B:AMAI.0000018580.96245.c6. [Accessed 19 January 2019].

[6] A. Agarap, "Deep Learning using Rectified Linear Units (ReLU)", pp. 1-7, 2018. Available: https://arxiv.org/pdf/1803.08375.pdf. [Accessed 21 January 2019].

[7] J. Kelsey, "A review of the epidemiology of human breast cancer.", *Epidemiologic Reviews*, vol. 1, no. 1, pp. 74-109, 1979. Available: https://www.ncbi.nlm.nih.gov/pubmed/398270. [Accessed 29 January 2019].

[8] B. MacMahon et al., "Age at first birth and breast cancer risk", *Bulletin of the World Health Organization*, vol. 43, no. 2, pp. 209-221, 1970. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2427645/. [Accessed 30 January 2019].