Abhay Dalmia

February 4, 2018

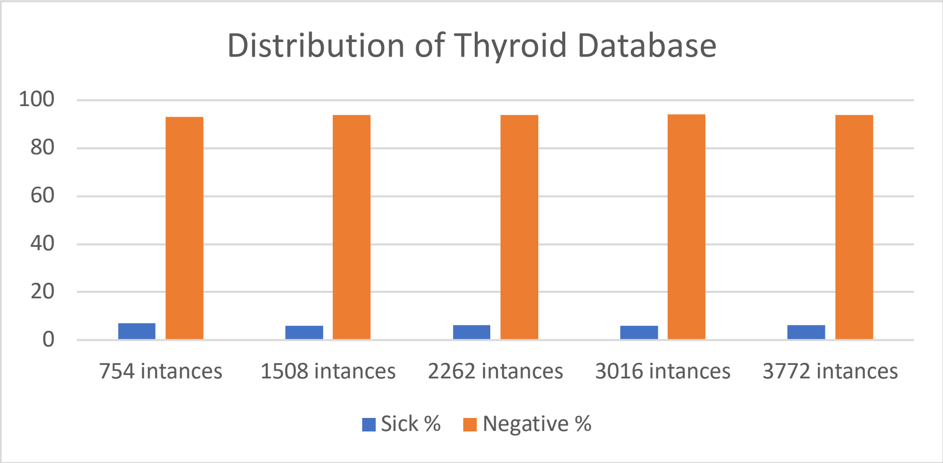
CS 4641 Assignment 1: Supervised Learning

Dataset Information

Thyroid Disease Dataset (Informally referred to as "sick" database)

This dataset uses 29 attributes in relation to hypothyroid and sick-euthyroid data to predict whether the patient has hypothyroid. The objective of this dataset then, is to be able to predict sickness (hypothyroidism) without performing the invasive surgery required. While this prediction would not be sufficient to replace the test altogether, it would provide guidance on whether the hypothyroid test is likely to be required.

The whole thyroid dataset contains 3772 instances. This data set is further split into 4 smaller datasets of different sizes, to analyze how behavior of the learning algorithms change with training set size. The distribution of the instances for each of the datasets is shown below.



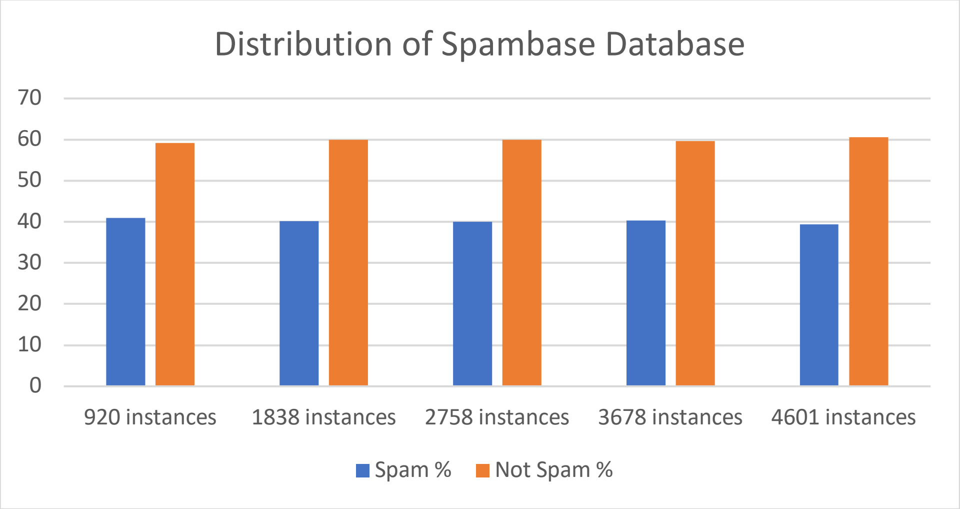
The distribution of the dataset is interesting because it has far fewer *sick* instances and opposed to *negative* instances (about 6% are sick and 94% are negative). This is likely to greatly affect the accuracy of the learning algorithms, due to the limited amount of information of one of the classifications. This will introduce an interesting factor of having incomplete or partial information, and evaluating model behavior due to this. This also resembles many other real datasets, because data is often limited and imperfect.

Spambase Dataset

This dataset uses 57 attributes to predict whether an email is considered "spam" or not. This is an increasingly important application, as email becomes an essential part of everyone's lives. With the hundreds of important emails that we get every week, it is important for email services to sort out the spam.

This dataset counts the frequency of word considered to appear in abnormal frequency in spam emails (like "credit" and "money") along with other important features, like the frequency of special characters and the frequency of capital words, to decide if the email is considered spam.

The whole spambase dataset contains 4601 instances that was split into 4smaller datasets, to analyze how behavior of the learning algorithms change with training set size. The distribution of the instances for each of the datasets is shown below.



The distribution of this dataset is more even compared to the thyroid one. About 40% of all instances are spam instances, and 60% are not spam. This is likely to help the accuracy of learning models (as opposed to a skewed distribution).

This dataset has a large number of attributes, compared to the number of instances. This might mean that not all attributes will be represented well by the instances, and learning models might choose to ignore some attributes completely, because their importance in determining the final state is not apparent in the dataset.

General Note on Analysis of Models

10-fold Cross Validation: For testing all models in this report, a 10-fold cross validation was used. The reason a 10-fold cross validation estimator was used is because it has much lower variance than just one test/train split. If a single test/train split is used, depending on how the data is split, it could “get lucky” and split the data in such a way that similar data is found in both the test and train set, hence increasing the accuracy. Or it could also get similarly “unlucky”. A 10-fold cross validation reduces the chances of great variation based on chance.

ROC Area: To evaluate models, the accuracy (% of correctly classified instances), sensitivity ability of a test to classify the instance under one of the classifications) and specificity (ability of a test to rule out an instance from a classification) is important. ROC curves plot sensitivity against specificity. A diagonal (with an area of 0.5) is considered to have “no power” and a prefect classifier (that goes up the y-axis) is considered “perfect” with an area of 1. Most classifiers lie in a range in between. To evaluate models, especially when accuracy is close enough to be insignificant (or rather within the margin of random error), the ROC curve is an important consideration. In this report, models are evaluated on how accurate they are, how long they took to train and their ROC area.

Test Strategy: To test all the different learning algorithms below, first, the effect of the learning parameters for each specific algorithm was evaluated. Once an optimum value was concluded on for the parameters, then the learning algorithm was evaluated on performance, based on differing training sizes (different number of instances). For the algorithms, the parameters that were optimized are:

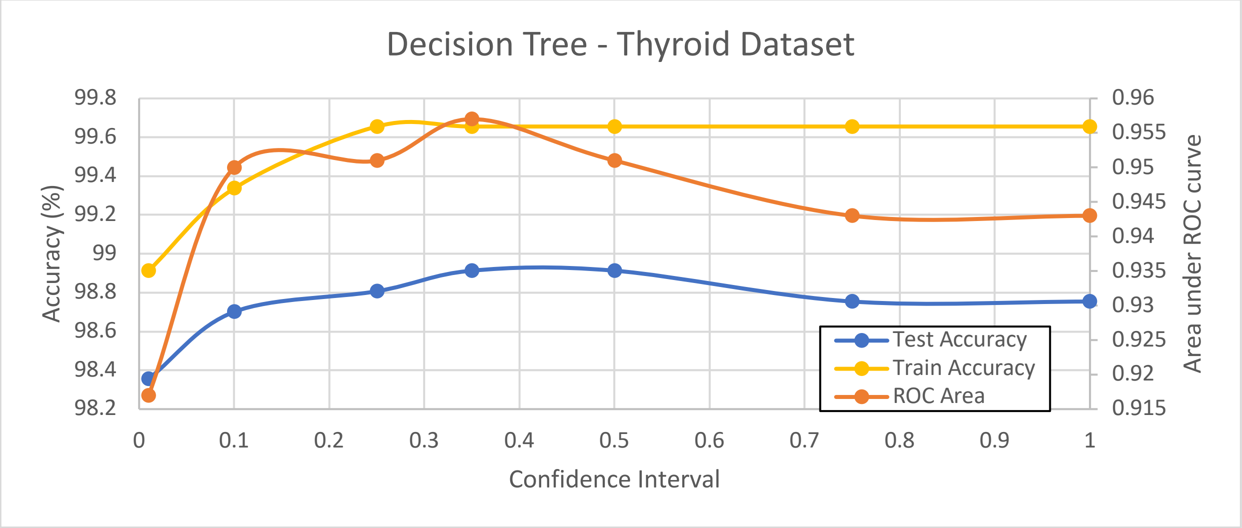
1. Decision trees: confidence interval for pruning (values between 0.01 to 1)
2. Boosting: confidence interval for pruning (values between 0.01 to 1) and number of iterations for boosting (values between 10 to 40)
3. kNN: weighted and unweighted kNNs with values of *k* between 1 and 40
4. SVM: 3 different kernels (linear, polynomial, RBF) with exponent values between 1 to 6 and gamma values between 0.0001 and 10
5. Neural net: learning rate (values between 0.01 to 1), momentum (values between 0.01 to 1) and number of perceptron layers (values between 1 and 30).

Decision Trees

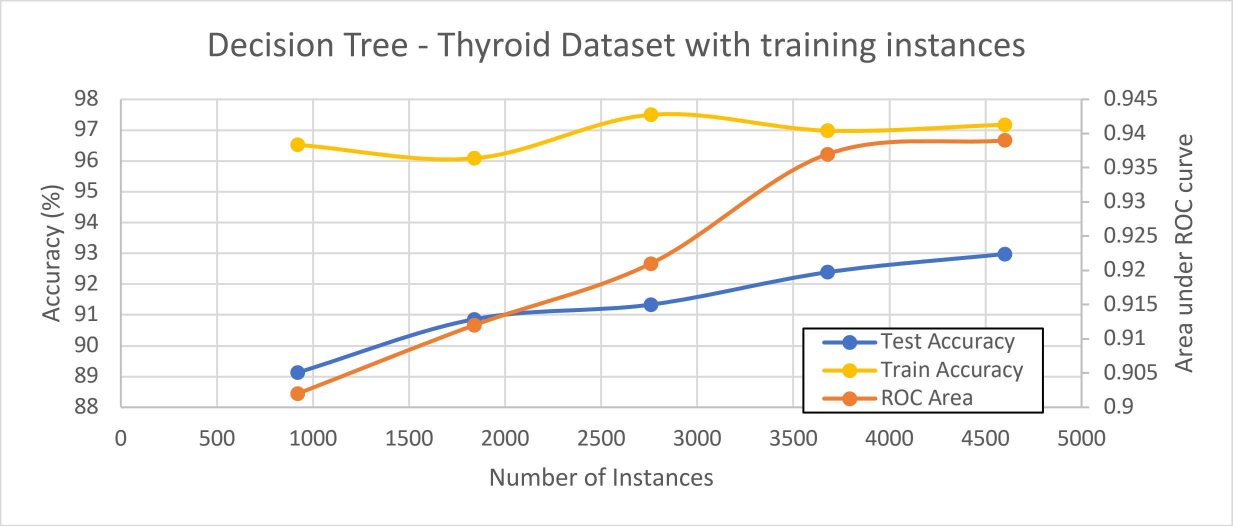
The decision tree algorithm used in this report is Weka's J48 algorithm, which is an implementation of the more popularly known C4.5 algorithm. It's similar to the ID3 algorithm, in that it uses information gain to decide on which attribute to split the data on next and hence, determines the structure of the tree. The algorithm also prunes the data by confidence interval (the lower the confidence interval the *larger* the pruning). This algorithm uses subtree raising (as opposed to subtree replacement), where a node may be moved upwards towards the root, replacing other nodes along the way. Subtree raising often doesn't have a clear way of predicating the utility of the option. Hence, multiple different values were investigated in this report to conclude on an optimal value.

Thyroid Dataset

From the results of the model at different confidence intervals, it can be concluded that the optimum confidence interval for this dataset and model is 0.35. Before 0.35, we see that the training and testing accuracy, and ROC area are lower. This is because the tree is over-pruned. The lower confidence interval lost too much data in the pruning stage. We can note this by the training accuracy increasing at higher confidence intervals. The higher confidence intervals (above 0.35) also dip in accuracy, due to overfitting. This is supported by the consistently high training accuracy but lower testing accuracy.



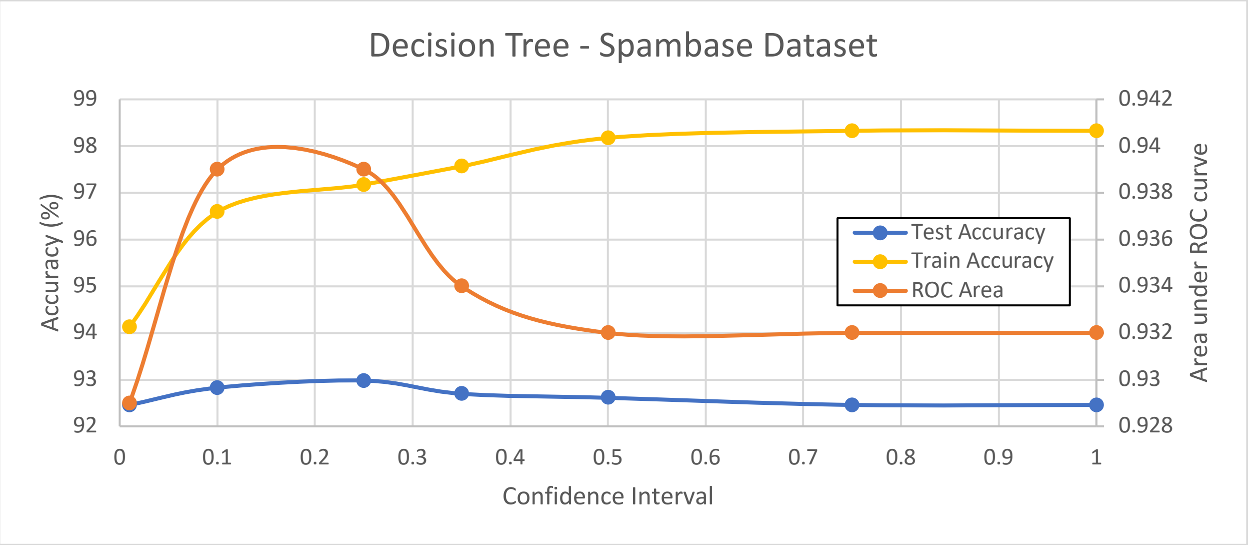
The confusion matrix for this dataset also shows that a much greater number of sick patients were misclassified as not sick as opposed to not sick patients misclassified as sick patients. This can be attributed to the fact that, as seen in **Figure 1**, there are far fewer sick instances to train on. Hence, the decision tree to conclude sick patients is not specific enough.



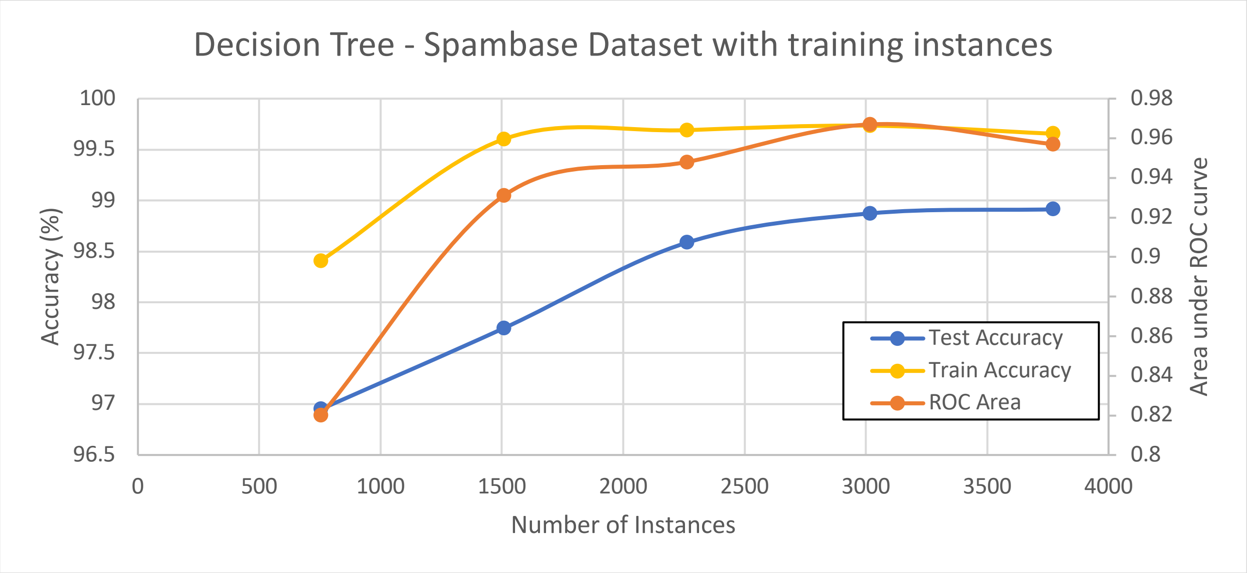
With the optimum confidence interval from the previous results (C=0.35), the model was trained with 5 different size training samples. As expected, the accuracies and sensitivities increased with a greater number of samples. With more samples, the decision tree is better able to decide which attributes are important (with information gain). Additional, the tree sizes are also larger (even with the same amount of pruning), with 754 instances having 18 nodes (11 leaves) and 3772 instances having 61 nodes (34 leaves). The model is more precise and separates on more attributes.

Spambase Dataset

A similar analysis was carried out for the spambase dataset. The testing accuracy and ROC, both peaked at C=0.25 for this dataset (as opposed to 0.35). This might be explained by the fact that this dataset has almost double the number of attributes as compared to the first one, but about the same number of instances. Not all the attributes then can be represented fairly by the dataset, and a greater amount of pruning is required for accurate results. Similar to the thyroid database, below C=0.25, the model is pruned too greatly, so important information is lost (and training accuracy is low). With higher confidence intervals, the model is insufficiently pruned and is over fit to the data (the consistently high training rates supports this conclusion).



The model behavior with different number of instances is similar to that of the thyroid dataset. The accuracies and sensitivities (ROC area) increase with large number of instances. Interestingly, the gain in accuracy is much greater initially, and levels off at around 4000 instances. This may mean that any larger number than this may not lead to any more significant improvements in the model. This could suggest that the decision tree trained on the 4000 instances is as accurate as it could get. Anymore data would not add much more information gain to the tree.



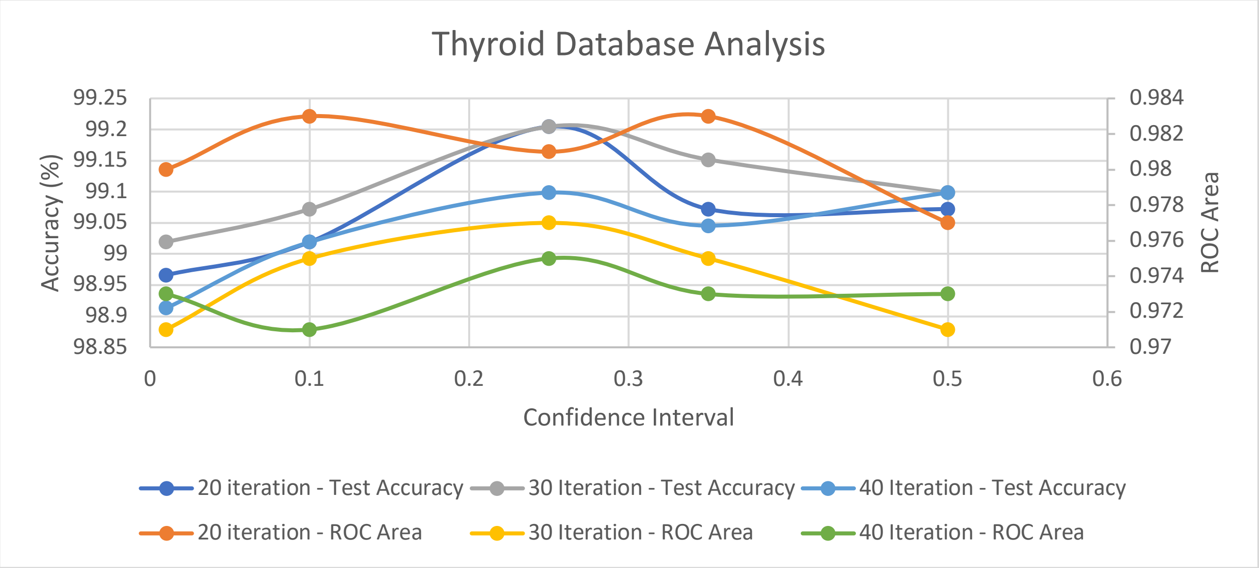
Boosting

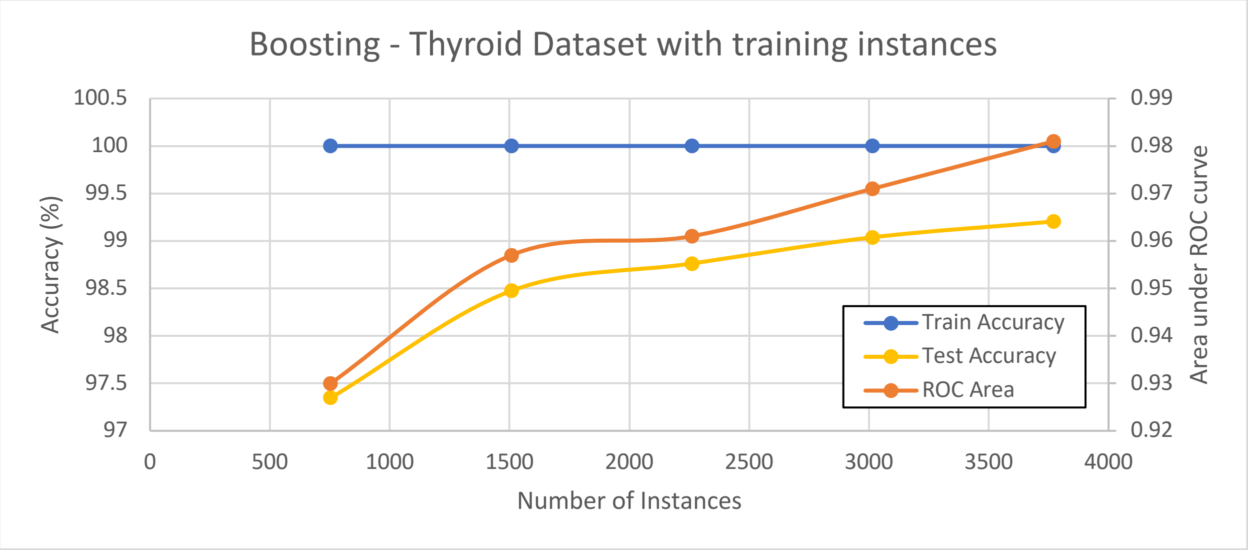
The algorithm used was an AdaBoost algorithm to boost the decision tree, based on the J48 algorithm (same as the decision tree above). Boosting combines several weak learners to one strong learner. Given how boosting works, the pruning can probably be more aggressive (lower confidence interval).

Thyroid Database

The boosting was conducted on 10-40 iterations with intervals of 10. Based on the chart below and the test accuracy and ROC area, the confidence interval that seems to work best with boosting is C=0.25. This is lower than the non-boosted version (C=0.35 from before) which is as expected. The optimum number of iterations can also be concluded to be around 20 iterations. On the chart, these values lead to the highest accuracies and ROC areas.

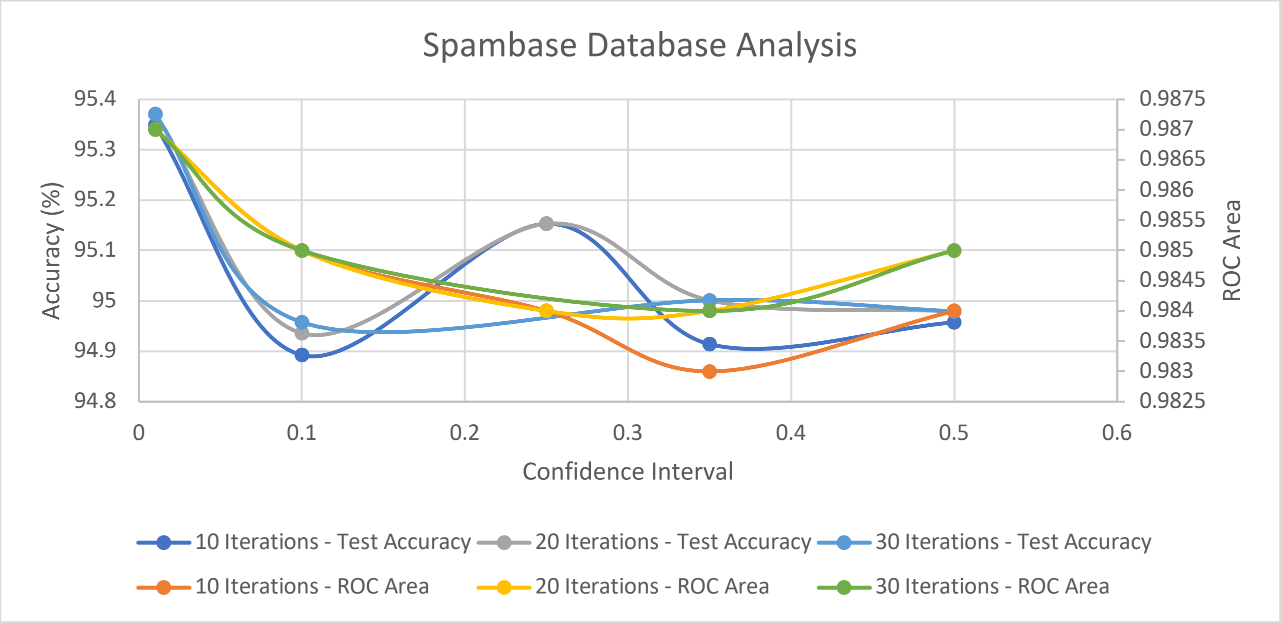
With the optimal confidence interval (C=0.25) and number of iterations (20 iterations), the model was evaluated on an increasing number of instances. In similar fashion to the decision trees, accuracies and sensitivities increased across the board. The reasons are similar to the decision trees and doesn’t warrant further discussion.



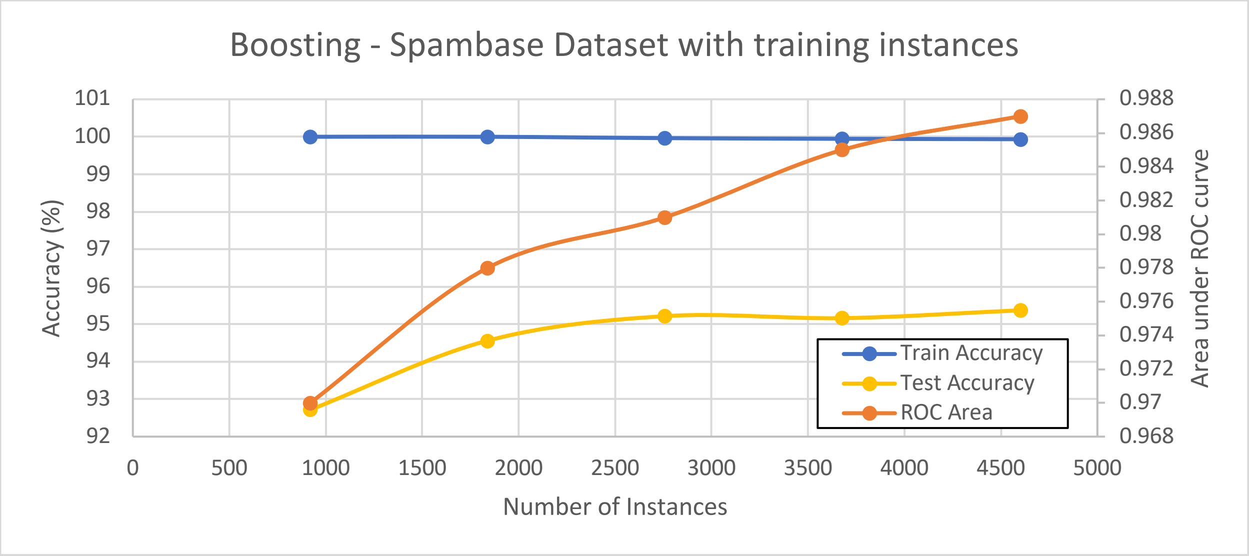


Spambase Database

The spambase dataset was treated similar to the thyroid database. The confidence interval for this decreased by a much greater amount compared to the non-boosted version, from C = 0.25, to C=0.01 for this case. This can be attributed to the fact that the dataset has many attributes with disproportionately few instances. Hence, when weaker models are constructed, only few attributes are needed to satisfy a weak model, and multiple of these weak models (with few strong attributes), combine to form a fairly strong decision tree. For this dataset as well, 20 iterations was found to result in the highest test accuracy and ROC area (grey and blue lines in the chart below).



With the optimal confidence interval, C = 0.01, and number of iterations, 20, the model was evaluated on an increasing number of iterations. It performed as expected of a decision tree and like the other dataset (already analyzed previously).

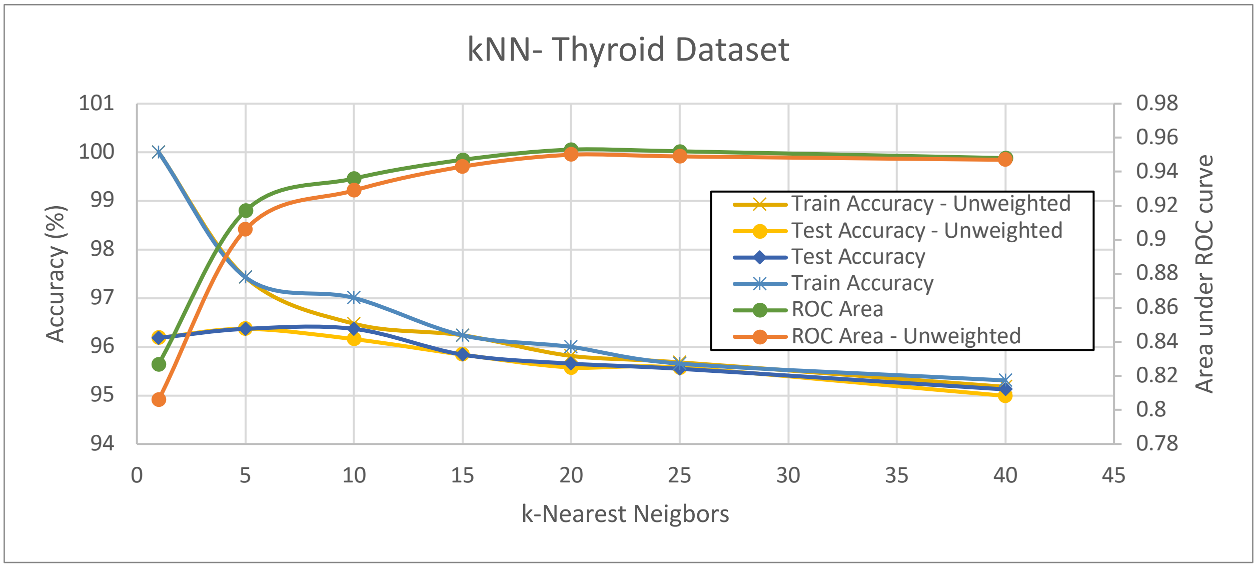


k-Nearest Neighbors

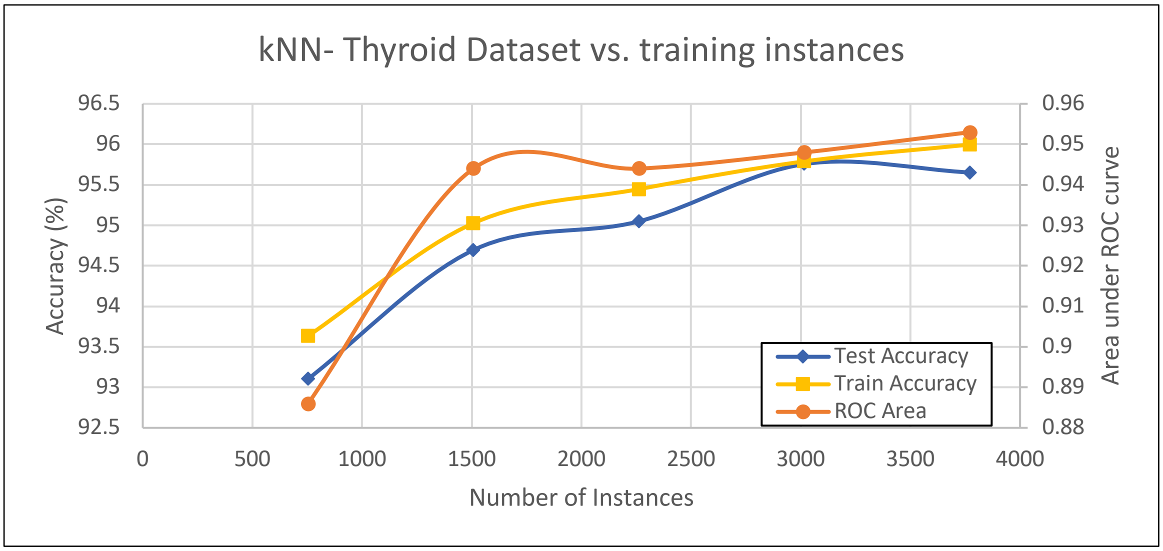
The algorithm used for both datasets was a linear nearest neighbor search, by varying neighbors between 1 to 40.

Thyroid Database

From the chart below, by changing weighted status and values of k, it can be seen that across the board, the weighted model performs better. The increasing values of k result in a mixed response. While the test accuracy decreases with increasing k (becomes less accurate), the ROC area increases (sensitivity and specificity). With the kNN algorithm, the model considers increasing number of neighbors and performs a voting-system like algorithm to decide the classification of a point. Hence, with increasing k, it is expected that testing accuracy would decrease, as more neighbors (who don't necessarily know anything about the point itself), are allowed to vote on its status. But because of the increased divergence of the algorithm in considering a greater number of viewpoints, the sensitivity (and by extension ROC area) increases. So, this is as expected. It can be estimated that a weighted **k=20** would work best, as it leads to a great increase in ROC area with marginal decrease in testing accuracy.

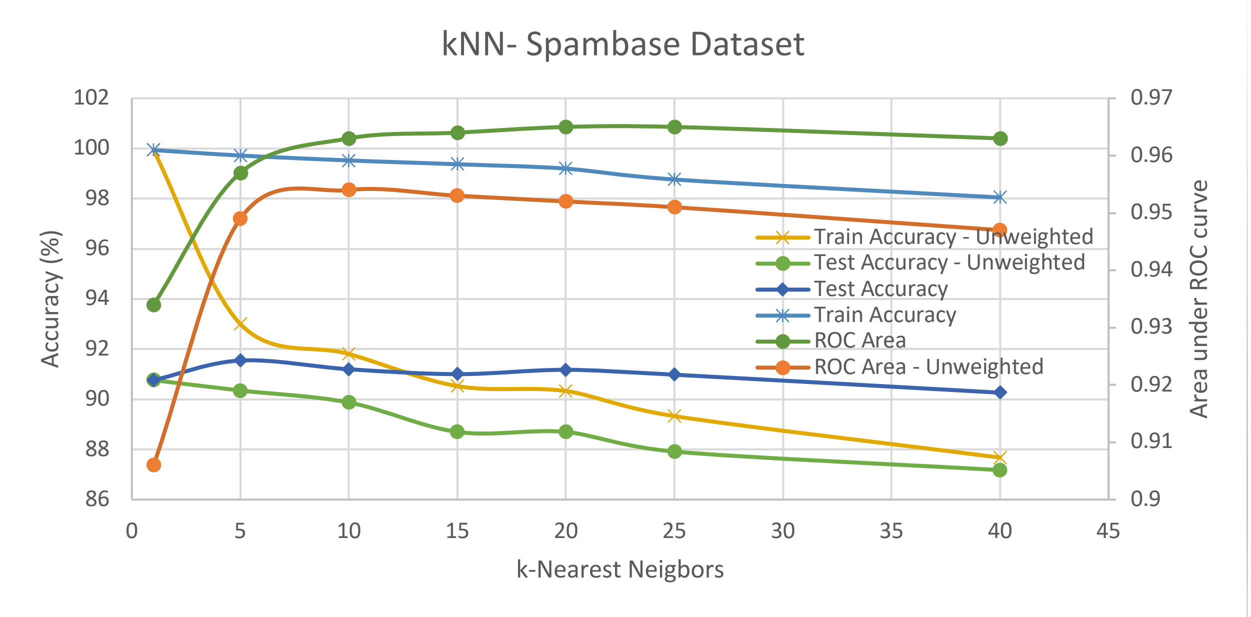


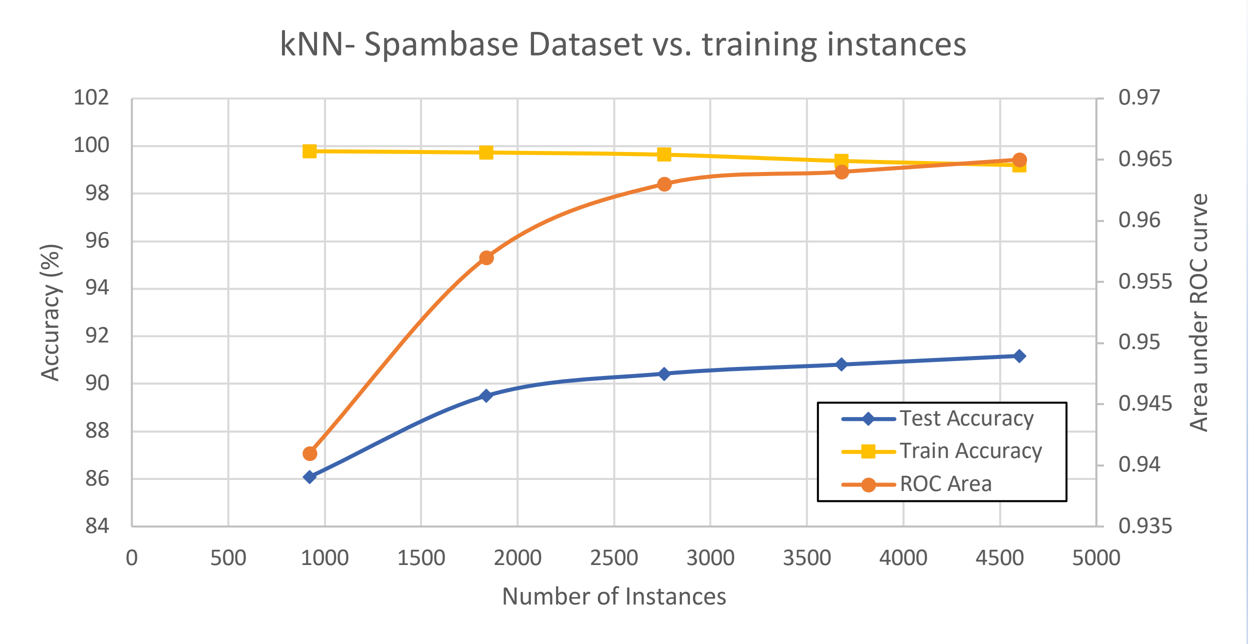
These optimal specifications were used to evaluate the model against an increasing number of instances. As expected from previous results, with a greater amount of information to train on, the model performs better. What was interesting was the slight decrease of test accuracy for the maximum number of instances. This can probably be attributed to the addition of noisier data in the last test split created. (Proven later to indeed be the case).



Spambase Dataset

The results were almost congruent for the spambase dataset as compared to the thyroid dataset. The ROC area and test accuracy and weighted status behaved in the same manner with increasing k. Given this behavior, **k=20** was again chosen to be the best value, based on a large increase in ROC area with a marginal decrease in test accuracy. With these values, the model was again evaluated on increasing number of instances. As expected, the ROC area and test accuracy increased with more instances. The slight decrease with more instances noted in the thyroid database did not make an appearance here.





Neural Networks

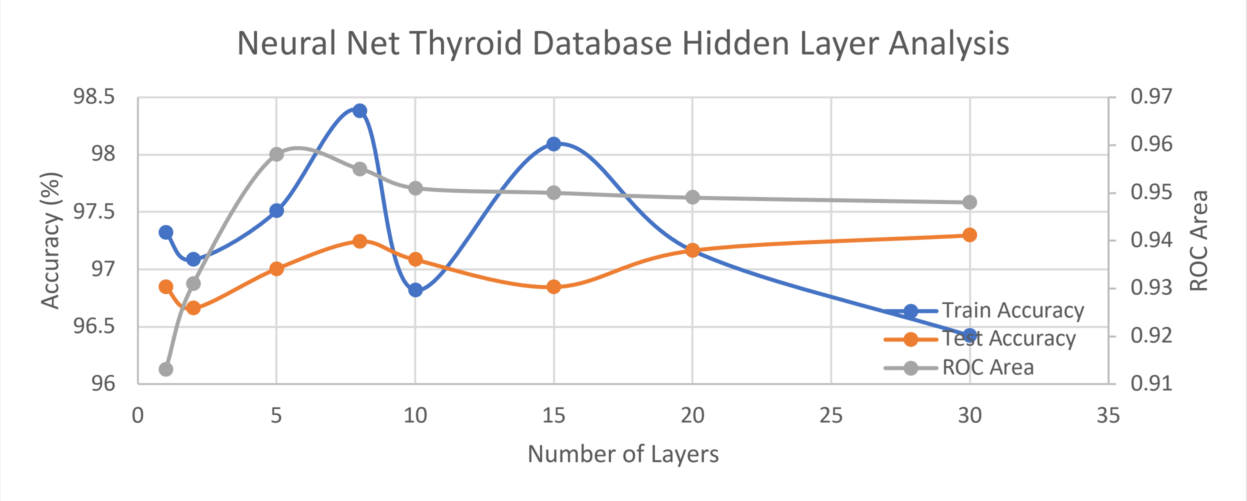
Both datasets were modeled with a multilayer perceptron neural network algorithm, with changing parameters like the number of hidden layers, the learning rate and the momentum. After an optimum value was concluded on for each of these parameters, the model was evaluated against increasing training model size.

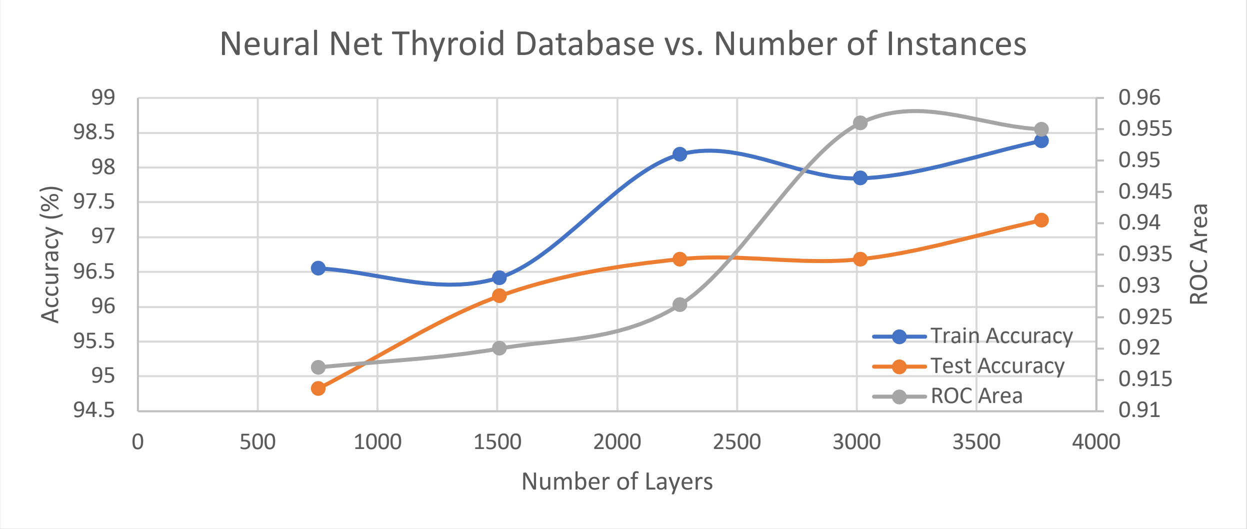
Thyroid Database

First, the effect of number of hidden layers was investigated on the thyroid database. Given the complexity of a neural net, the results weren't very surprising. The test accuracy seemed to peak at 8 hidden layers before dipping to a low, and then increasing again to another peak. But the ROC area also peaked at around 8 hidden layers and decreased consistently after. Hence, even though the model was more accurate with >40 hidden layers, it wasn't as sufficiently sensitive. Additionally, a 8 hidden layer model was much quicker to train (by a factor of 10) as compared to a 40 hidden layer model. Hence, **8 hidden layers was chosen as the optimal value**.

The learning rate and momentum were similarly iterated on and the optimal value chosen was L = 0.3 and M = 0.2. (Complete details in excel file). The momentum, when too small or too large, lead to decreased test accuracies, due to the model wither undershooting or overshooting.

With these optimized values, the model was evaluated on different number of instances. It improved with more instances, as the neural net passed more data through the hidden layers and its backpropagation/gradient descent algorithm, and evaluated more accurate and precise weights for the nodes.

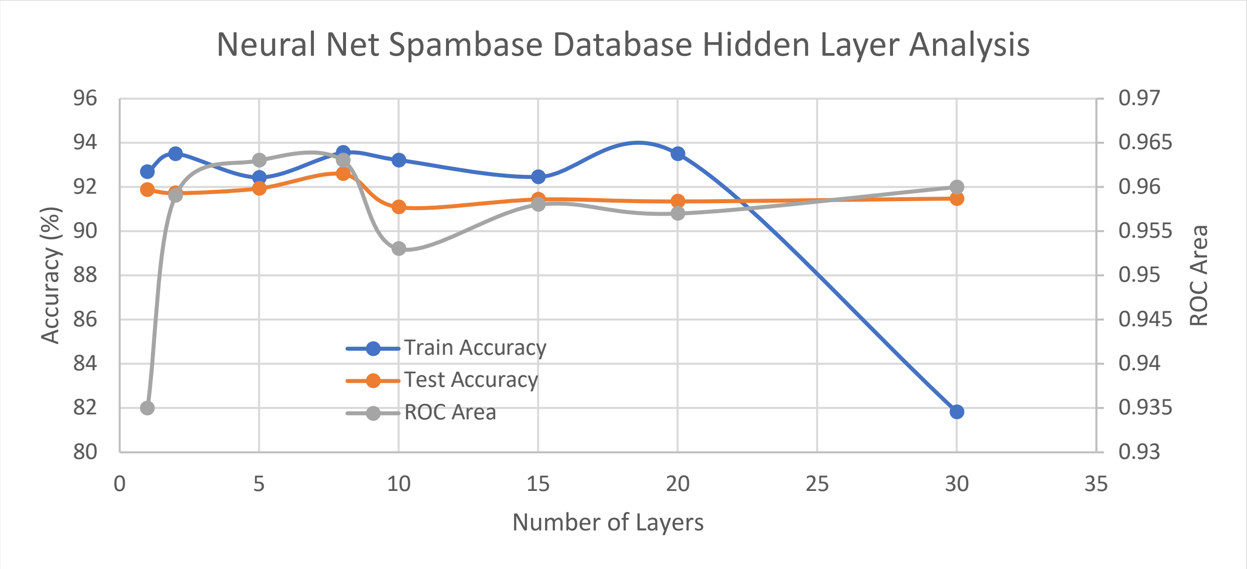


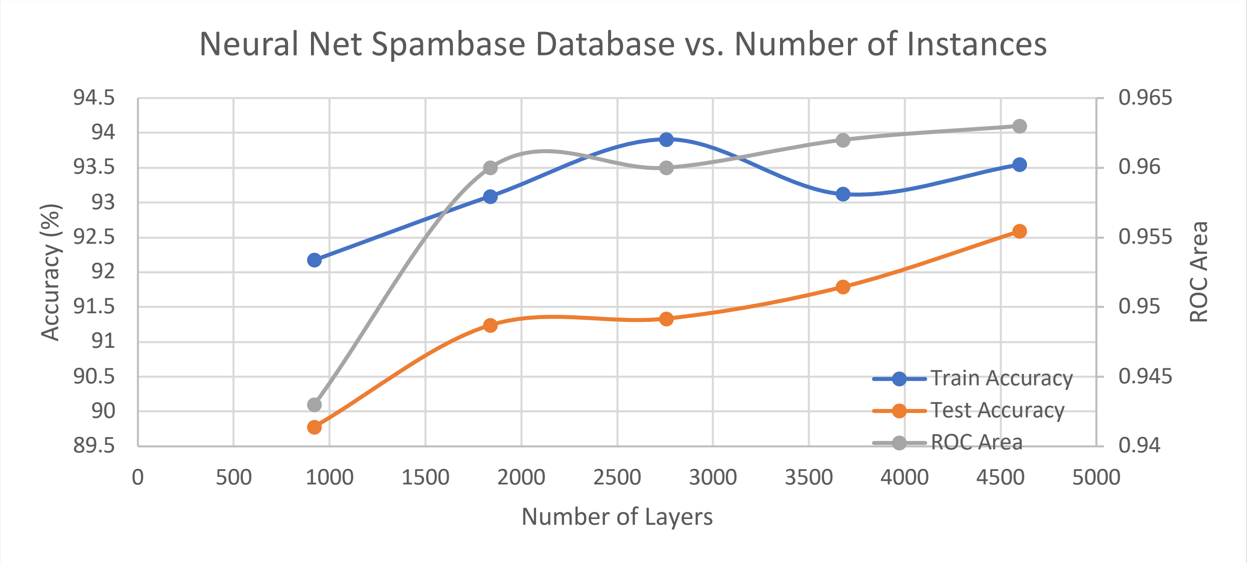


Spambase Dataset

The spambase dataset behaved in similar fashion to the thyroid dataset. However, in this case, the test accuracy and ROC area clearly peaked at 8 hidden layers and did not improve after. This can be attributed to the increased complexity of this dataset, with twice as many attributes, as compared to the thyroid dataset. Hence, with many hidden layers, the model becomes unnecessarily complex, and there aren't enough instances to accurately solve for all the node weights in all the increased number of layers. So, in this case, **8 hidden layers** was the clear winner, with a model complex enough to not under fit the data, but not too complex to risk overfitting.

As before, the learning rate and momentum were iterated over (full details in excel file) to arrive at the optimum values, which were also L=0.3, M=0.2. As with the thyroid dataset, it behaved similarly for the same reasons to an increasing number of instances (backpropagation explained before).



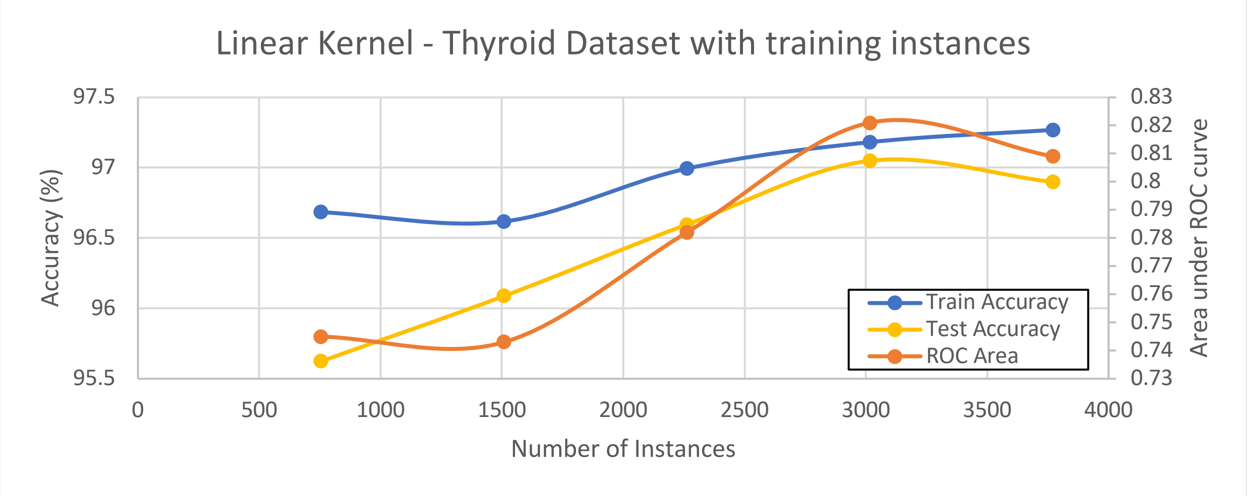


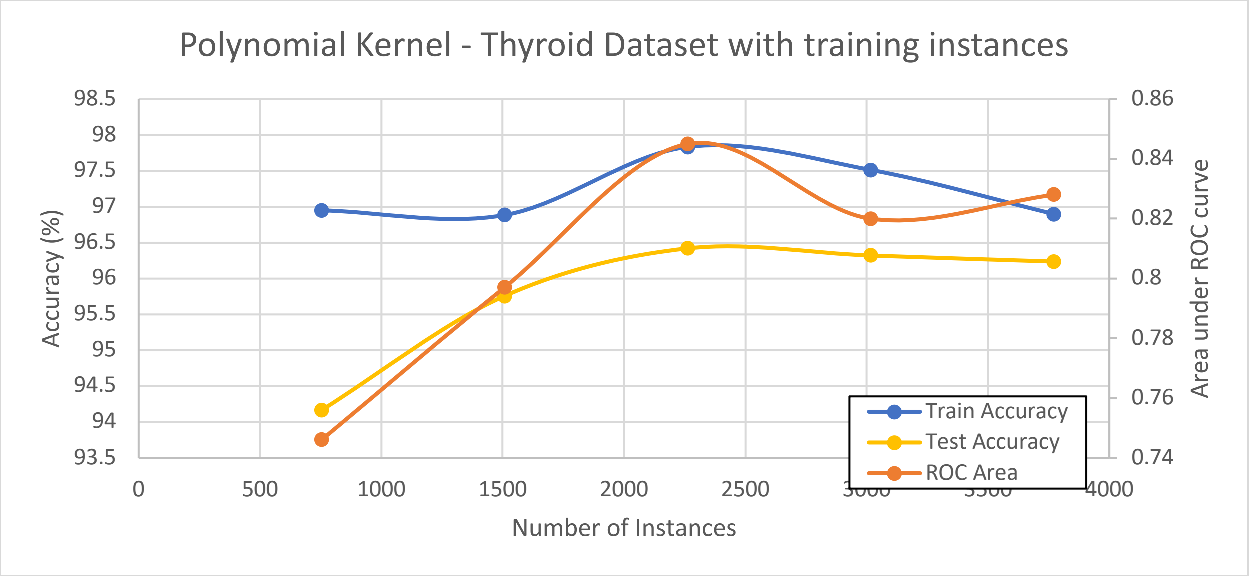
Support Vector Machines

In general, due to the application of SVMs regularization, they tend to resist overfitting. However, if the kernel does not fit the type of data, random and inconsistent results are observed due to the inherent nature of the kernel itself.

Thyroid Dataset

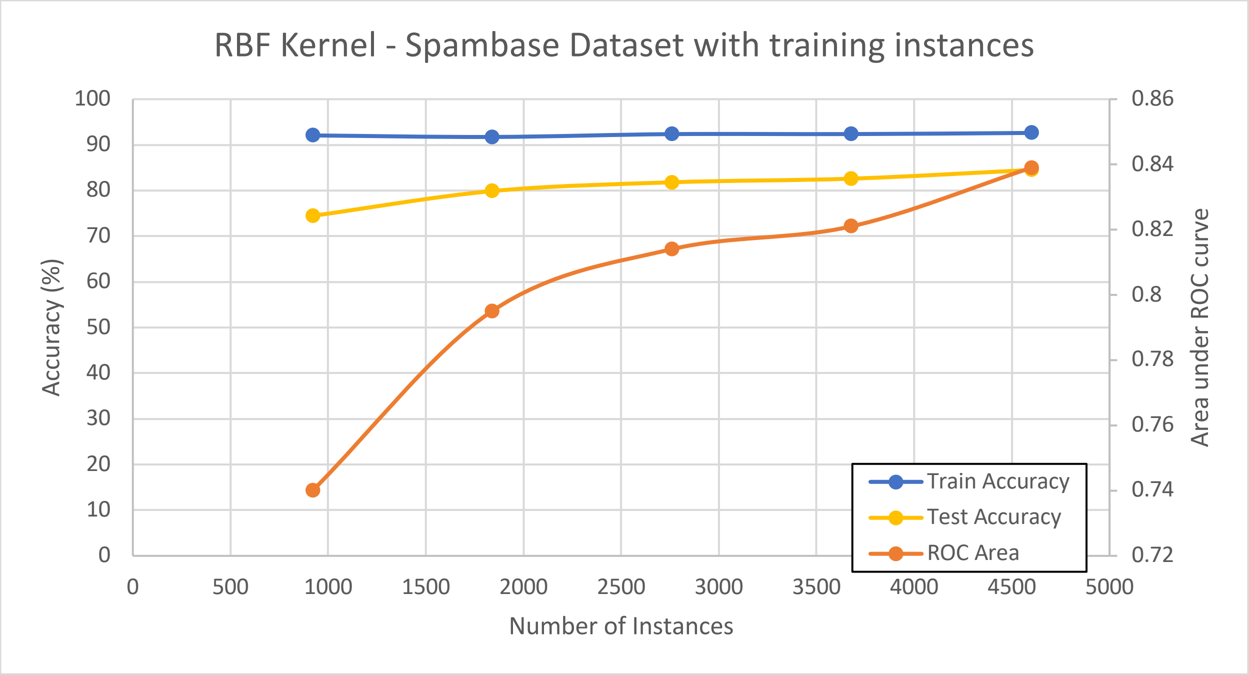
The dataset was trained using the linear kernel first. It seemed to perform quite well and as expected. However, there was a dip in testing accuracy and ROC area for the largest number of instances, while testing accuracy still increased. Given these parameters, it can be concluded that overfitting likely occurred. This is the first instance, where too much data clearly led to lower accuracies in a model. The same trends are observed in the polynomial kernel, with degree 2. Degrees 1 through 6 were iterated over before optimizing it to the 2nd degree (full data and details in excel file). Even with the polynomial function, with too much data, overfitting occurs. This seems to suggest that neither might be the best fit for this application.





Spambase Dataset

The spambase dataset was evaluated using the same linear and polynomial kernels as above but behaved in exactly the same manner, and hence is left out of this discussion (full data details and charts are available in the excel file though). The. RBF Kernel implements several improvements to other kernels, because SVMs don't generally behave well with a large feature space or a large number of instances. This can be observed in the chart below. Even with the maximum number of instances (and a fairly complex dataset with >50 attributes), the test accuracy and ROC area increase consistently (unlike the linear and polynomial kernels seen above). Hence, given that both datasets are fairly complex and have a large number of attributes, the RBFKernel could be the most applicable for an accurate model.


RBF Kernel - Spambase Dataset with training instances
Accuracy (%)
Area under ROC curve
Number of Instances
Train Accuracy
Test Accuracy
ROC Area