Ragy Haddad | Graduate Student
CS 7641

Supervised Learning Project

Content/Map:

Data Sources:

http://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data
https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq

Dataset I Explanation:

For my first classification problem I will be using a packet capture dataset that was hosted to build a network intrusion detector, the goal is to create a predictive model to classify normal connections 'good' and attack connections 'bad'. The dataset is created to simulate network packet data in a military environment.

The motive behind choosing the classification problem were two things:

1. Gaining Intuition about solving cyber security issues using machine learning.
2. Creating a model for some servers I manage to predict malicious attacks and hopefully the model can be integrated into some python scripts I use to analyze packet captures.

Dataset II Explanation:

For my second classification problem I will be classifying different types of cancer tumors based on gene expression profiles. The reason why I chose this topic is because I am a current Bioinformatics student and there is a huge influx of gene expression data in the field that can be analyzed using machine learning techniques to draw inferences. Gene expression profiles are highly multivariate and this can also help compare different algorithm performance on the accuracy compared to the cyber security intrusion dataset.

Feature Explanation And Pre-Processing for Dataset I:

The dataset contains packet data and a label of the attack type the packet data corresponds to. So a row in the dataset looks like the following table:

Total number of features: ***39*** *+* ***general label.***

Total number of rows: ***300,000***

Noise: ***The dataset includes some noise where the same features occasionally correspond to different labels.***

F1	F2	Fn	Label

Handling Categorical Features:

Categorical features were transformed to *Integer* labels using *sci-kit learn's OneHotEncoder*

Feature Explanation And Pre-Processing for Dataset II:

Gene 1 Expression	Gene 2 Expression	Gene n Expression	Label

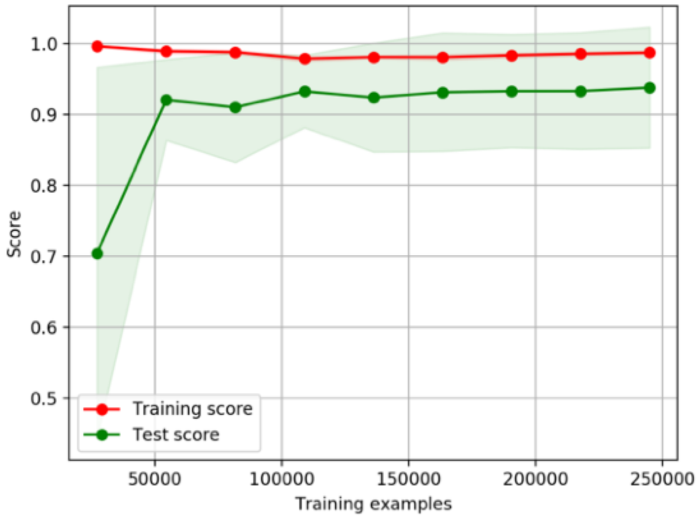
Total number of features: ***20532*** *+ label.*

Total number of rows: ***800***

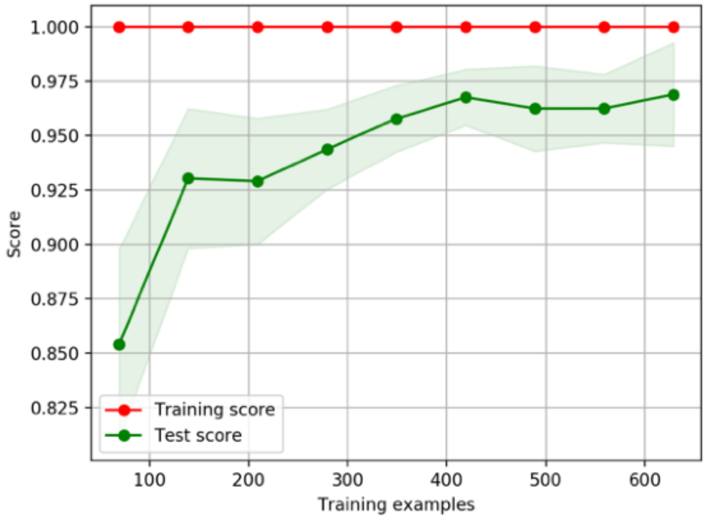
Noise: ***The dataset includes minimal noise.***

Decision Tree

Cyber Security Intrusions



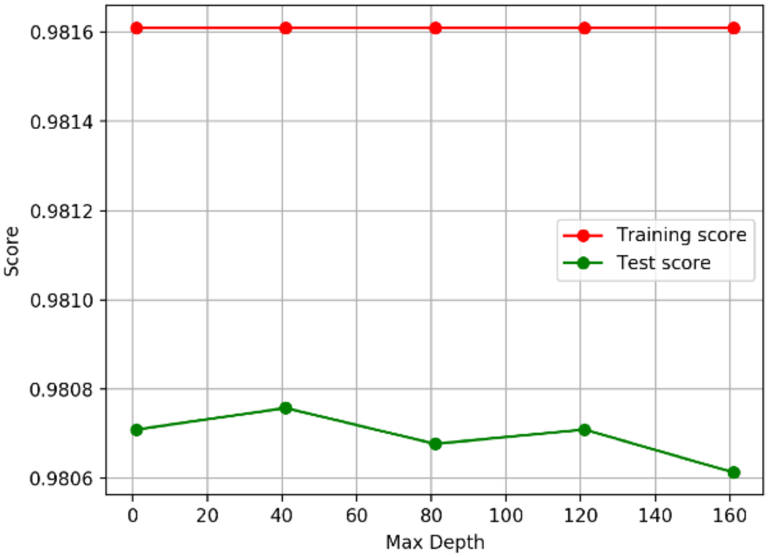
Cancer Cells



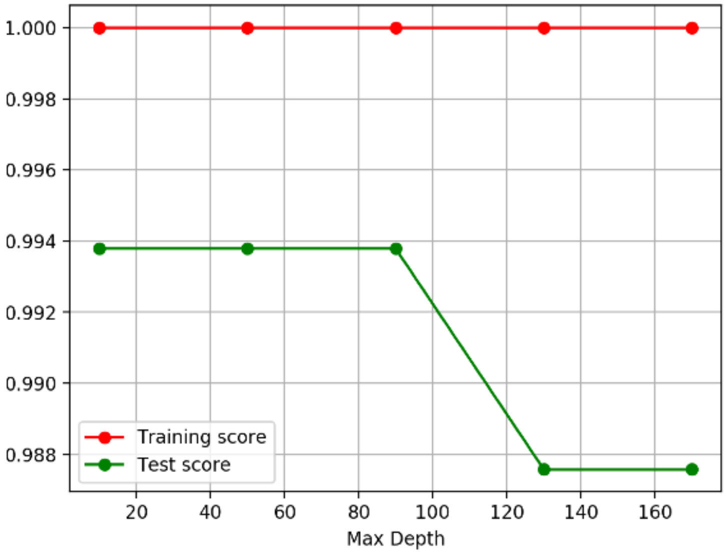
In the figures we plot the train size vs the accuracy of the model to inspect the level of overfitting that is happening during fitting the model. From the figures, there is a clear trend on how it is not necessarily good to fit the model with more data since the model becomes less general and becomes less accurate when over trained (**overfitting)**. In addition, we can also see that under training your model can also cause for **under-fitting.** In my case, I chose to use the ratio of 80% for training and 20% for testing. The reason is to slightly over fit the model then apply **post-pruning** in order to create a more robust model that is accurate and generalizes well with the data. In the cyber security data there is some noise so the over fitting trend isn't exactly clear but it is still occurring to an extent.

Decision Tree Pruning:

Cyber Security



Cancer Cells



Pruning effect on accuracy:

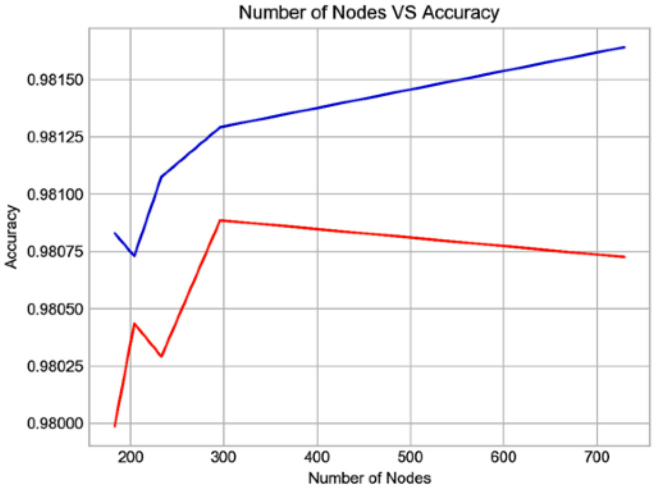
The figures above show the effect of pruning on the accuracy, as we limit the max depth the tree we can see that we reach an optimal accuracy level, and as the maximum depth increases we see the effect of overfitting (having too many nodes) on the accuracy

Pruning for Cyber Security:

For cyber security dataset pruning I tried pruning the trees using this algorithm from stackoverflow

figure on the right, This method is a post-pruning method that performs significantly well for ***bi-classification problems.*** However I decided to use max depth for validation since I wanted to be consistent with my analysis. In addition max depth still performed well.

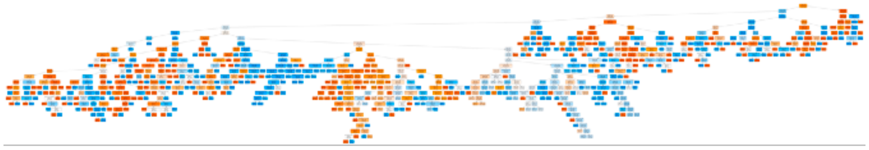
Cyber Security Intrusions Manual Post Pruning



Pruning for Cancer Cells:

For cancer cells I chose to use max depth as a method of pre-pruning, the reason why is because the method mentioned above for post pruning does not perform as well in ***multi-classification problems*** this happens due to over pruning since the pruning rule was based on minimum class count which can become zero in multi-classification problems resulting in over pruning. Another approach could be pruning based on impurity as the pruning rule.

Pre-Pruning Representation of Tree



Post-Pruning Representation of Tree

__Decision Tree performance with optimal parameters:__

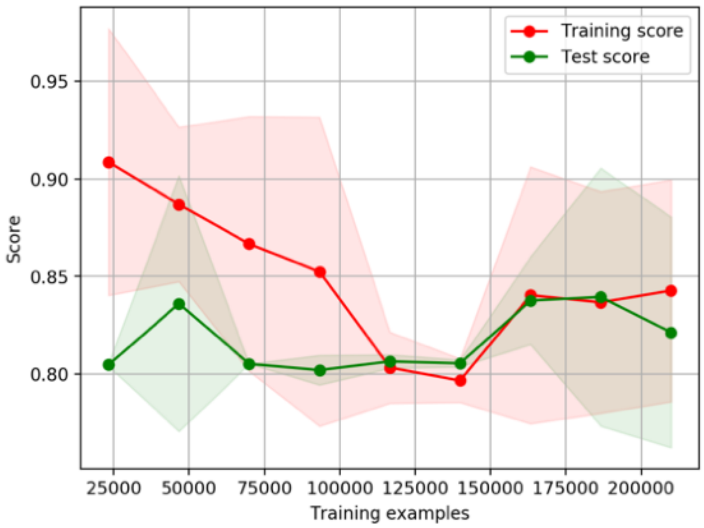
	Cyber Security	Cancer Cells
Train Size	0.8	0.8
Test Accuracy	**0.9807199734216421**	**0.968944099378882**
Train Accuracy	**0.9815680690795517**	**1.0**

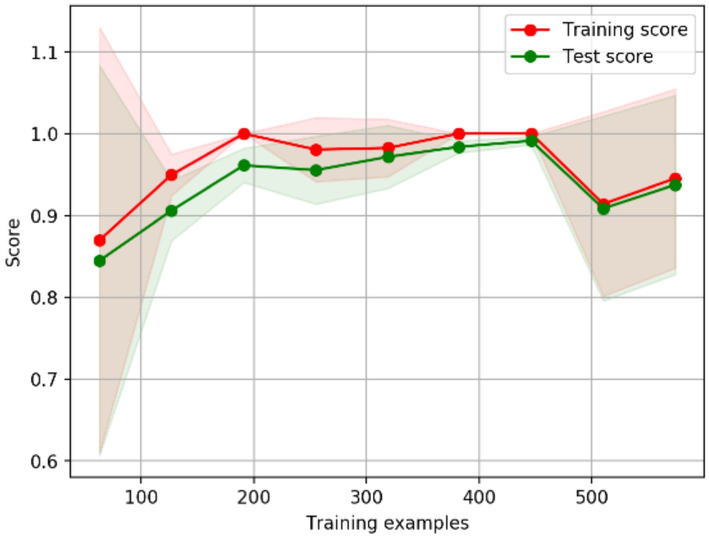
For classification of cyber security attacks the decision trees appeared to be performing slightly better than cancer cell classification, and that is using optimal parameters and pruning for both cases.

__Neural Networks__

Neural Network is a very strong algorithm that incorporates multiple perceptrons with activations and weights incorporated in them, neural networks have hidden layers which we can expend to create more complex networks.

In my analysis I used **logistic activation** for as my activation function,('RELU' performed slightly worse but was **faster**) that is to control my activation for my neurons, since a lot of the variables I am handling are sensitive and a small change in them can have a strong effect. And then I applied stochastic gradient descent with 'adam' setting (recommended for larger datasets) stochastic gradient descent to converge to a local minima but 'lbfgs' seemed to be more accurate on cancer cells. I applied multiple variations to the number of hidden layers and the number of nodes per layer. In addition I applied multiple tests on training and testing to test for overfitting of the data.

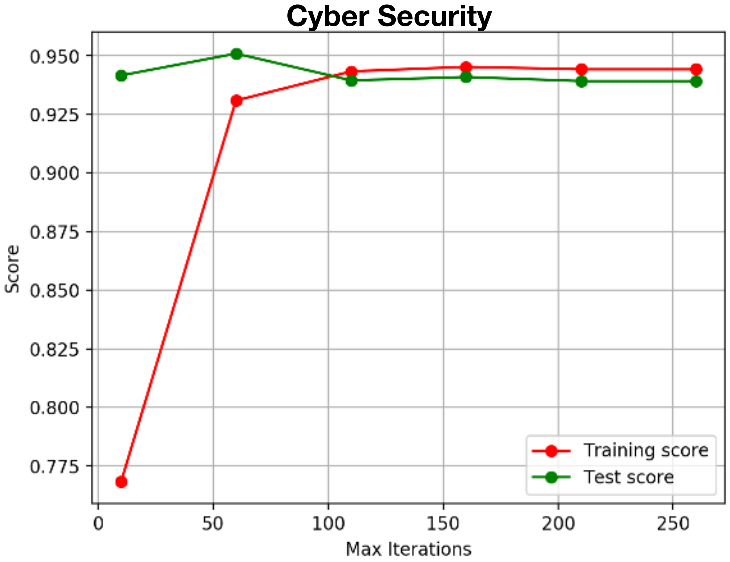


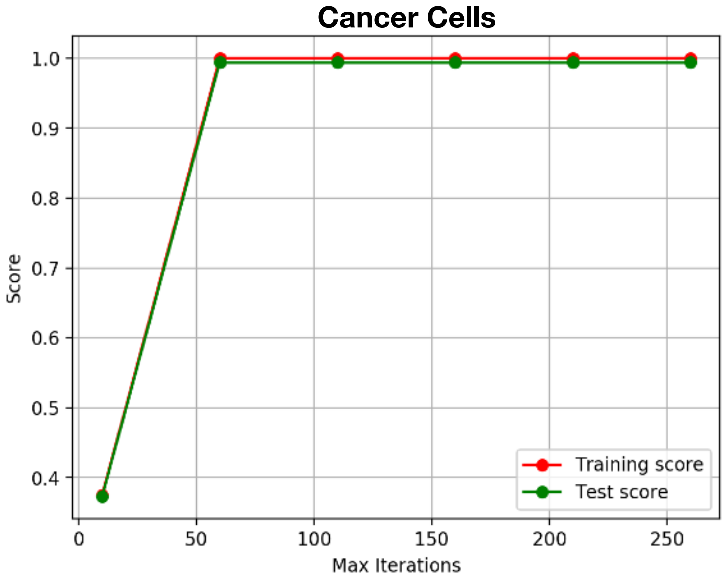


The Figures above show how the neural networks are overfitting as we increase our training examples size the model tends to overfit on the test score.

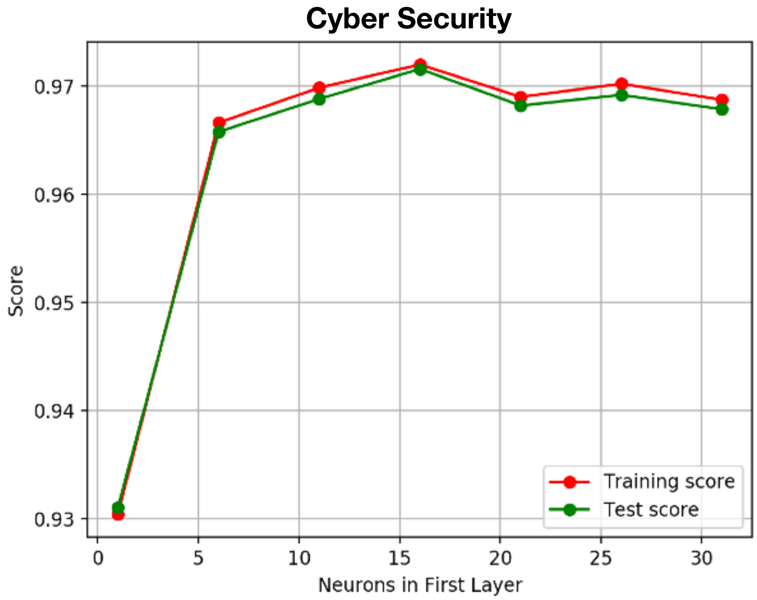
These figures help us find our optimal train size which ends up being 80% of our datasets for each of our classification problems.

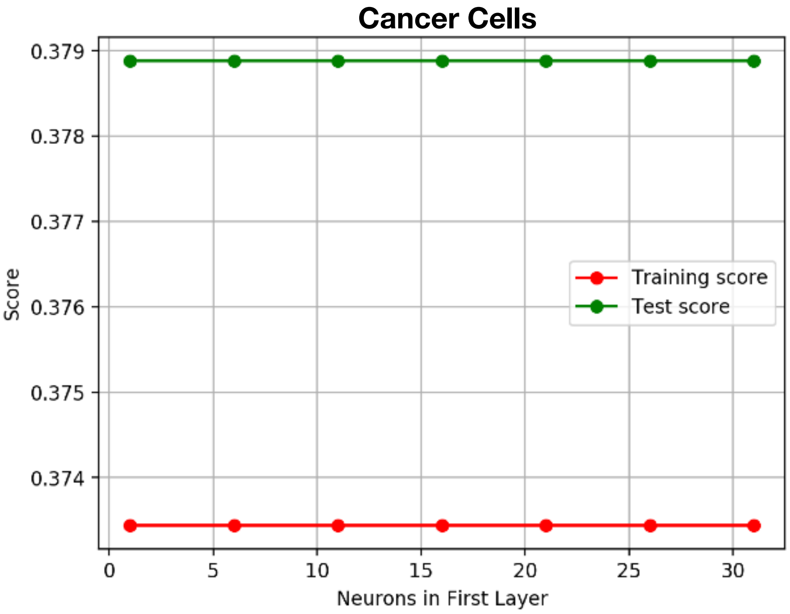
Now we move on to experimenting with different numbers of iterations (epochs) for our neural network, that is the number of times we back propagate and readjust the weights based on the error calculated.





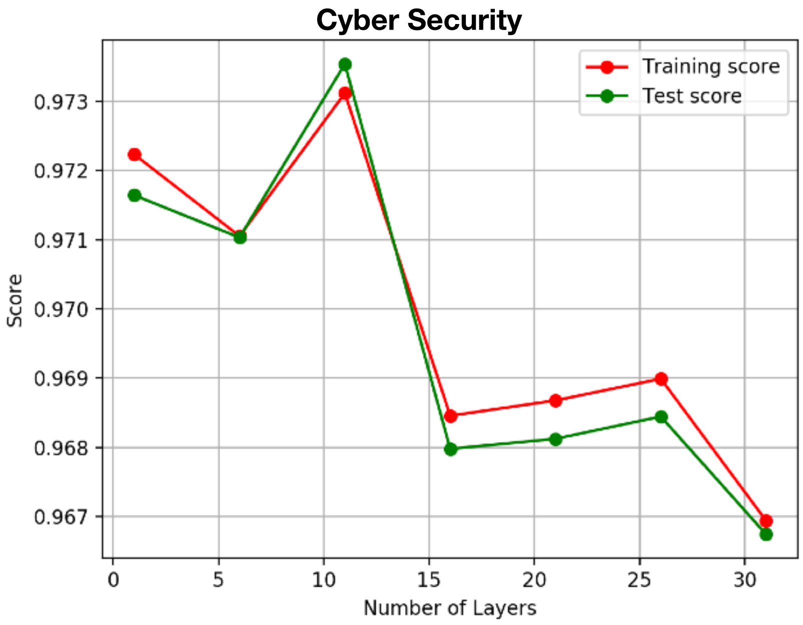
The figures show how the number of iterations (epochs) can influence the accuracy score of the mode. It is obvious that it is more obvious in the cyber security dataset that more iterations can have a negative effect on the accuracy of the model. In the cancer cell dataset increasing the max iterations did not have significant negative effect on the model accuracy however the time taken to run the model was significantly higher. So conclusions about iterations is that they can do two things **a)** cause overfitting of our model **b)** can significantly increase the time in training the model and without significant value.

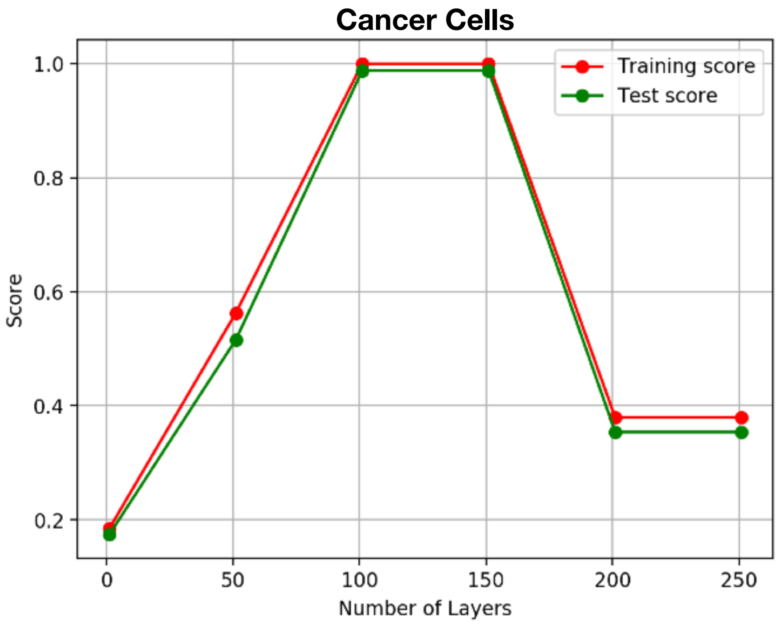




In the above figures we are holding the
number of layers to 1 and changing the number of neurons in that layer. For the Cyber security dataset you can clearly see how the number of neurons is affecting accuracy, the number of neurons is performing well for cyber security because the data is linearly separable which is also demonstrated by the decision tree. However for cancer cell classification, the dataset is highly multivariable and a lot of it is non linearly separable which could explain why the performance is low with just one layer.

Now that we identified some good counts for the neurons in the first layer we will keep them constant and now vary the number of layers of the neural network



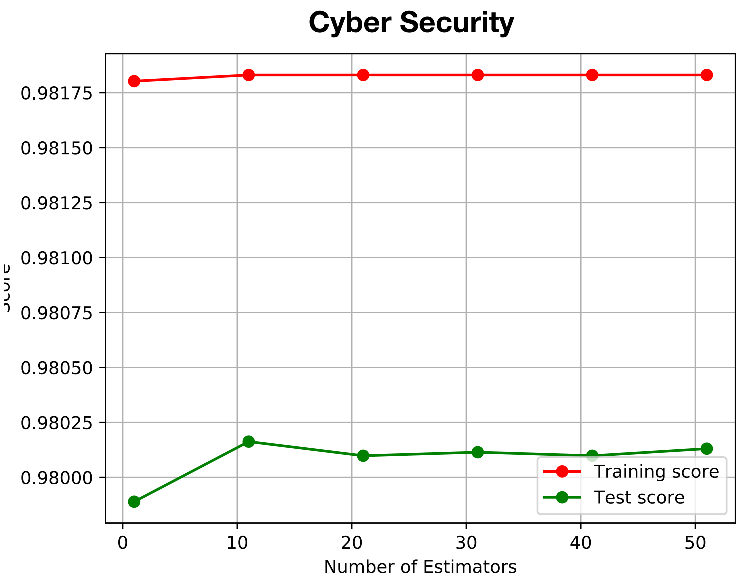


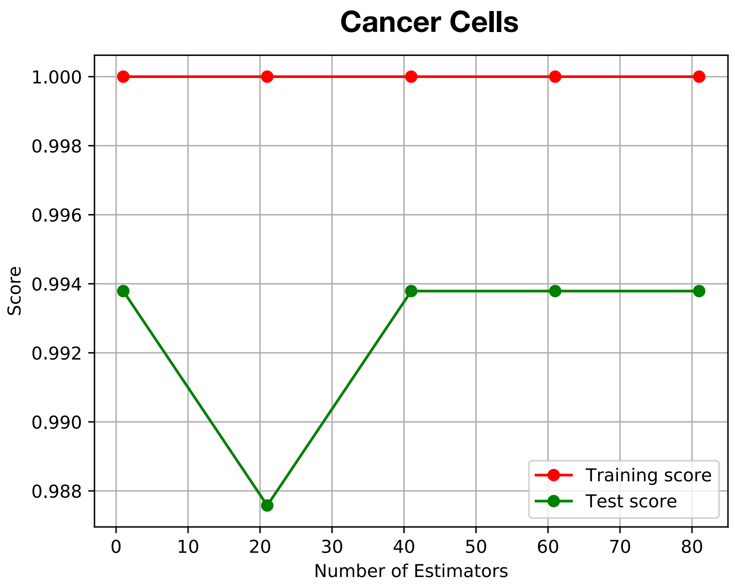
It now becomes obvious the cancer cell dataset is highly affected with the number of layers. There is a significant increase in performance of the neural network model for the multivariable non linearly separable data compared to a minimal increase of performance to the cyber security dataset. In addition it is clear in both figures that increasing the number of layers beyond a certain level can be harmful for the model and could cause overfitting of the model.

	Cyber Security	Cancer Cells
Train Size	0.8	0.8
Test Accuracy	**0.97345**	**0.9978**
Train Accuracy	**0.9815680690795517**	**1.0**

Adaptive boosted Decision Tree

Adaptive boosting, is a method of machine learning by which you combine a set of weak learners (ones that perform slightly better than random guessing) and combine their output as a weighted sum that represents the final output of the output classifier.





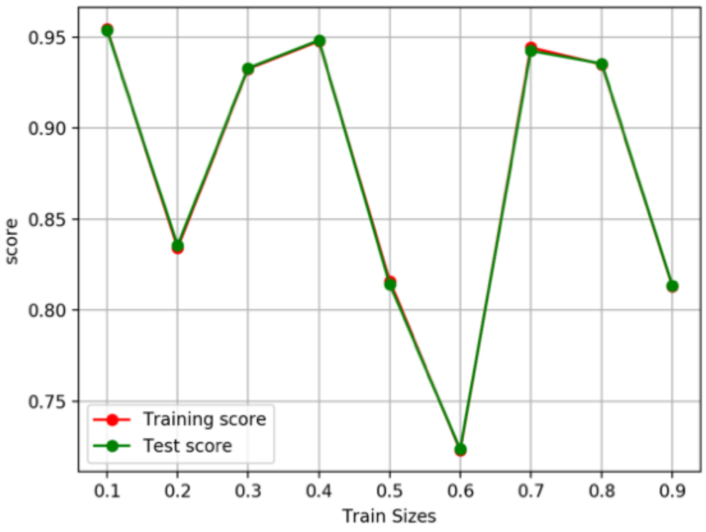
The way adaptive boosting achieves its power is by carving up the hypothesis space in several ways. The learning process is done iteratively. The number of estimators in the above figure shows the amount of times we carve up the space. It now becomes more obvious why adaptive boosting can be sensitive to noise in the data since carving the space becomes more difficult the more noise you have. For our data adaptive boosting performed well for cyber security dataset and significantly well for cancer cells **in both cases adaptive boosting has performed only slightly better than our regular decision tree. However not by a significant**

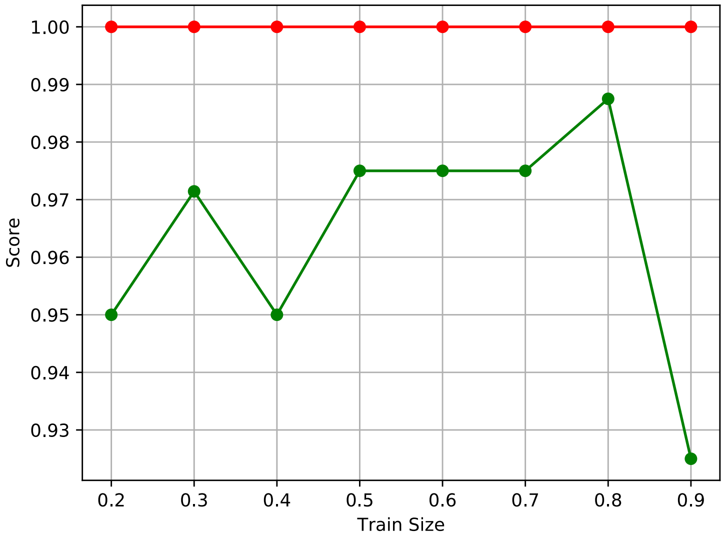
amount and that is because boosting needs weak learners which our initial tree was not really week

	Cyber Security	Cancer Cells
Train Size	0.8	0.8
Test Accuracy	**0.9820756840176189**	**0.9875776397515528**
Train Accuracy	**0.9813843575553707**	**0.990625**

Support Vector Machine:

For my SVM I implemented linear kernel for the support vector machine, SVM is a strong algorithm and is usually capable of performing well on noisy data and can also identify outliers.



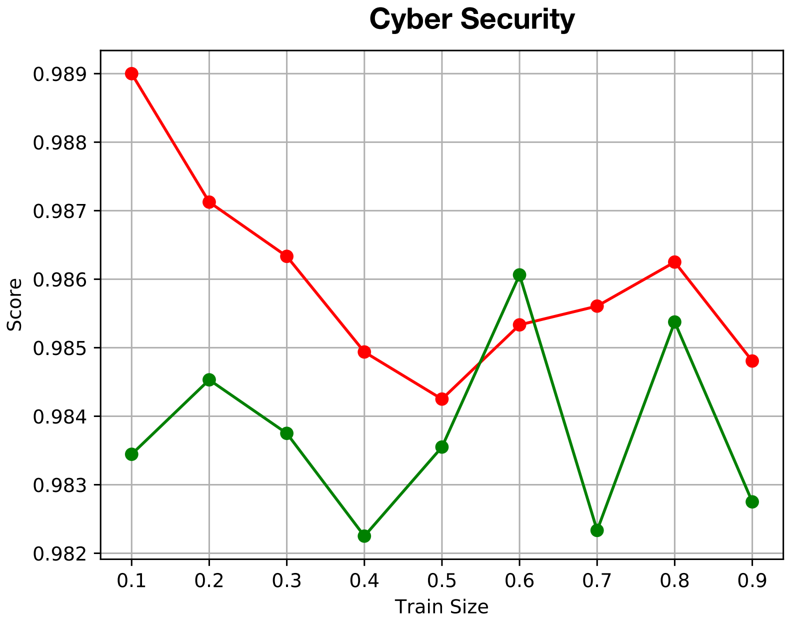


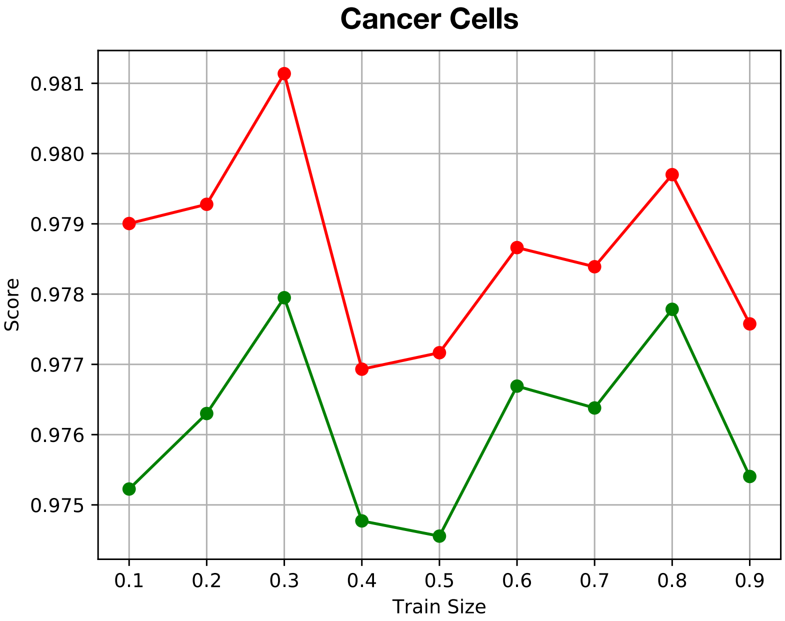
The SVM performs really well for gene expression data and is not time as time consuming as other algorithms, we can also experiment with different kinds of kernels to see our performance difference, but for the purpose of this assignment I only ran a linear SVM as instructed.

	Cyber Security	Cancer Cells
Train Size	0.8	0.8
Test Accuracy	**0.9165514580587082**	**0.9972222222222222**
Train Accuracy	**0.9169208634250049**	**1.0**

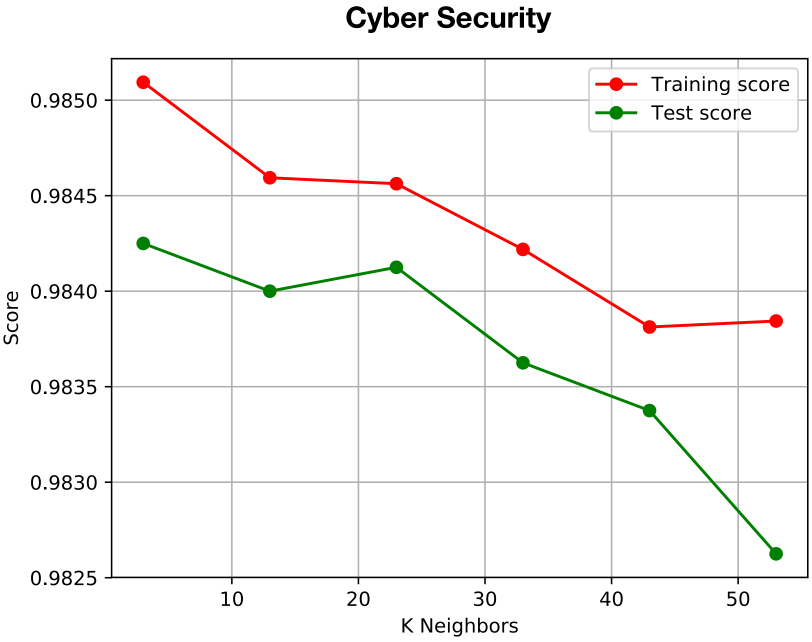
K Nearest Neighbor

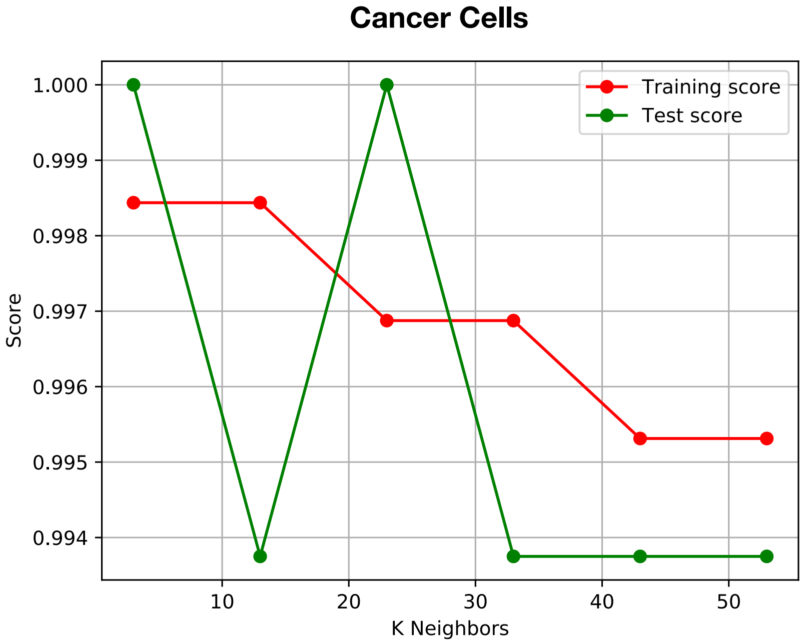
Knn is an instance based learning algorithm which is powerful but very time consuming, Knn has proved to be very strong in classifying Cancer Cell data (High Dimension) but did not seem to be performing significantly better than other algorithms in the case of cyber security dataset.


Cyber Security


Cancer Cells

We Then explore the effect of changing the K value and try to see the effects it has on the accuracy of the data.


Cyber Security

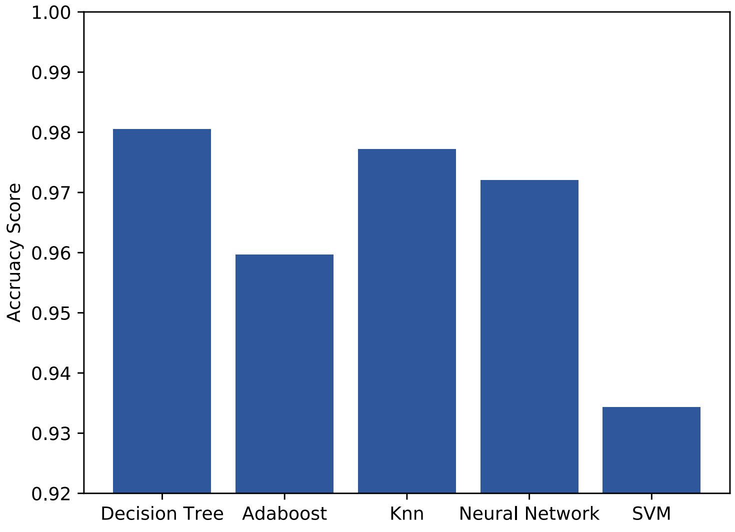

Cancer Cells

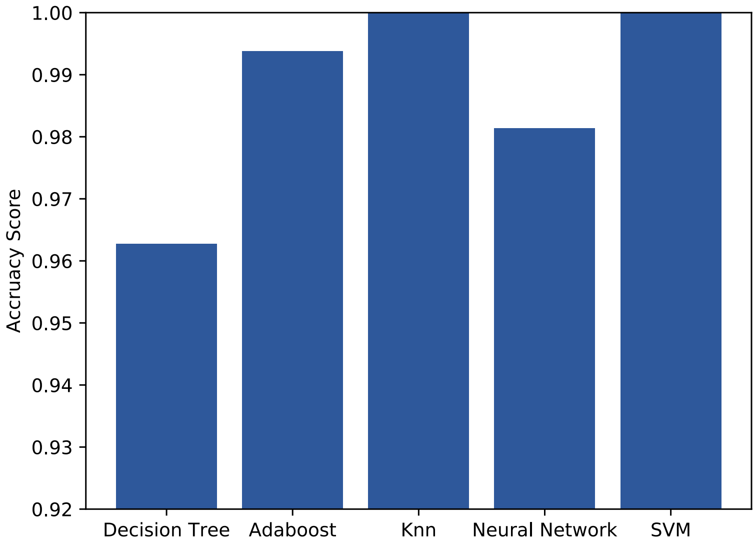
The above figures show in one case (Cyber Security) Increasing the K was causing the model to become less accurate, while in the case of the effect is not clear, This could be due to the high dimension of the data, however the model was always performing significantly well in the case of cancer cells. We could experiment further with the K count for the Cancer Cells Data set, however, Knn is extremely slow for large datasets, especially highly dimensional datasets such as cancer cell classification.

	Cyber Security	Cancer Cells
Train Size	0.8	0.8
Test Accuracy	**0.985375**	**0.9777834935536764**
Train Accuracy	**0.98625**	**0.9797004296226635**

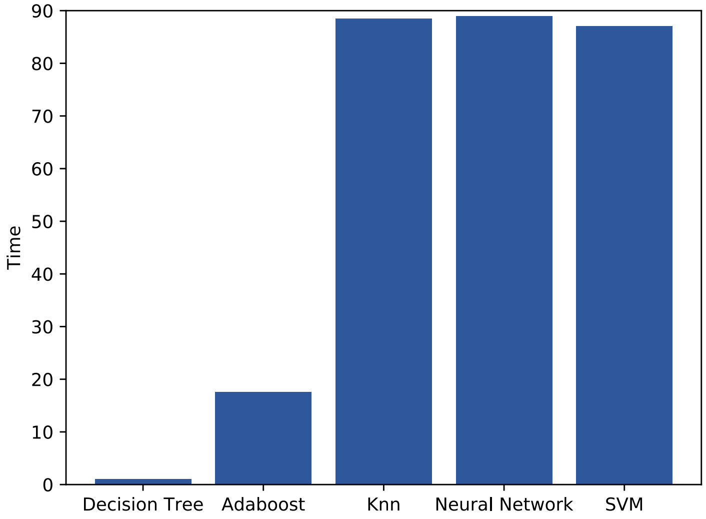
Conclusions And Comparisons:

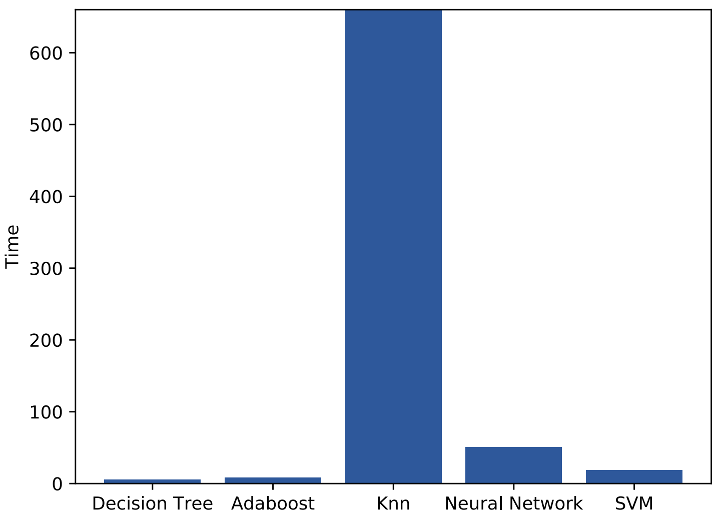
Now we compare the 5 algorithms discussed above on the two different datasets.
It appears that Decision Tree and Boosted Decision Tree works well for cyber security intrusions and that could be because the data is linearly separable which allows the Decision tree to perform well, on the other hand you can see that for Cancer Cells Classification more complex algorithms seem to perform significantly better such as Knn and SVM which have proved to work well with high dimensional data. If we combine some dimension reduction using PCA then run SVM for cancer cells we can create a more robust model.
NOTE: ***THERE IS INCONSISTENCY OF KNN BECAUSE IT TOOK HOURS TO RUN SEVERAL TIMES ON MY MACHINE, THE SET USED FOR KNN IS SLIGHTLY MINIFIED.***


Cyber Security


Cancer Cells

Knn and SVM are also performing similarly as they are better for non-linearly separable data which gene expression data is. Surprisingly I was anticipating the neural network to have done better at gene expression identification but Neural Networks are sensitive to noise and could have performed worse due to noise in data. Another thing we need to consider is the scalability of the algorithm based on time taken to train, this is important if we were to train on massive amounts of data and we do not have enough computational resources.


Cyber Security


Cancer Cells

In the case of cancer cells the Knn with 3 neighbors and 0.8 train size takes around 11 minutes however it performs pretty well. In addition SVM performs almost the same but takes less time.

To truly compare our best performing algorithms for each dataset we can get more data for cross validation, that is data that has not been used for training or testing and then see how well those algorithms perform. Then we would be able to make better decisions.
In the case of cyber security dataset it would be useful to train the model on more relevant datasets since the dataset used is outdated so extracting network packets to be used in training could be useful. For Cancer Cells getting gene expression data could be challenging since it requires patients to undergo tests and gene expression data is hard to get because it is usually private or for certain studies.

Now to narrow down our comparisons I selected The Decision Tree as the best classifier for Cyber Security and the SVM for Cancer Cells. This was not just based on accuracy it was based on time, precision and recall, and accuracy.

Cyber Security Decision Tree:

	Precision	Recall	F1-score
Attack	0.99	0.99	0.99
Normal	0.96	0.95	0.95

Cancer Cells Support Vector Machine:

	Precision	Recall	F1-score
BRCA	1	1	1
COAD	1	1	1
KIRC	1	1	1
LUAD	1	1	1
PRAD	1	1	1

In both situations the models highly precise and highly sensitive, in addition the two algorithms do not require much time to train.
To conclude, data with different characteristics and different sizes require different kinds of algorithms, multi dimensional and non linearly separable data appeared to be the most challenging to classify and could require further exploration using different machine learning techniques.

Parameters:

Algorithm	**Cyber Security**	**Cancer Cells**
Decision Tree	Max Depth = 30	Max Depth = 40
Neural Network	hidden_layer_sizes=(10,100) Max Iterations = 100	hidden_layer_sizes=(35, 1000) Max iterations = 80
Knn	K = 2	K = 9
Adaboost	N_estimator = 12	N_estimator = 6
SVM	C = 1.0	C = 1.0