Seng 474 Assignment 1 Report

Shayla Grymaloski
V00884262

Second Classification Problem:

Description of dataset

This dataset is using measurements from recorded voices in order to determine if the voice is male or female. There are 20 different values of acoustic properties that were recorded for each voice. Of these properties' values, the datasets last column notes whether the voice is male or female based on the acoustic properties of the voice recording. The full contents of this data can be found here: https://www.kaggle.com/primaryobjects/voicegender

Why is this a good comparison?

This dataset is a good comparison to use because it is a binary classification task where there are only two categories for the data to fall under, male and female. The other data set that we were provided is also a binary classification task so it is good to have both of these datasets used for this assignment.

What makes this data interesting?

This data is interesting for us to consider when developing and improving voice recognition software. Voice recognition software is essential in devices such as google home or alexa. Being able to successfully and correctly analyze different voices is also key for voice recognition software that turns recordings into text. This type of software is extremely important to have to increase accessibility to technology. The better accessibility software like voice recognition software, the better user experience of technology for those with disabilities.

Implementation:

This section will explore different possible parameter values using decision trees, random forests and neural networks to classify the heart data that was given for this assignment as well as the data set that I chose to observe. The graphs of this section plot the training accuracy (in blue) and the test accuracy (in red).

Decision Trees (with pruning)

Exploring test size

I chose to look at the different possible test sizes that can be given to a decision tree. I looked at the test sizes ranging from 0.1 to 0.9 to find which test size produced the most accurate result test result.

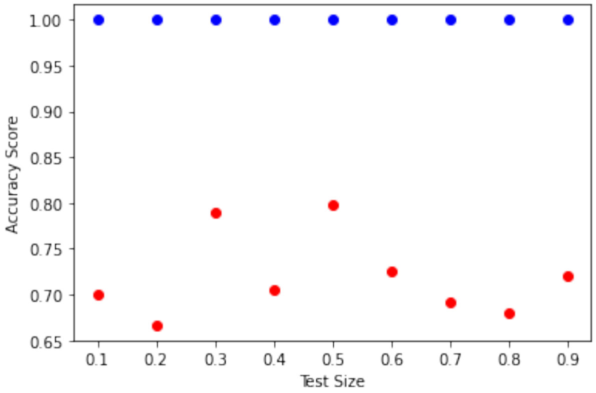

Figure 1, Dataset 1

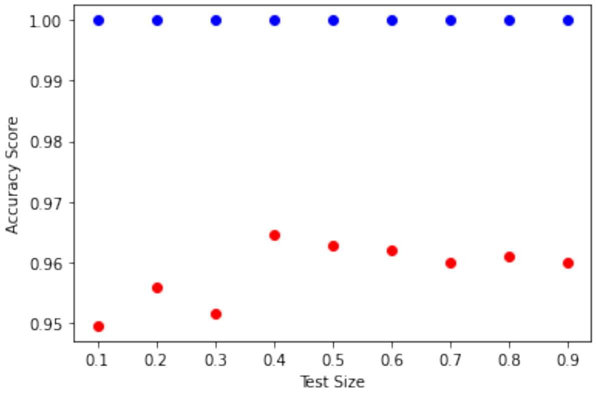

Figure 2, Dataset 2

Exploring Max Depth

Next, I looked at the value of max depth in the decision tree which controls longest path in a tree from the root to leaf node. Note, this value should never be 0.

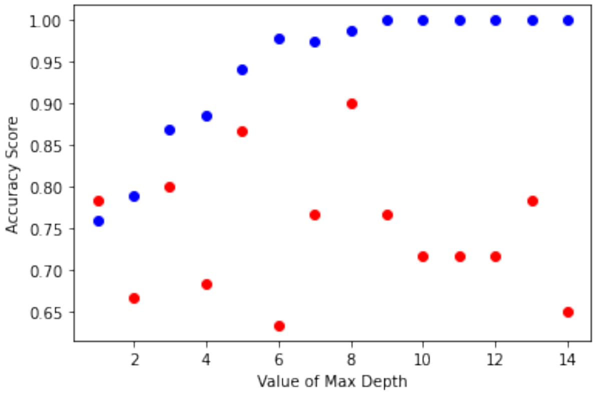

Figure 3, Dataset 1

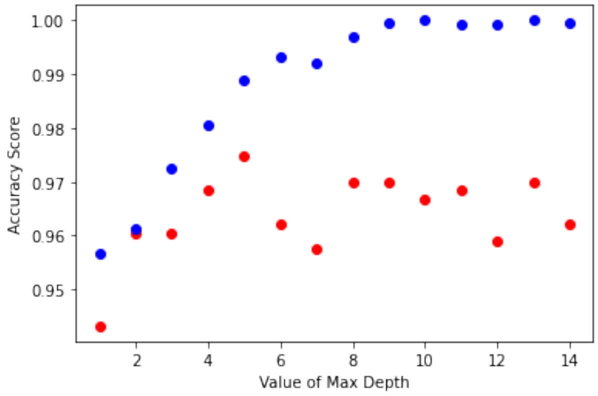

Figure 4, Dataset 2

Exploring Random State

This experiment also explored changing the random state of the decision tree.

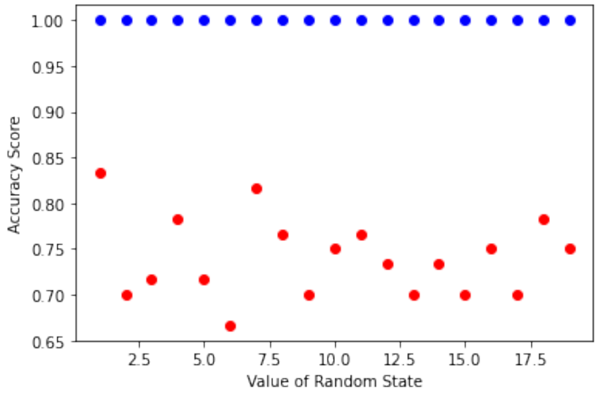

Figure 5, Dataset 1

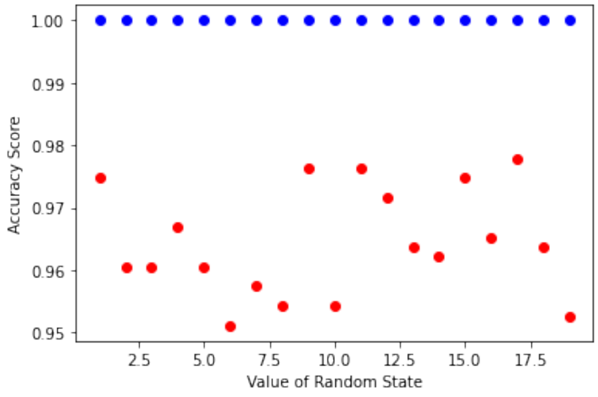

Figure 6, Dataset 2

Exploring Gini vs Entropy

In this experiment, I looked at comparing the effects of entropy vs gini. I used the experiment for exploring the best test size for this part. By default, the criterion will be set to 'gini' so I plotted that test value in red and then rant the same test value with criterion set to 'entropy' and plotted that in green.

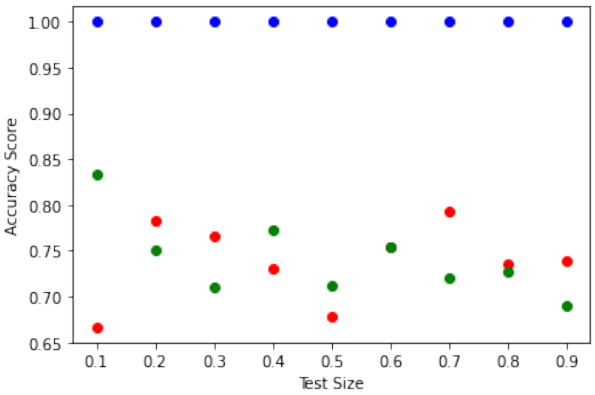

Figure 7, Dataset 1

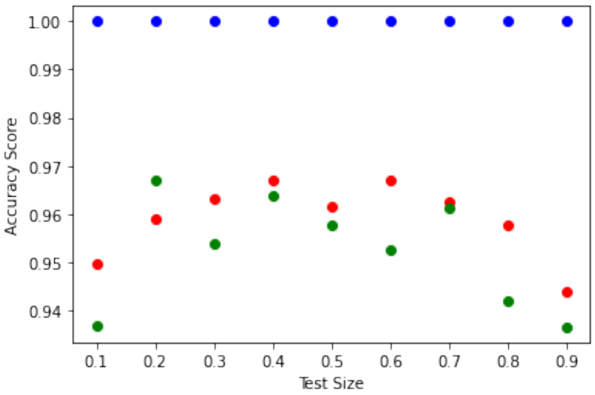

Figure 8, Dataset 2

Exploring Max Leaf Nodes

This experiment looked at what happened to the accuracy as the number of max leaf nodes were increased. I chose a range for 2 to 50 incrementing by 5 which allowed me to see when the training data would hit 100 percent accuracy. Normally the max leaf Nodes is set to infinity.

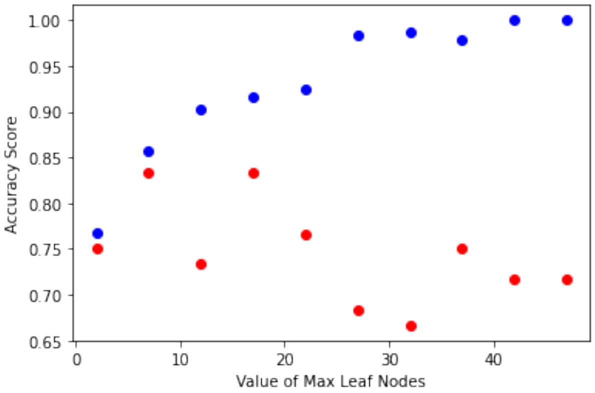

Figure 11, Dataset 1

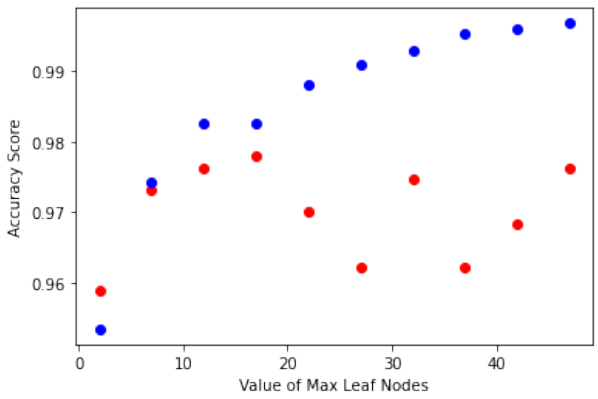

Figure 12, Dataset 2

Random Forests (no pruning):

Exploring N Estimators

In this experiment, I explored changing the value of N estimators or the number of trees in the random forest. Since the default for this parameter is 100, I chose to rank the parameter value in the range of 1 to 200 to see which value of n estimators gave the most accurate test result.

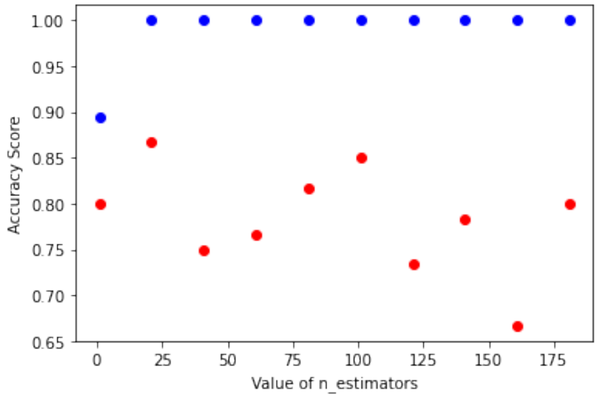

Figure 13, Dataset 1

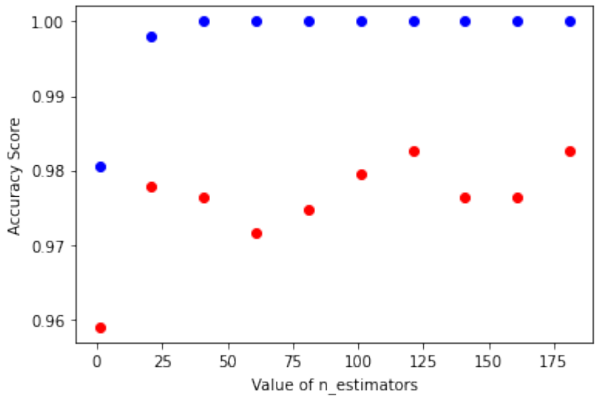

Figure 14, Dataset 2

Exploring Max Features

This experiment looked at the parameter of max features. I had the data run from 1 as the max feature to the number of features of the dataset.

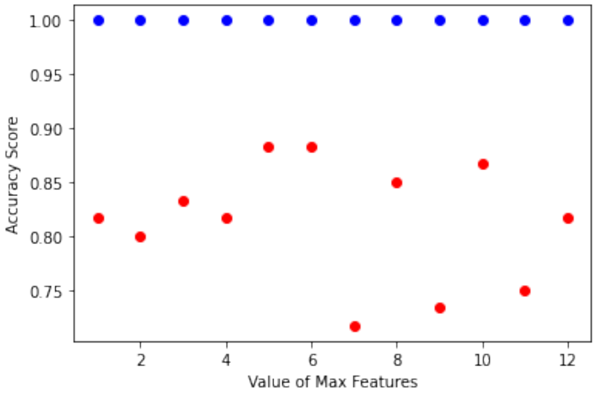

Figure 15, Dataset 1

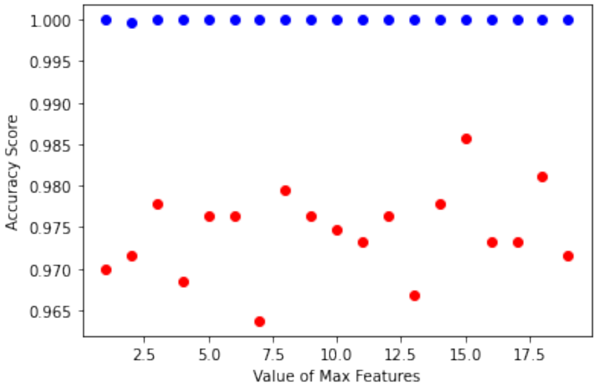

Figure 16, Dataset 2

Exploring Minimum Sample Split

This experiment explores the minimum number of samples needed in order for a node to split. I passed a float through so the value of minimum sample split will be the ceiling of the number of samples times the floating value passed. The default parameter is 2 so I wanted to test if I could get a value around 2 to be the most accurate point when passing a float from 0.1 to 0.9.

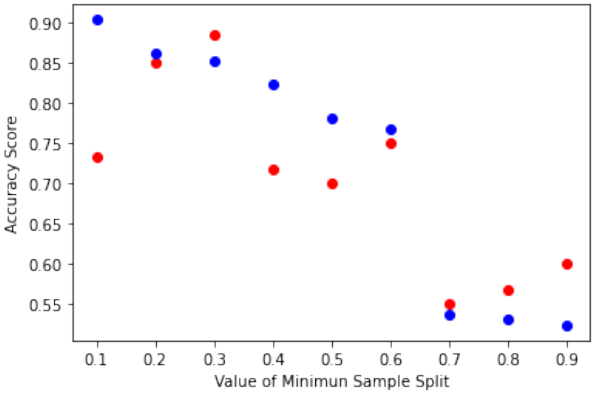

Figure 17, Dataset 1

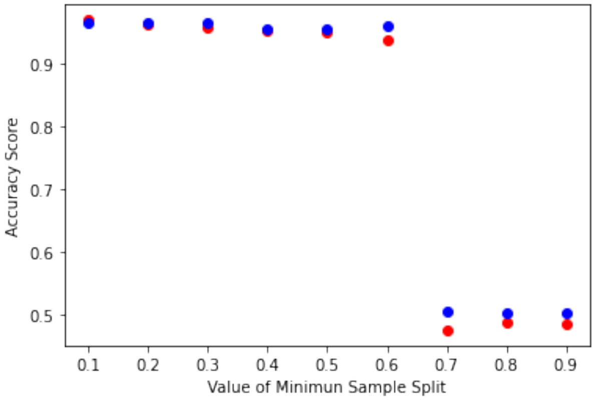

Figure 18, Dataset 2

Neural Networks:

Exploring Alpha

In this experiment, I looked at changing the value of alpha in the neural network to see the most accurate test results. When picking the range of data I chose to go from 0.00001 to 0.001 so that the default alpha value, 0.0001 was in the middle of the graph.

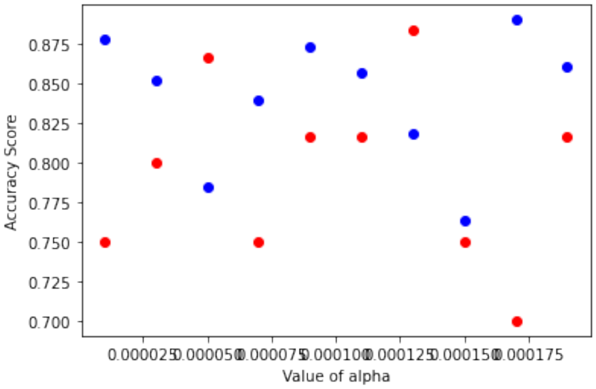

Figure 19, Dataset 1

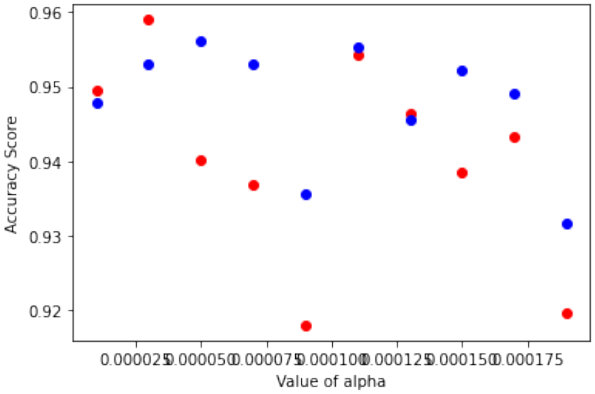

Figure 20, Dataset 2

Exploring Initial rate, Constant learning

In this section, I looked at changing the learning rate of the neural network with constant learning as the rate. I knew the default learning rate was 0.0001 so I wanted to have a range of values around 0.0001, in this case 0.0015 to 0.0005.

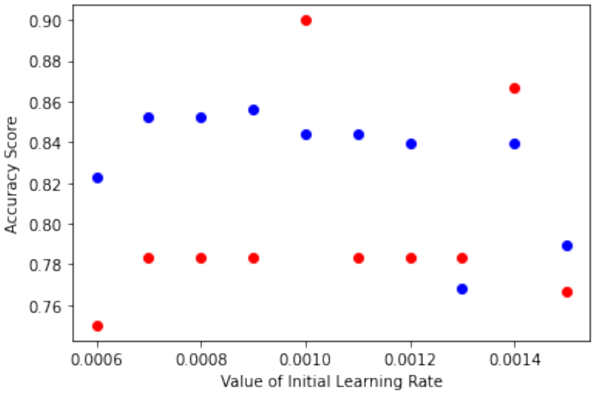

Figure 21, Dataset 1

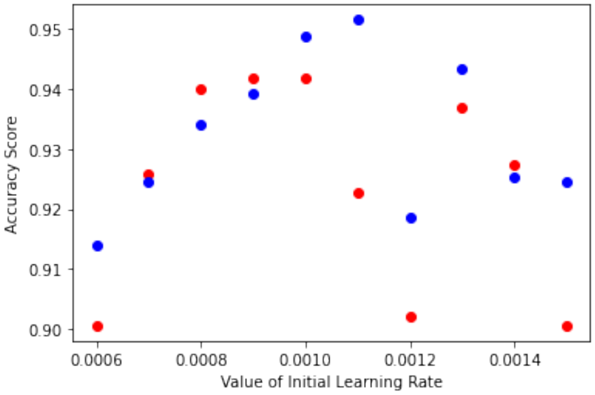

Figure 22, Dataset 2

Exploring Initial rate, Adaptive learning

In this section, I looked at changing the learning rate of the neural network this time with adaptive learning. I knew the default learning rate was 0.0001 so I wanted to have a range of values around 0.0001, in this case 0.0015 to 0.0005.

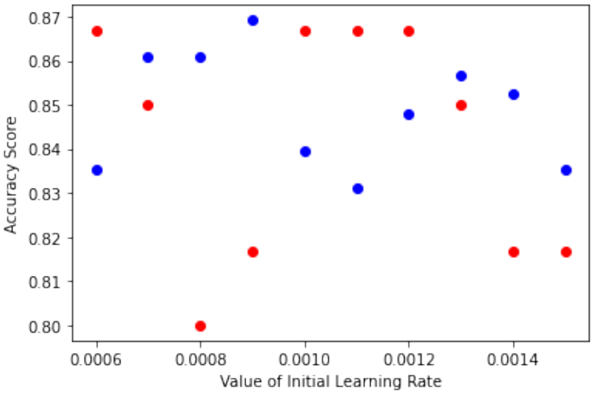

Figure 23, Dataset 1

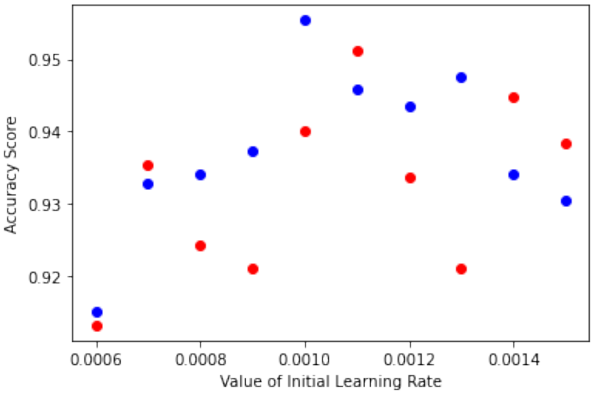

Figure 24, Dataset 2

Exploring Initial rate, Invscaling learning

In this section, I looked at changing the learning rate of the neural network this time with Invscaling learning. I kept the same data range as I had with the other two learning rate experiments.

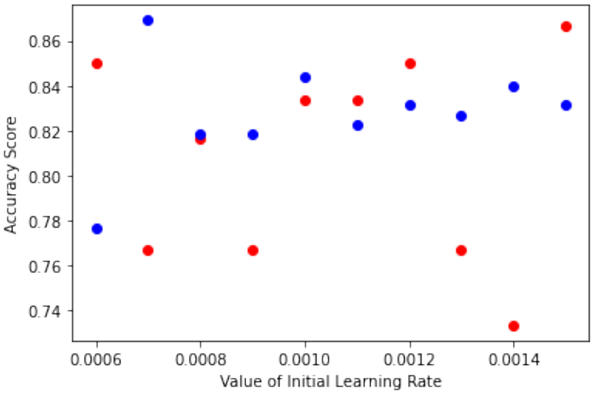

Figure 25, Dataset 1

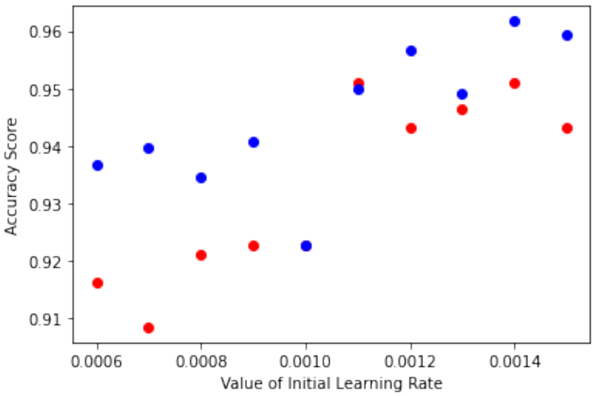

Figure 26, Dataset 2

Exploring Hidden Layer Size

In this experiment I looked at changing the value of the hidden layer size. The default of this parameter is (100,) so I tested a range that includes this value from (120,) to (80,).

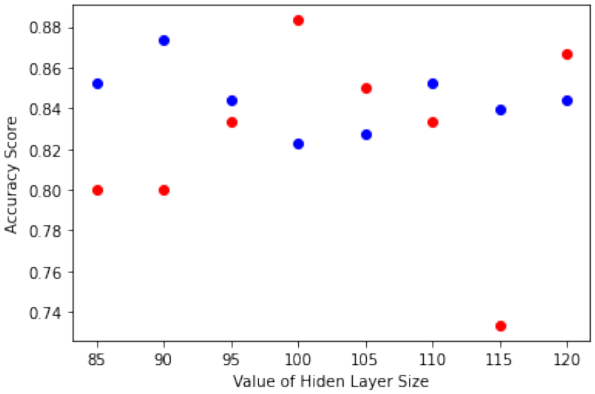

Figure 27, Dataset 1

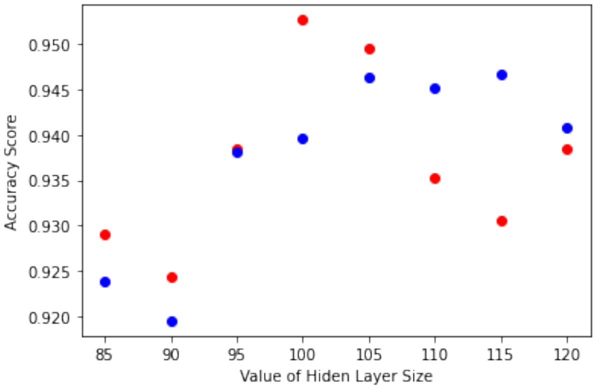

Figure 28, Dataset 2

Analysis:

Design choices

In this section, I will describe for each model of machine learning, decision tree, random forest and neural networks why I chose the parameter that were used in the experiments.

Decision Trees

For the decision trees section, I choose to look at changing the parameter effecting the result for both the data splitting method as well as the decision tree method. The variables of parameters I explored and changed in the decision tree section were test size, max depth, random state, gini vs entropy and max leaf nodes. I felt like when exploring decision trees for the first time, these would be good parameters to start off with. Looking into how test size effects splitting data for decision trees was also something I wanted to explore. I only chose to run this for decision trees and not the other models. Exploring gini vs entropy was something that I was interested in looking at and chose to look at this comparison with only one of my experiment (test size vs train size) was sufficient for getting a result.

Random Forest

For random forests, I chose to only focus on the method that creates random forests and not the data splitting method as I had done exploration on this method in decision trees. The different parameter variables that I chose to explore in this section was n estimators, max features and sample split. These methods where the ones that stood out most to me when reading the scikit learn API doc as I felt like these methods could produce interesting and clear results.

Neural Networks

For neural networks, I again chose not explore the method that splits the data but only the method that creates the neural network. The parameter variables that I chose to explore in this section was alpha, learning rate constant, learning rate adaptive, learning rate invscaling and hidden layer size. For neural networks, I was most interested in looking at the initial learning rate of the neural network with constant adaptive and invscaling. I also was interested in exploring alpha and the hidden layer size.

Results

Decision Trees – test size

For the first data set and second dataset, no matter the test size, the training accuracy was always 100 percent. The max test size in this data set seems to be 0.3 for the first data set and 0.4 for the second data set. Based on this experiment on these two data sets, a **test size of 0.35** would provide the most accurate result.

Decision Trees – max depth

For both datasets the training accuracy did not get to perfect until 9 or greater was enters as the max depth. The most accurate test data point of max depth of the first data set was at 8 and the second data set was 5. Based on these results, the best value to get the most accurate test data would be to have a **max depth of around 6.5.**

Decision Trees – random state

In both data sets, the training data is always 100 percent accurate for both datasets. The points on the graph are scattered and there **doesn't seem to be a solid and obvious accurate data point** that appears close to the same x value in both data sets.

Decision Trees – gini vs entropy

The training data was perfect for both datasets looking at gini vs entropy. For dataset 1 there were 5 red (gini) data points that had the most accurate test point and 4 green (entropy) test points. In the second dataset there was 8 red (gini) data points that had the most accurate test results and 1 green (entropy) point with the most accurate test point. These two data sets show that gini performed more accurately overall. From this experiment, setting **'gini' as criterion for the most accurate test data results.**

Decision Trees – max leaf nodes

When looking at the training data, the accuracy of the training data is perfect just after 40 leaf nodes. However, the most accurate test set seemed to be at around 17-18 nodes for both data sets. Based on this experiment **running the max leaf nodes of around 18** gives the most accurate test results.

Random Forest – n estimators

Looking at the training data it does not reach a perfect accuracy score until the value of around 25 or greater. For dataset 1, the most accurate test data plot was at 25 and 100. For dataset 2, the most accurate test data plot was at 125. From this result, it seems like a value between 100 and 125 would get the greatest results. Therefore, according to these results, the value for n estimates that would provide the **most accurate test data is around 113.**

Random Forest – max features

For this dataset the results of the training data was that the accuracy was always perfect. The test accuracy data point that had the highest value for the first data set was 6 where the number of features was 13 and for the second data set was 15 where the number of features was 20.

Random Forest – sample split

For both sets of data, when looking at the accuracy of the training data, it looks like it sinks in accuracy once the sample split is set to 0.7 or higher. For dataset one the most accurate test data point was at 0.3 and for dataset 2 the most accurate test data point was at 0.1. Therefore, based on this experiment, the **best parameter variable magnitude for sample slit is around 0.2.**

Neural Networks – alpha

When changing the value of alpha, the training accuracy score varied at every data point for both data sets. As well, the data seemed to vary for the test accuracy score as well.

Neural Networks – learning rates (constant adaptive and invscaling)

The test accuracy **for constant and adaptive** learning for both data sets saw the data point with the highest value to be **0.001 for initial learning**. For invscaling the test accuracy for both datasets seemed to jump around more and there wasn't as clear of a result in this experiment.

Neural Networks – hidden layer size

For this experiment, the training accuracy score seemed to vary for each data plot. The test training data points for both data sets had the **best accuracy score at hidden layer size of (100,).**

Comparing methods

All of the methods had an experiment where the training data accuracy score was perfect. For all of the methods, decision trees, random forests and neural networks, the test data accuracy scores had different maximum accuracy values. For decision trees, of all the experiments the highest accuracy test data point was 89 % for dataset one during the experiment for exploring maximum depth. Dataset 2 had the highest accuracy test data point of 98% during the random state experiment. For random forests, of all the experiments the highest accuracy test data point was 88% for the first data set in the exploring minimum sample split. Dataset 2 had the highest accuracy test data point of 98.5%. For neural networks, of all the experiments highest accuracy test data point was 90% for the experiment exploring initial learning rate with constant learning. Dataset 2 had the highest accuracy test data point of 96% in the exploring alpha experiment. For dataset 1 the model that gave the highest accuracy test data point of all the experiments was neural networks. For dataset 2 the model that gave the highest accuracy test data point of all the experiments was random forests.

Future improvements

It would be interesting to explore the effects of the different parameters used in the split data method for random forests and neural networks, to get a better understanding of how the way you split data effects the result. It would also be interesting to look at how post pruning the decision tree would change the results of my explorations. To improve the findings from this assignment, it would be useful to use these models with more datasets than just two. In the future I would have liked to take the final results for each experiment and create the best test accuracy score decision tree, random forest and neural network with the two data sets.