Logo

Description automatically generated

**Datamining Phase 2**

|  |  |
| --- | --- |
| **ID** | **Name** |
| 172739 | Omar Atef |
| 171820 | Karim Mohamed |
| 166436 | Mansour Wael |

**Introduction**

**Decision Tree**

Decision tree is one of the classification models that used rapidly in the data mining projects.  A Decision tree is a flowchart like tree structure, where each internal node intends a test on an attribute, each branch express an outcome of the test, and each leaf node “terminal node” carry a class label. Decision tree is famous model because it is output is intuitive and reported good accuracy. In addition, it can used in extracting patterns from large data for discrimination and predictive modeling. Decision trees classify instances by putting them down the tree from the root to some leaf node, which add the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch interrelated to the value of the attribute as shown in figure (1).

Diagram

Description automatically generated

This process is repeated for the subtree rooted at the new node. DTC have some strengths point like it is able to generate understandable rules also it is simple classification approach which classify without requiring much computation. In addition, its present clear indication of the most important attribute for classification and it can handle both category and numerical values. DTC have some weakness points like when there are many classes and relatively few training examples in the classification problem Decision trees are prone to errors. Moreover, the computational cost of training a decision tree can be high. The process of growing a decision tree is computationally expensive. At each node, each candidate partition field must be sorted before the best partition can be found. In some algorithms, field combinations are used and a search for the best combination weight must be performed. The pruning algorithm can also be expensive because many candidate subtrees must be formed and compared. DTC faced some problems like dimensionality problem and with relative fixed small training data the accuracy is very good and decrease as long as the features increase. It consists of three parts first one is the root node second one is the interior nodes, and they are representing decision tree stages and the third part is the terminal node and it is representing the final classification result. The mechanism of the decision tree is to apply the tests from the root to the terminal nodes as shown in the figure.

**Construction of Decision Tree:**

The tree can be "learned" by dividing the source set into subsets based on attribute value tests. This process is repeated on each derived subset in a recursive way called recursive partitioning. When a subset of nodes all has the same target variable value, or when the division no longer adds value to the prediction, the recursion completes. Building the decision tree classifier does not require any domain knowledge or parameter setting, making it suitable for exploratory knowledge discovery. Decision trees can handle high-dimensional data. In general, the decision tree classifier has better accuracy. The decision tree induction method is a typical classification knowledge learning induction method.

**Decision Tree Representation:**

The tree can be "learned" by dividing the source set into subsets based on attribute value tests. This process is repeated on each derived subset in a recursive way called recursive partitioning. When a subset of nodes all has the same target variable value, or when the division no longer adds value to the prediction, the recursion completes. Building the decision tree classifier does not require any domain knowledge or parameter setting, making it suitable for exploratory knowledge discovery. Decision trees can handle high-dimensional data. In general, the decision tree classifier has better accuracy. The decision tree induction method is a typical classification knowledge learning induction method.

Most decision-tree classier (e.g., CART and C4.5) perform classifications in two phases: Tree Building and Tree Pruning. In tree building, the decision tree model is built recursively splitting the training data set based on a locally optimal criterion until all or most of the records belongs to each of the partitions bear which is the same class label. To improve generalization of a decision tree, tree pruning is used to prune the leaves and branches responsible for classification of single or very few data vectors. There are many scoring criteria to evaluate the decision tree the most popular criteria are Informatic Gain and the Gini Index. Each one of them have it is own equation and the process to develop.

**Informatic Gain**

We calculate the Info (Entropy) first as shown in the equation:

A picture containing text

Description automatically generated

Then we calculate the Gain Ration as shown in the equation:

A picture containing text, watch

Description automatically generated

**Gini Index**

It measures the reduction in the class impurity from partitioning the feature space.

As shown in the equation:

Text

Description automatically generated

**Artificial Neural Networks:**

Artificial neural network (ANN) main idea is inspired from the biological neural network, it consists of huge, large number of simple processors and interconnections. The ANN models try to use some principle that we believe that it exists in the humans. Artificial neuron function mainly is to process information or data, some models of ANNs are used to study and control the behavior in both animals and machines other models are used in engineering field for several tasks such as data compression, pattern recognition and forecasting. The mechanism of how artificial neural network work that it takes input in the form of a vector after that each of the input is multiplied by the wight corresponding to it these wight are attached with a row that represent information flow and these weights represent the power of the interconnection so if the weight is high the input will be stronger, after summing the weight inputs it goes through an activation function. Activation function is a group of transfer function that are used to get the output wanted from it, the activation function has different types but mainly split into linear or non-linear.

**Activation functions:**

**-ReLU (Rectifier linear unit)**

**y = max(0, x)**

it is the most common activation function used, ReLU have also some variants such as PReLU, since it is a non-linear it can backpropagate errors easily, it also accelerates the of stochastic gradient descent when it is compared to sigmoid and since it does not activate all the neuros at one time it become easy for computation and efficient. While on the other side it has some disadvantages as it can make dead neurons, the output is not zero centered and the mean value of activation is not qual zero.

**- ELU (exponential linear unit) activation function**

**y = ELU(x) = exp(x) − 1; if x<0  
y = ELU(x) = x; if x≥0**

this function assemble cost to zero faster, and it also have more accurate results, ELU is basically the same as ReLU aside from negative data sources and unlike ReLU it can produce negative outputs. It has advantages like higher accuracy, non-saturating activation function, continuous and differential at all point and it does not suffer from dying neuros problem, while it also has some cons as it is slower than ReLU.

**Backpropagation:**

Diagram

Description automatically generatedBackpropagation is considered as the core of the neutral net training, it is the calibrating of the weights based on the rate of error achieved previously and it is method to train the ANN and this method calculate the angel of loss function with consideration to all weights in the network, backpropagation have some pros such as it is fast, easy and simple to be programmed, the only parameters need to be calibrated is only the inputs and flexible. Even it is a very good method it has some cons as the performance is dependent on the input and it is somehow sensitive when it deals with noisy data.

**Upsampling**:

To begin with, the dataset in the previous phase had 240 samples equally distributed among 6 target classes.

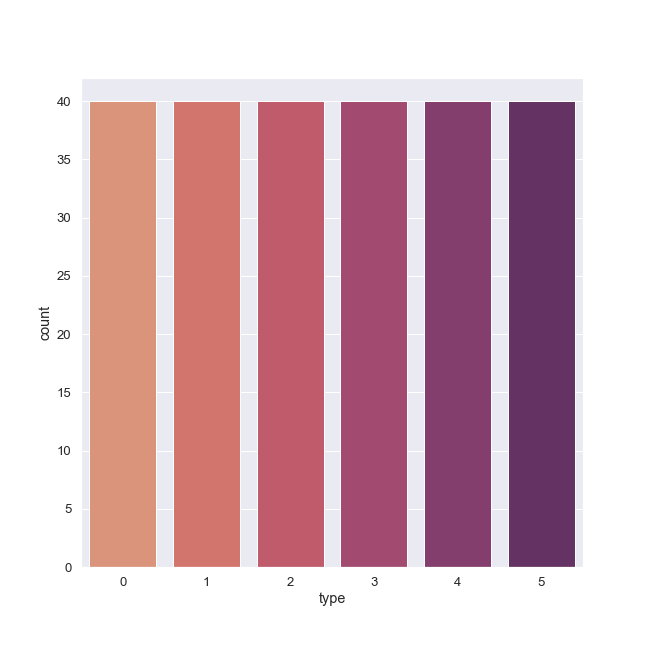


Figure (1): The number of samples in the dataset belonging to each of the target classes

240 samples only were decided to be not enough and thus, more samples should be generated in the train dataset.

Next the data is split into 4 parts: x\_train which is the features used to train the model without the target column, y\_train which is the target column corresponding to the x\_train samples, x\_test which is the small part that will be used to evaluate the model and help in determining the text accuracy, and y\_test are the actual labels of the features that will be tested/evaluated with the models once they are built. Having an actual label and a prediction determines the accuracy of the model in hand. This splitting is done before sampling because if sampling was done before splitting the dataset into train and test data, splitting might split a row that is also present in the training part and thus a certain sample is both in the training and testing set. Testing a model with a sample it has already seen in the training phase would give high but not factual results. Next, the x\_train and y\_train matrix are concatenated again. Next, the train part is split horizontally according to each class. After splitting the dataset horizonally, it was noticed that class of star type 0 is the highest most class containing samples in the training set by 37 samples so it was decided that each class will be up sampled not only to 37 but to 37 \* 4. The classes are now equal in the training set and have 148 samples each.

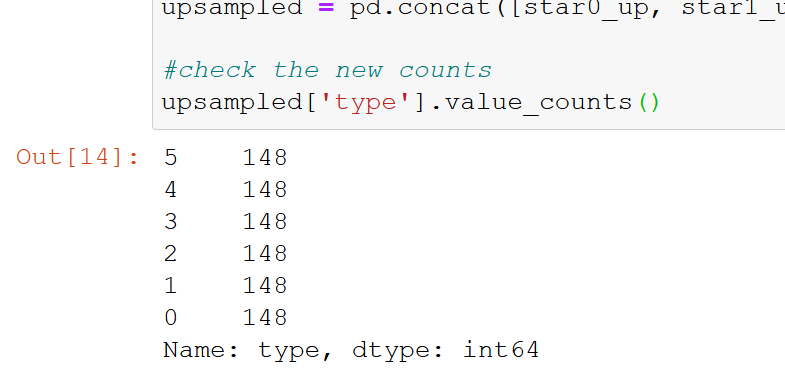


Figure (2): The number of samples belonging to each of the target classes

The training up-sampled part will be randomized in order to avoid feeding the algorithm with data belonging to the same class consequtively. Once the training part is up-sampled, it is split gain into two groups; one having all the training features and one has a column of the target labels.

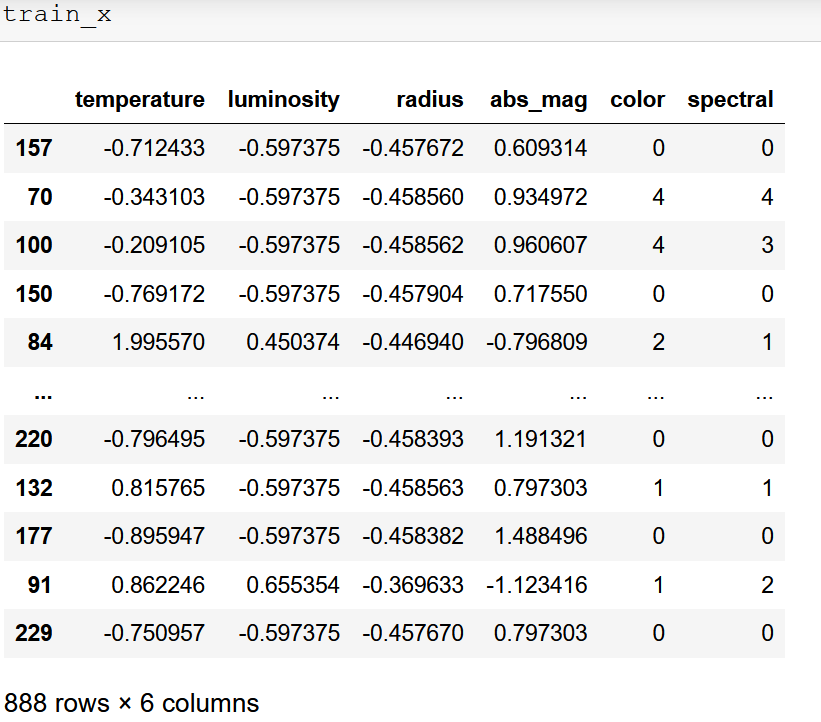


Figure (3): An image of the up-sampled and randomized x\_train part

**Comparison between the two algorithms**

Regarding the decision tree, the classifier provided by sklearn was used and the tree is fitted on the training data and their corresponding outputs/ labels or targets. The Accuracy obtained by the decision tree was 0.9583333. Below a visualization of the tree.

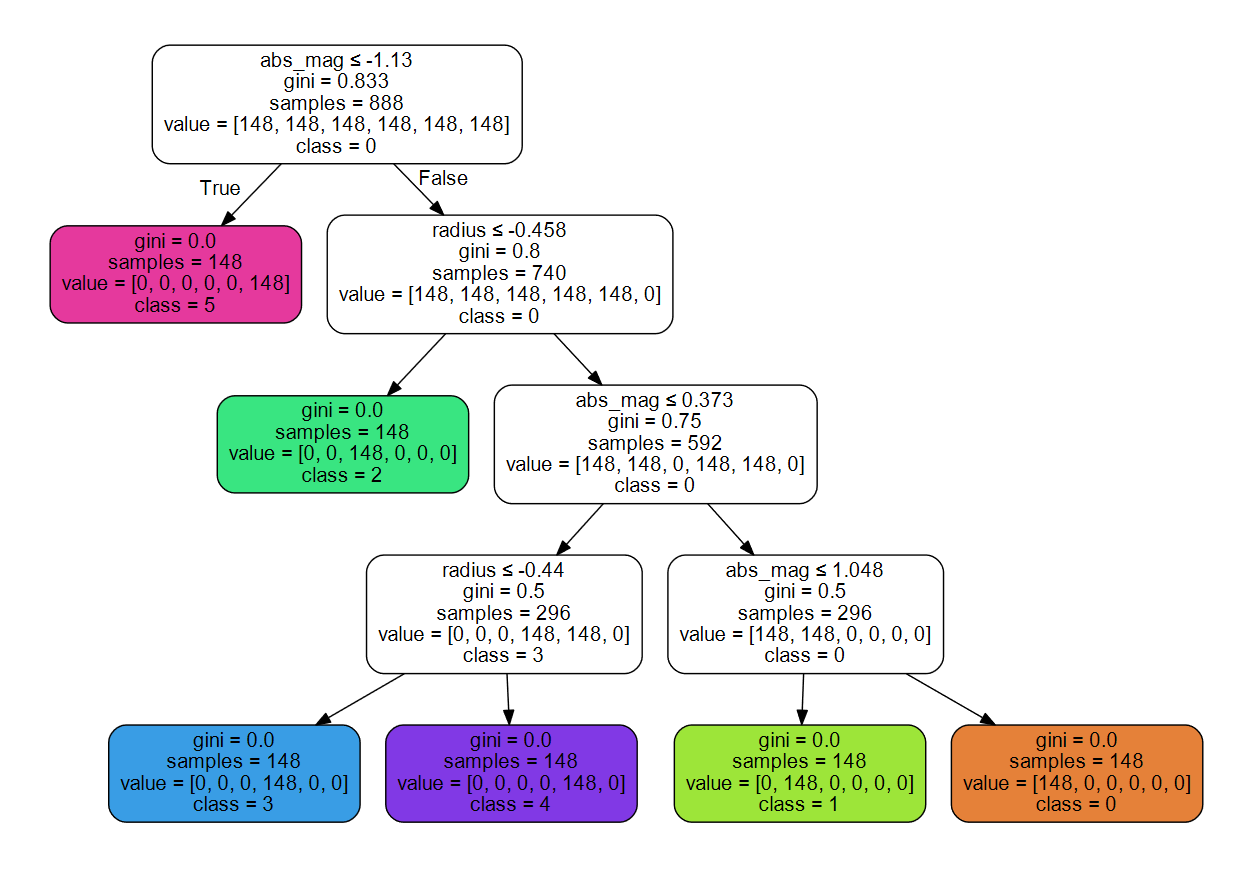


Figure (4): The visualization of the decision tree applied to the star dataset

Regarding neural network, a plan was put to pick the best network from a set of 4 differently configured networks as the accuracy of neural networks depend on various attributes and it is a complex problem to choose the best hyper parameters and arguments that would give the best result and thus, it is considered a trial-and-error problem. First, a network with 3 hidden layers with number of units 50, 100, and 50 respectively and with activation function relu was tested. The activation function of the last layer was determined to be softmax. This configuration gave 0.9930 accuracy on the training data and 0.9166667 accuracy on the testing data. When observing the validation and training loss, it was found that the validation loss is less than the training loss which reflects that there is a problem as it is impossible that the model was more accurate on data that it has not seen before during validation than on the training data that it makes epochs around repetitively. A decision was made to try another configuration and keep training.

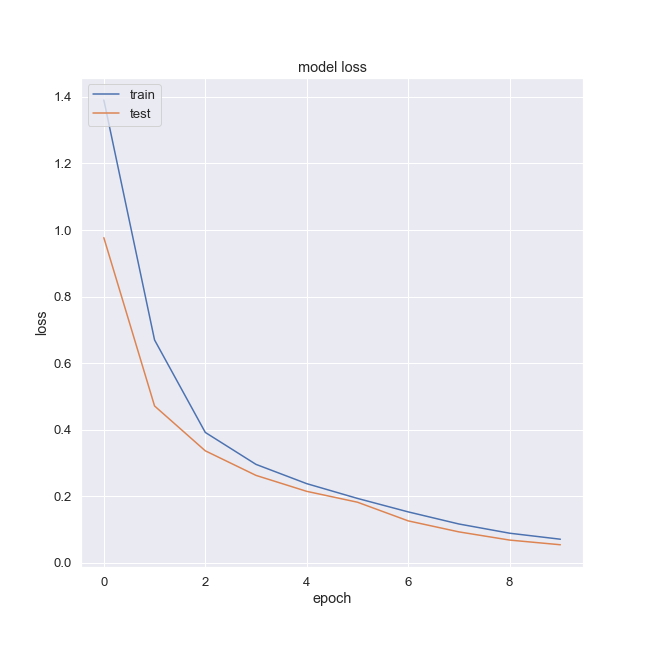


Figure (5): The validation and training loss of network 1

Next, a new network was configured on a smaller number of neurons in each layer with the same activation function of the output layer (softmax) and with the same 3 hidden layers with number of neurons of 12, 24, and 12 respectively but this time the activation function used for the hidden layers was changed from RELU to ELU. This network achieved 0.9839 accuracy on the training data and 0.9791 accuracy on the test data which are both better than those of the last network.

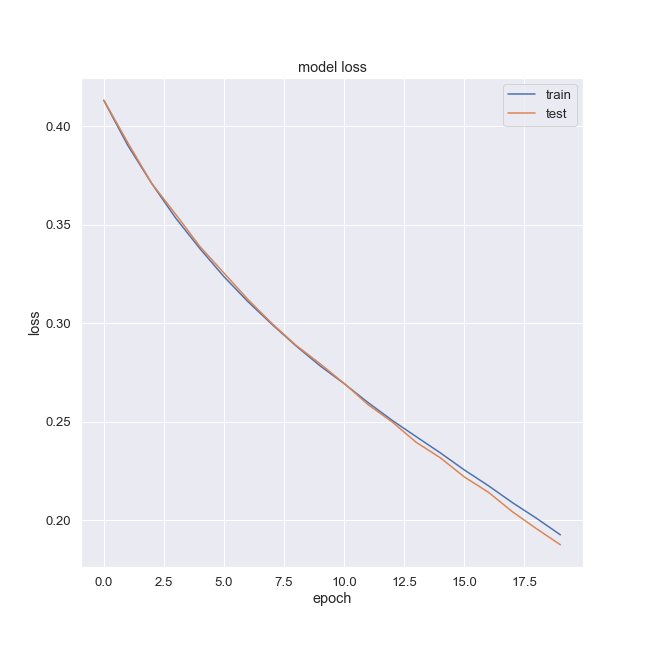


Figure (6): The validation and training loss of network 2

It can be observed that the validation and testing losses this time are closer to each other. However, during the final epochs the validations loss is below the training loss which gives a slight indication that the problem is still present.

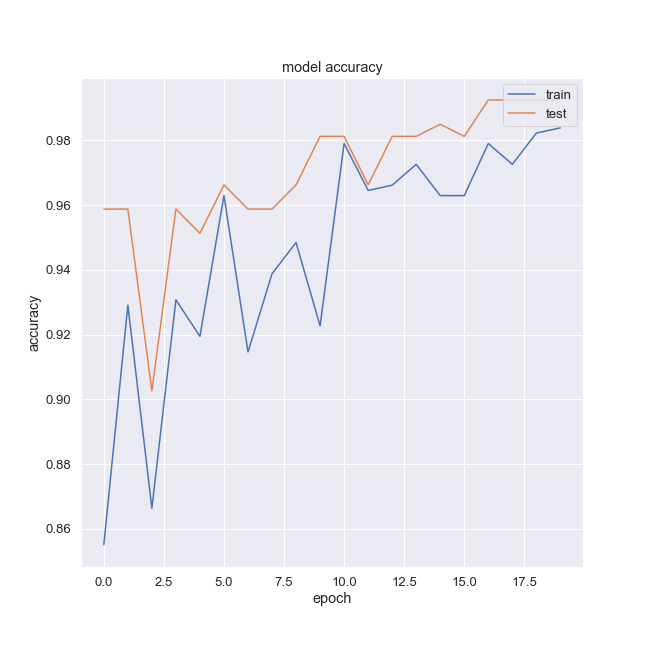


Figure (7): The validation and training accuracy of network 2

By observing the accuracy this time, it can be noticed that the validation accuracy is much higher than the training accuracy which indicates a high chance that the model is overfitted on the data it was trained on and was unable to generalize on other data that it has not seen before.

For the third network, all configurations were kept the same except for the learning rate which was modified from 0.0002 to 0.0001 and the number of epochs that increased from 20 epochs to 50 epochs. The third network achieved 0.9710 accuracy on the training data and 0.8958333 accuracy on the test data. However, the third network seemed to have fixed the overfitting problem that was present in the previous networks.

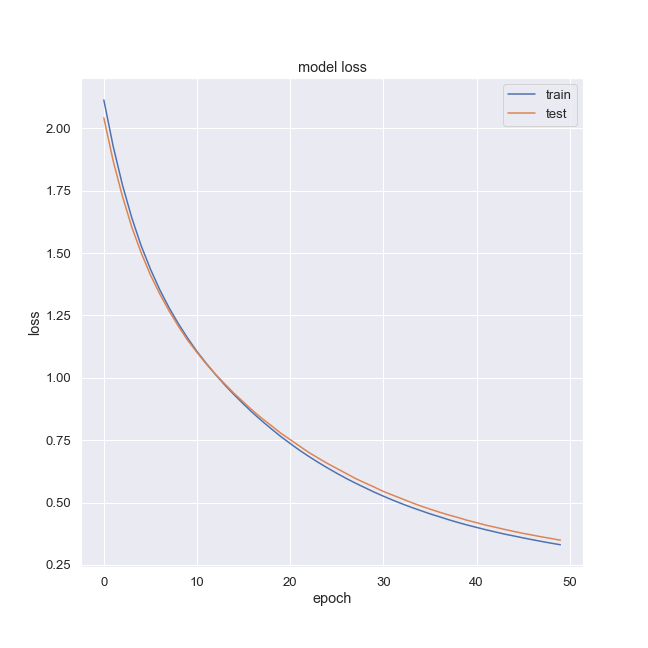


Figure (8): The validation and training loss of network 3

It can be noticed that at the start of the epochs the validation loss was less than the training loss but by the time the network did more epochs, the validation loss becomes more than the training loss which makes sense and conforms with logic.

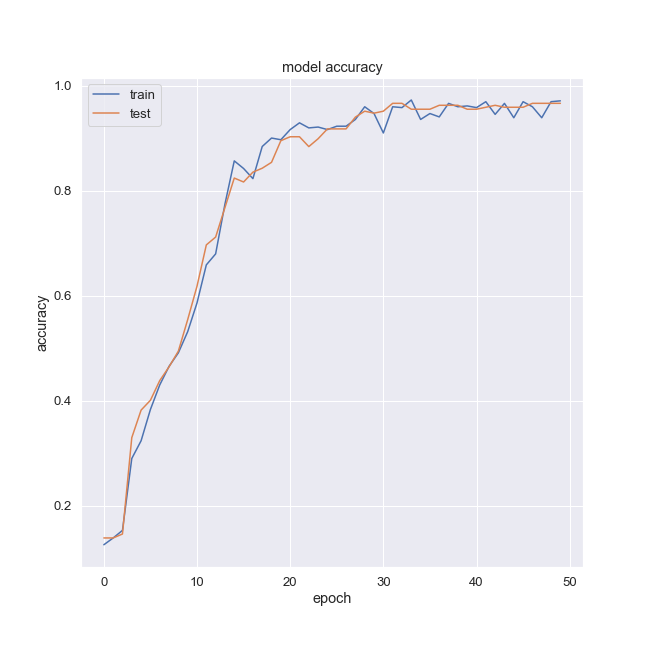


Figure (9): The validation and training accuracy of network 3

By observing the validation and training accuracy, it is obvious that the training accuracy and the validation accuracy are close to each other with the training accuracy surpassing the validation accuracy sometimes and sometimes the opposite case is achieved where the validation accuracy is higher than the training accuracy.

For the fourth and last network, all configurations were kept the same as the third network but the number of epochs was further increased to 200 epochs. Regarding the validation and training accuracy, they looked very similar to those of the third network which does not constitute a great difference.

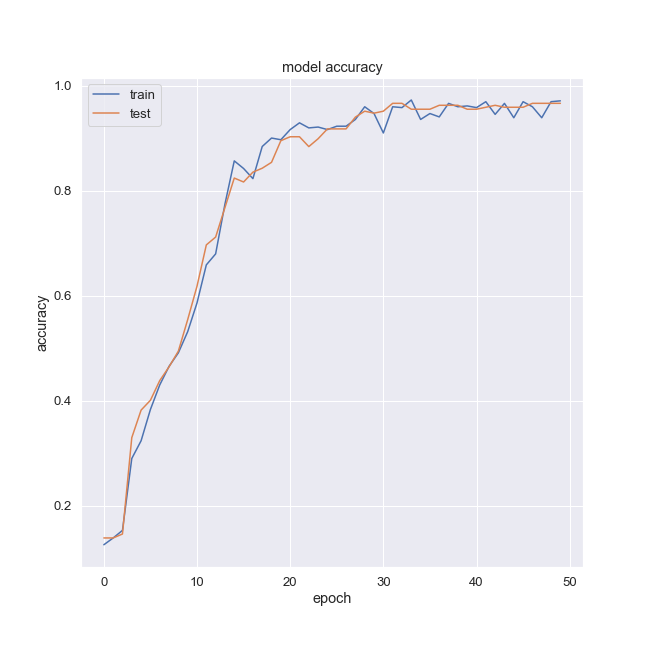


Figure (10): The validation and training accuracy of network 4

For the validation and training loss, it is apparent that they are very close to each other and almost appear as one line. However, the third network is determined to be the best as the validation loss exceeds the training loss at the end of the 50 epochs made.

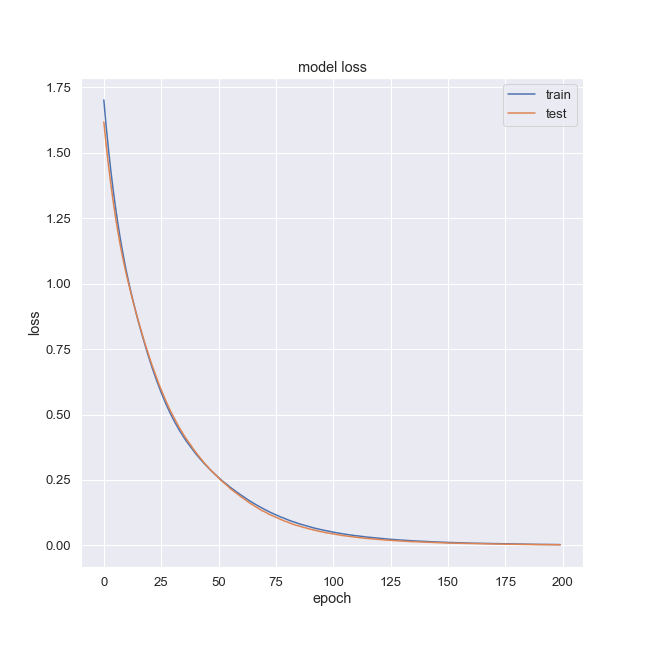


Figure (11): The validation and training loss of network 4

**Conclusion:**

In conclusion, the decision tree performed better on the star dataset with a testing accuracy of 0.9583333333333334 which is above 95% and the best neural network configurations has a testing accuracy of 0.8958333 which is above 89%. The results can well be reflected in the confusion matrix of each algorithm as the decision tree algorithm has done a better job of predicting the correct actual label of the sample out of the 48 test samples.

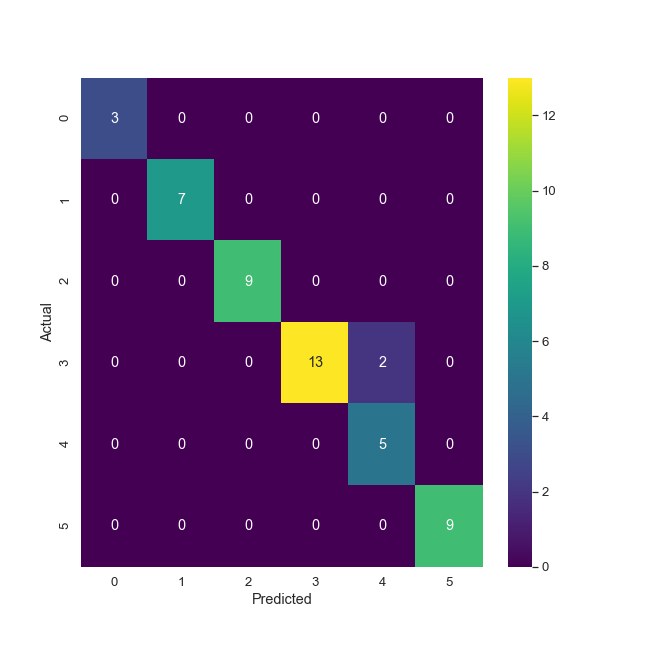


Figure (12): The confusion matrix of the decision tree algorithm

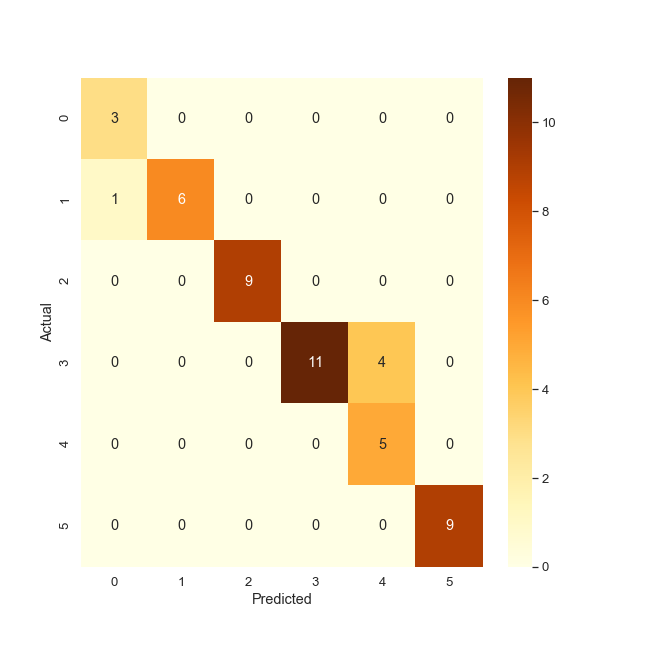


Figure (13): The confusion matrix of the neural network algorithm

**References:**

[1] H. H. PHILIP H. SWAIN, "The Decision Tree Classifier: Design and Potential," The Decision Tree Classifier: Design and Potential , pp. 142-147, jully 1997.

[2] R. N. L. A. a. S. D. Anthony J.Myles, "The Decision Tree Classifier: Design and Potential," JOURNAL OF CHEMOMETRICS, pp. 275-285, 2004.

[3] N. Gupta, "Artificial Neural Network," Inter national Conference on Recent Trends in Applied Sciences with Engineering Applications, pp. 24-27, 2013.

[4] S. Singh, "ELU as an Activation Function in Neural Networks," Deep Learning University, [Online]. Available: https://deeplearninguniversity.com/elu-as-an-activation-function-in-neural-networks/. [Accessed 4 june 2021].

[5] V. Kakaraparthi, "Activation Functions in Neural Networks- What are they?, How they work? and Where to use them?," 8 february 2019. [Online]. Available: https://prateekvishnu.medium.com/activation-functions-in-neural-networks-bf5c542d5fec. [Accessed 4 june 2021].

[6] M. Buscema, "Back Propagation Neural Networks," Substance Use & Misuse, pp. 233-270, 3 july 2009.