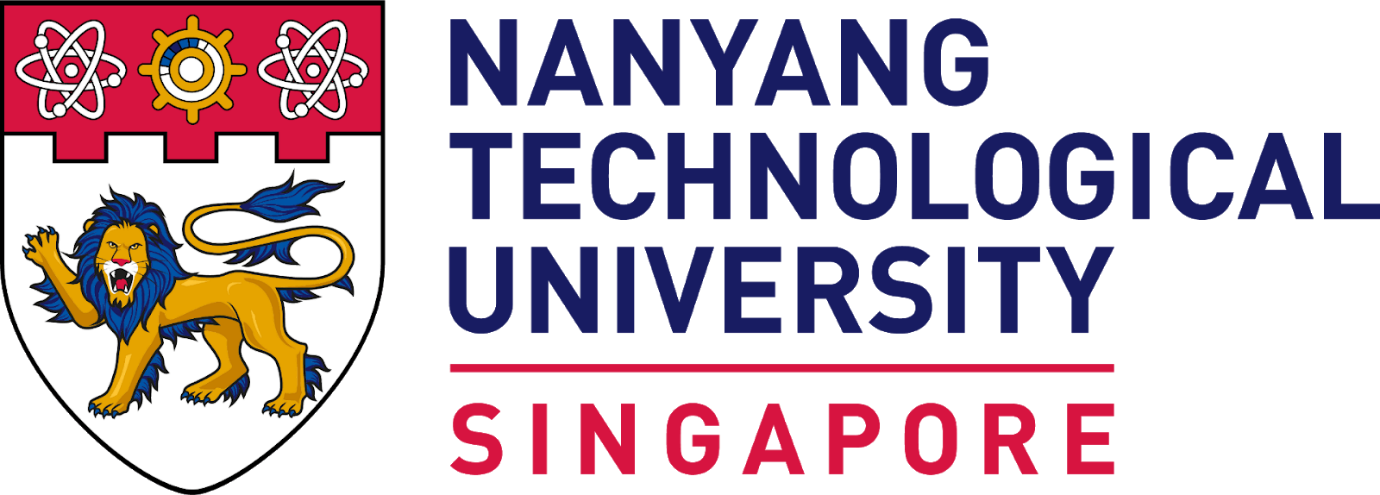
****

**SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY**

**CZ4042**

**ASSIGNMENT 1**

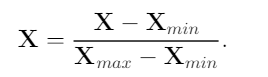
**DONE BY:**

**JANAKI H NAIR (U1622879J)**

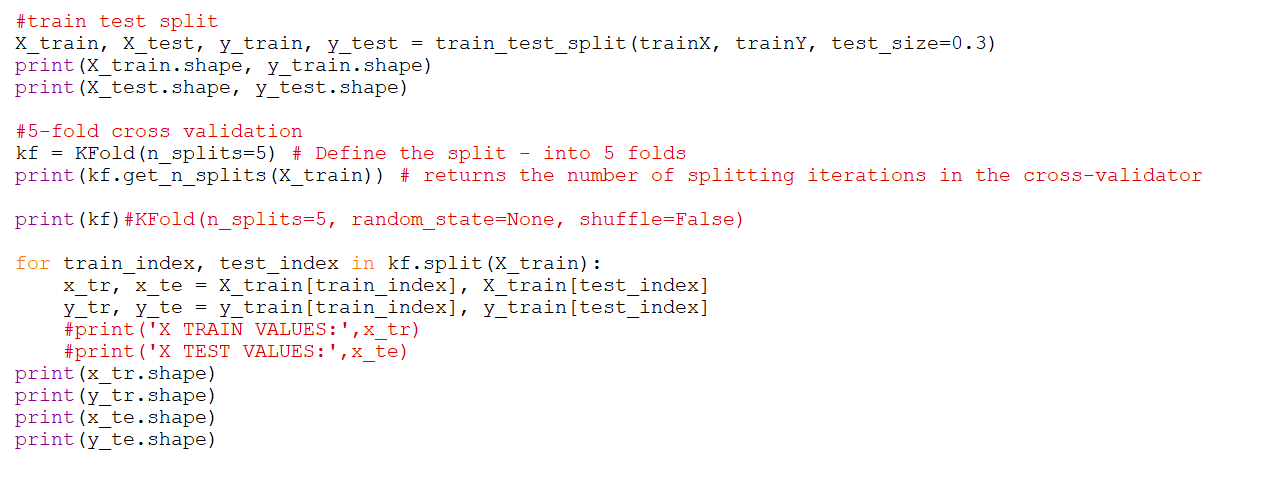
**DODDA SHARON OLIVIA (U1621792K)**

**PART A: Classification Problem**

We were given the Cardiotocography dataset which consists of 21 input attributes and 1 class label namely the NSP label with values 1, 2, 3. This problem aims at training the neural network to classify the test set with NSP labels b designing a feedforward neural network.

The given dataset consist of 2126 samples out of which 70%, is used to train the model and the remaining 30% is utilised as the test data. The dataset is scaled according to:

The 5-fold cross-validation methodology is then applied on the training set to select the optimal model.

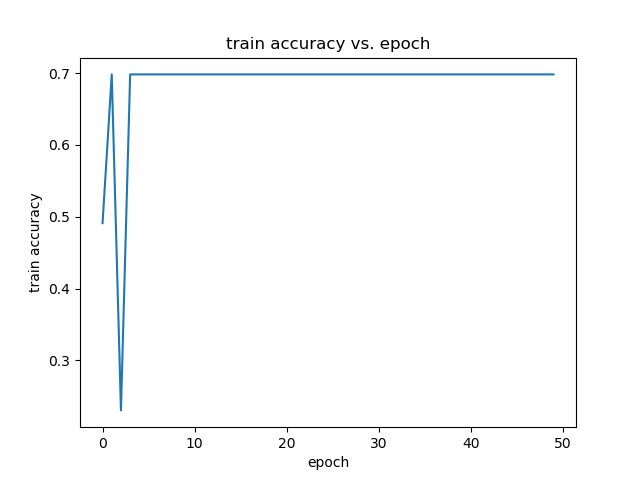


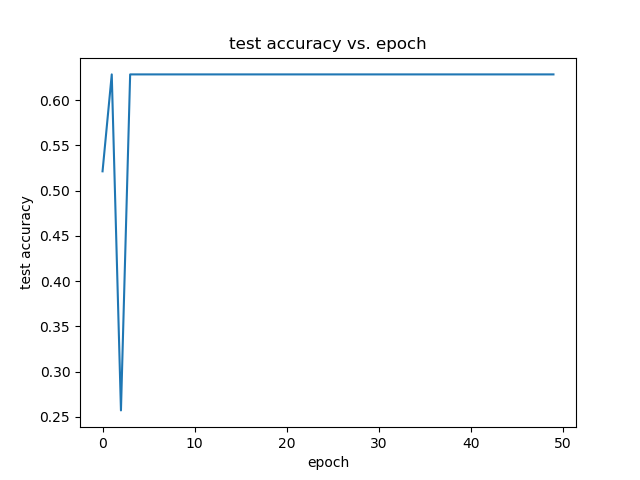
The 3-layer feedforward neural network consists of an input layer, one hidden layer and one output softmax layer. The hidden layer consists of 10 neurons having ReLU activation function. A mini batch gradient descent is then used to train the neural network. A learning rate of 0.01, L2 regularization with weight decay of β =10-6 and a batch size of 32 is assumed in the initial training.

**1.1 Train and Test Accuracies**

The following figures show the training accuracies and testing accuracies against the number of epochs respectively.

Batch size = 32:



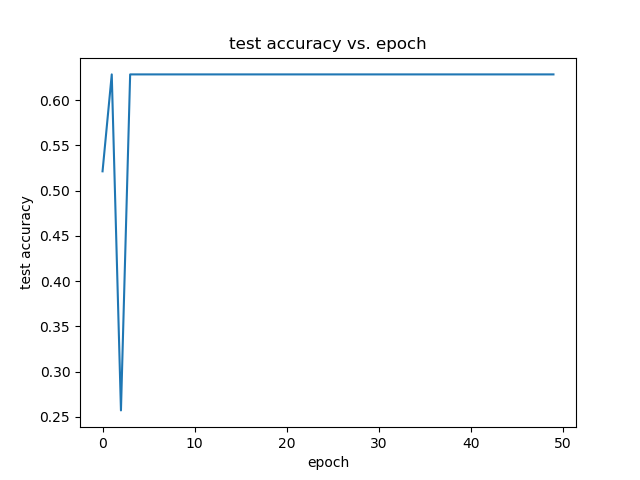


From the plots, it is evident that the test error converges within about 5 epochs.

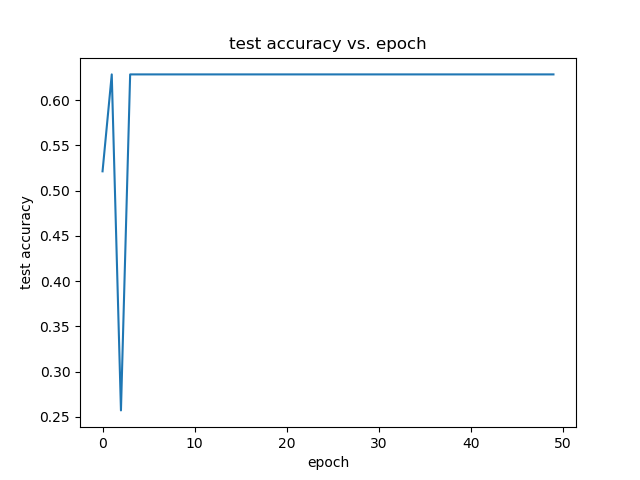
**1.2 Optimal Batch Size**

The neural network is then trained with different batch sizes [4, 8, 16, 32, 64]. The graphs below plot the time taken for one epoch as well as the cross-validation accuracies against the different batch sizes. They will be useful in selecting the optimal batch size.

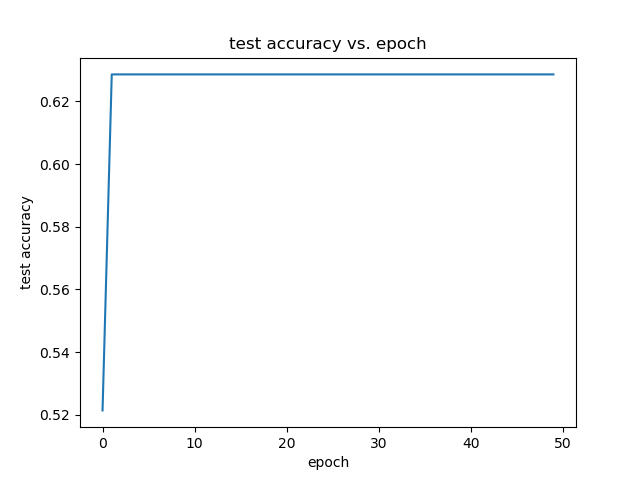
Batch size = 4:



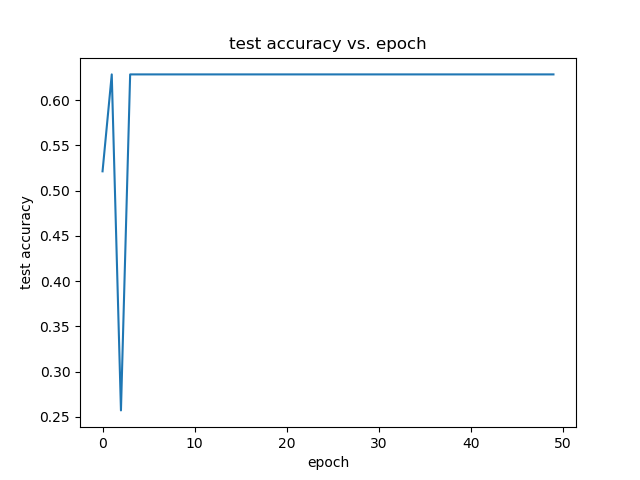
Batch size = 8:



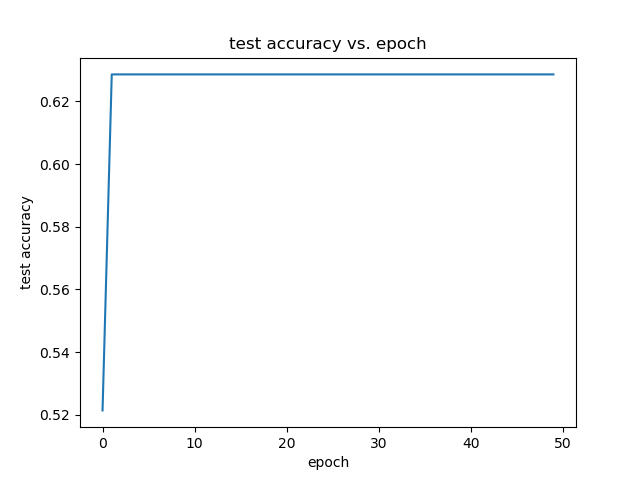
Batch size = 16:



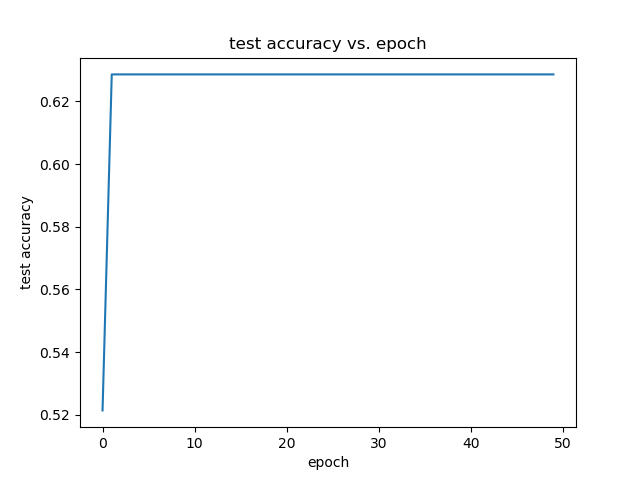
Batch size = 32:



Batch size = 64:



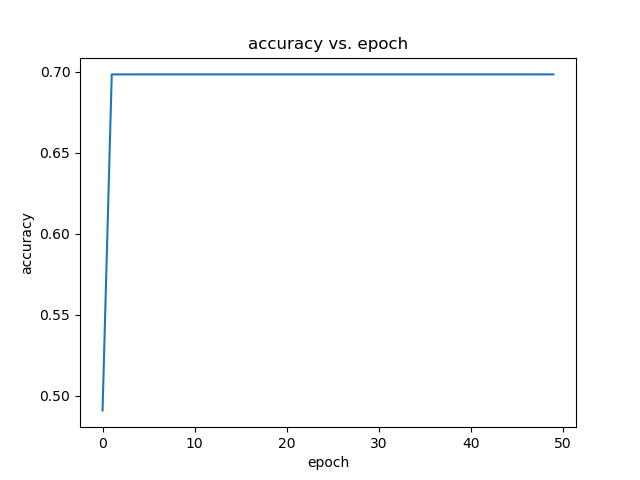
The figures below show the training and test accuracies for the optimal batch size chosen, where batch size=64.



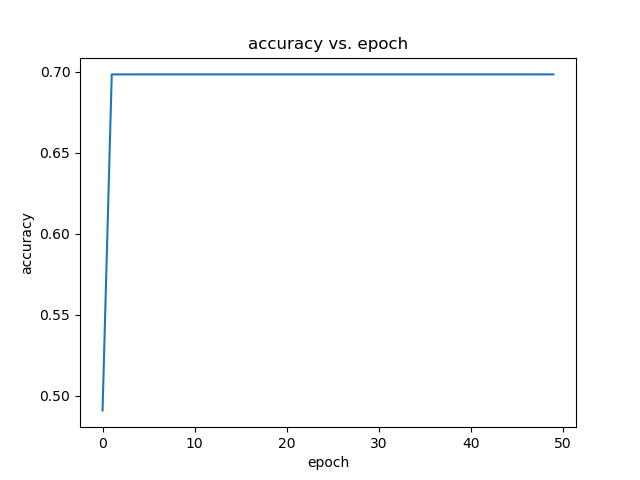
**1.3 Optimal Number of Hidden neurons**

The neural network is now trained to select the optimal number of hidden neurons for the hidden layer from among [5, 10, 15, 20, 25]. We will be using the batch size that we have selected in the previous run. The figure given below show the cross-validation accuracies against number of epochs for the different number of hidden neurons. From the figure, we can prove that the optimal number of hidden neurons is 25.

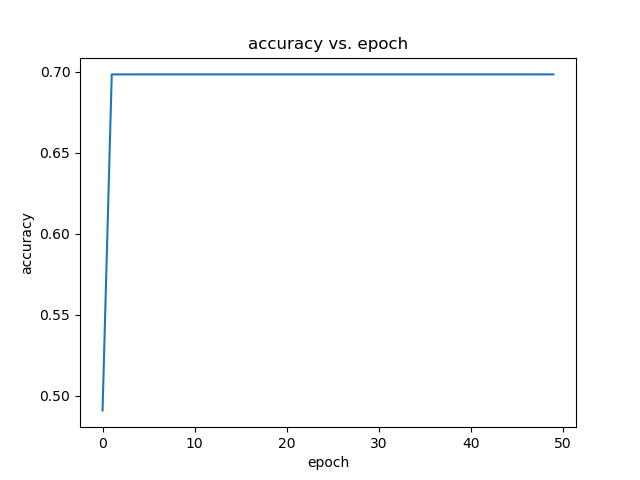
Num of neurons=5



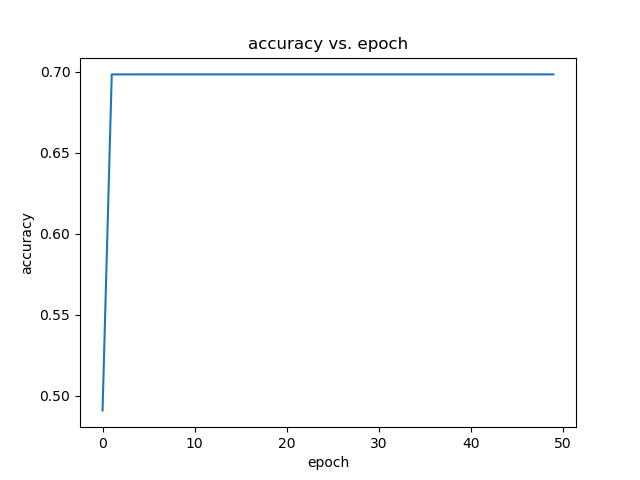
Num of neurons=10



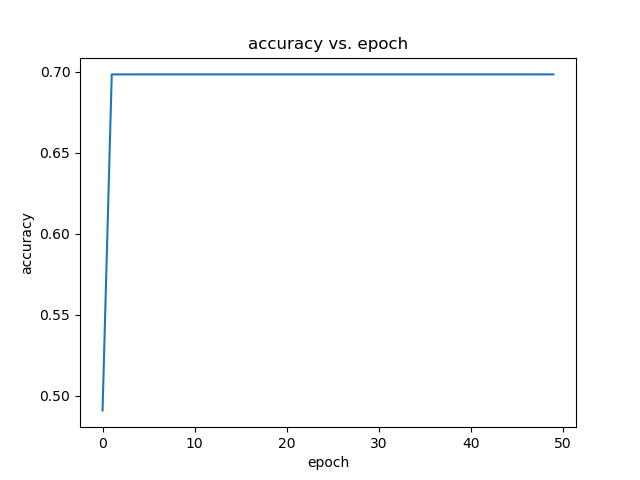
Num of neurons=15



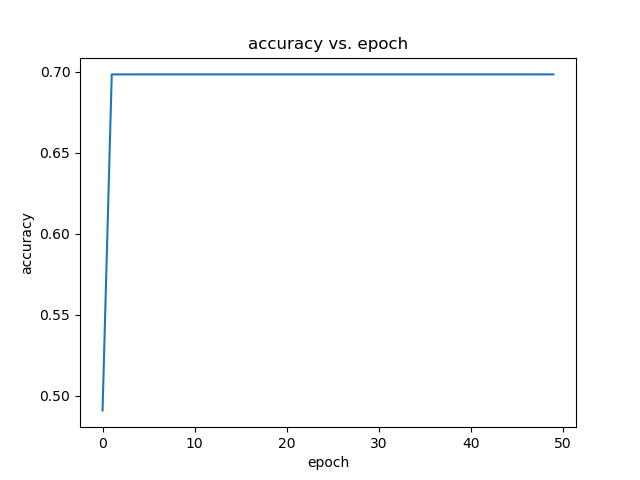
Num of neurons=20



Num of neurons=25



The figures given below show the training and testing accuracies against epochs for the optimal hidden neurons chosen earlier., where number of neurons = 25



**1.4 Optimal Decay Parameter**

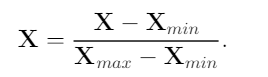
After selecting the optimal batch size and the optimal number of hidden neurons, we proceed to train the network to choose the optimal decay parameter. The different values of the decay parameter are [0, 10-3, 10-6, 10-9, 10-12]. The figure given below show the cross-validation accuracies against number of epochs for the different decay parameters. From the figure, we can prove that the optimal decay parameter is \*insert graph and observation\*

The figures given below show the training and testing accuracies against epochs for the optimal decay parameter chosen earlier. \*insert graph and observation\*

**PART B: Regression Problem**

We were given the Graduate Admission Predicate dataset which contains 7 input attributes and 1 target variable (the probability of getting an admit).  Our task is to build a neural network to predict the probability of getting an admit based on the 8 input variables.

The given dataset consist of 400 samples out of which 70%, is used to train the model and the remaining 30% is utilised as the test data. The dataset is scaled according to:



The 3-layer feedforward neural network consists of an input layer, hidden layer of 10 neurons with activation function ReLU, and alinear output layer. A mini batch gradient descent is then used to train the neural network. A learning rate of 0.01, L2 regularization with weight decay of β =10-6 and a batch size of 32 is assumed in the initial training.

**1.1 Train and Test Accuracies**

Figures 1 and 2 given below show the train and test errors against 20000 epochs.

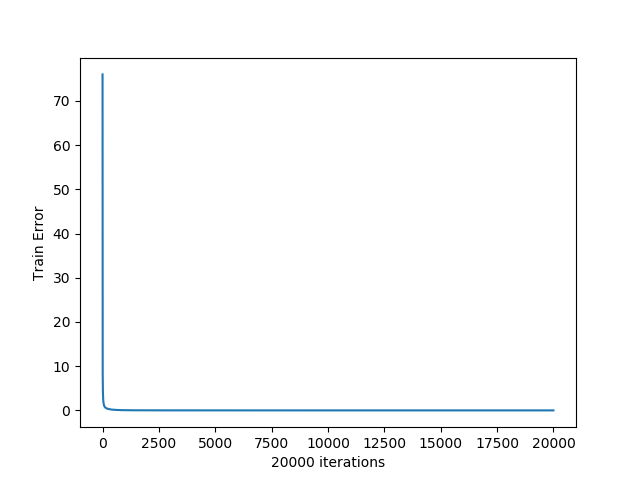


Figure 1

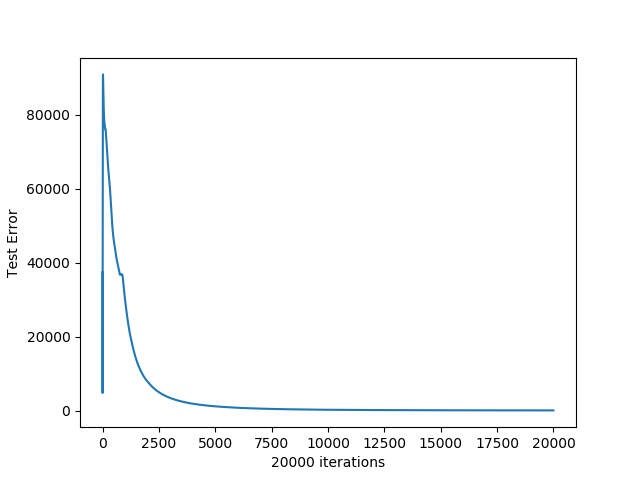


Figure 2

From figure 2, we can observe that the test error converges within the first 550-600 epochs and the final test error upon 20000 iterations is about 100.

Figure 3 shows a scatter plot of predicted values against target values for 50 test samples. From the figure, we can see that the predicted values of admit, more or less, follows the same gradient as that of the actual admit values. We can also observe that the data points of the predicted values are more crowded for higher values of admit as compared to lower values.

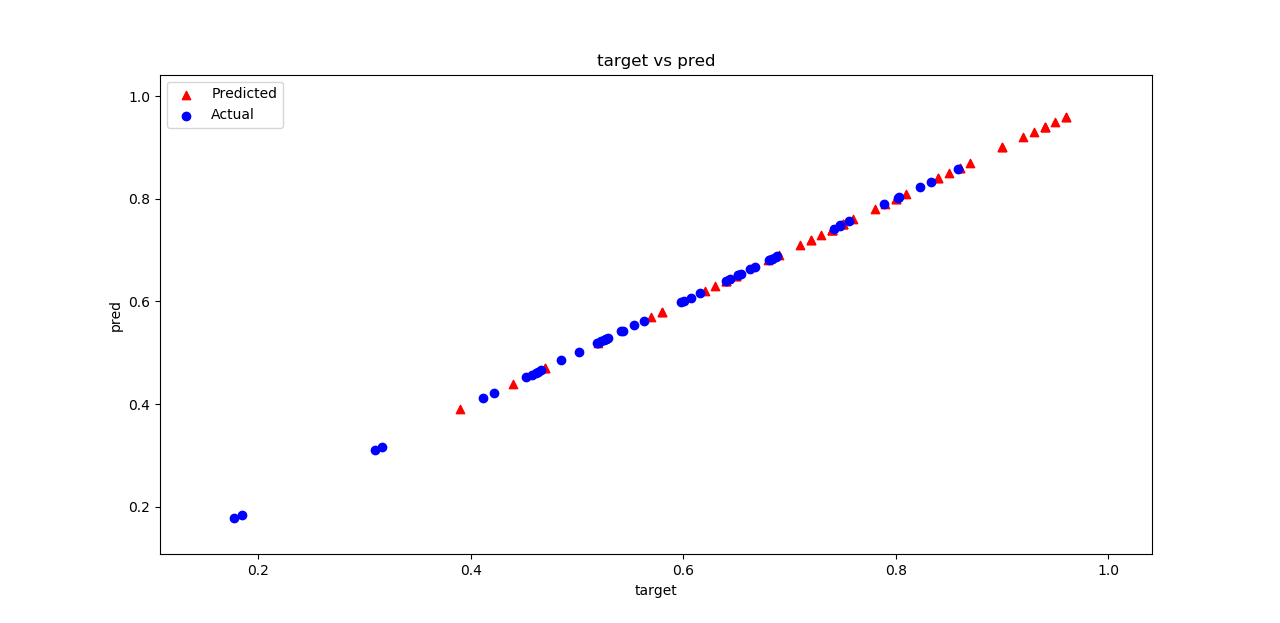
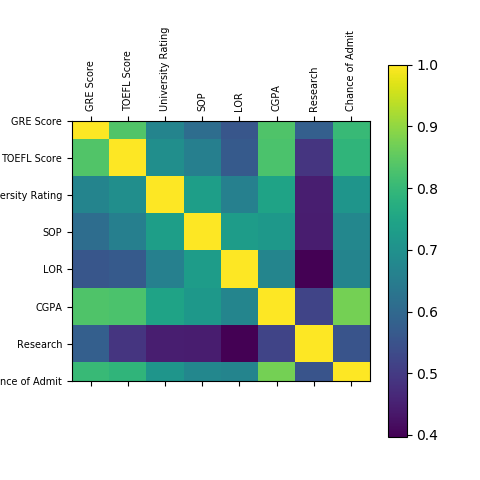


Figure 3

**1.2 Correlation Matrix**

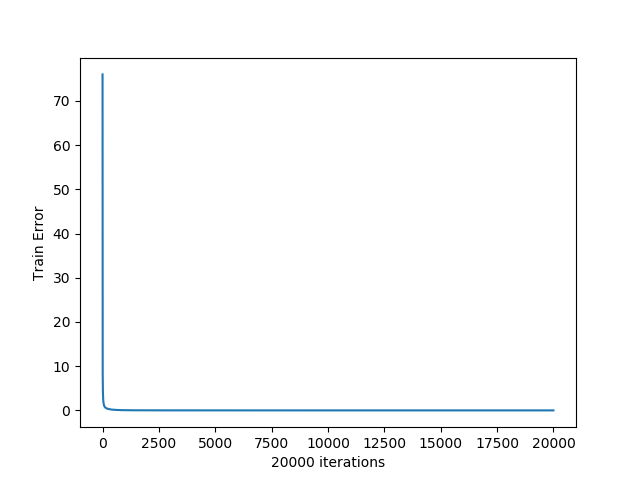
A correlation matrix is used to summarize data for advanced analysis with each cell in the matrix showing the correlation between the variables. (Figure 4)

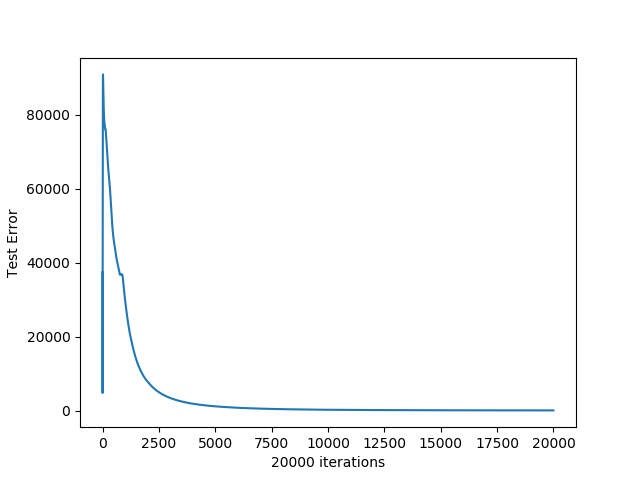
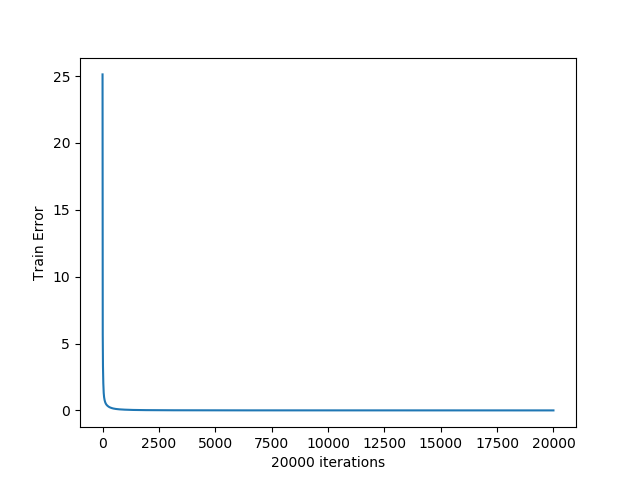
  
  
  
  
  
  
  
From figure 4, we can see that the variables {GRE Score, TOEFL Score, CGPA} are strongly correlated to one another which can be justified in the real world because these three scores combined are potential predictors of getting admission to universities. On the other hand, LOR is less correlated with the university rating as seen from the matrix.

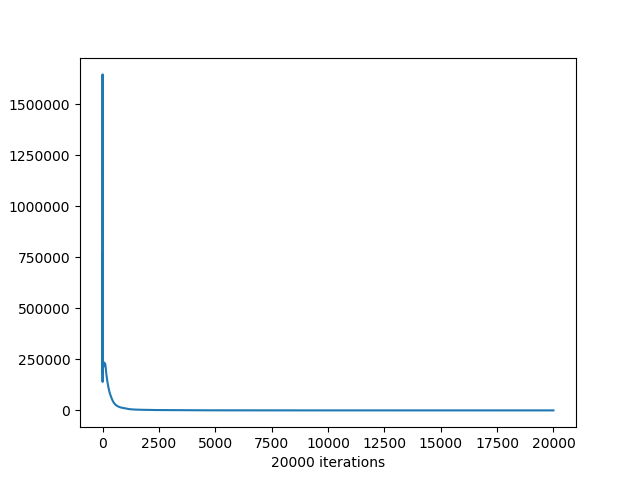
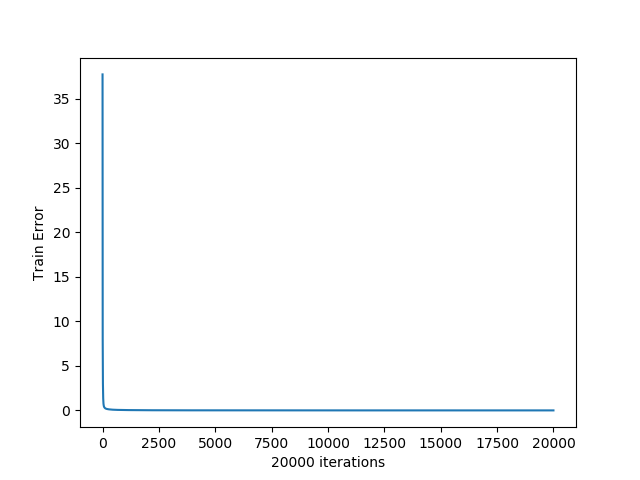
The feature variable CGPA is the most significant correlated input variable to the output variable,i.e, chance of admit. This way we can visualize variables to get an idea of how variables are correlated.

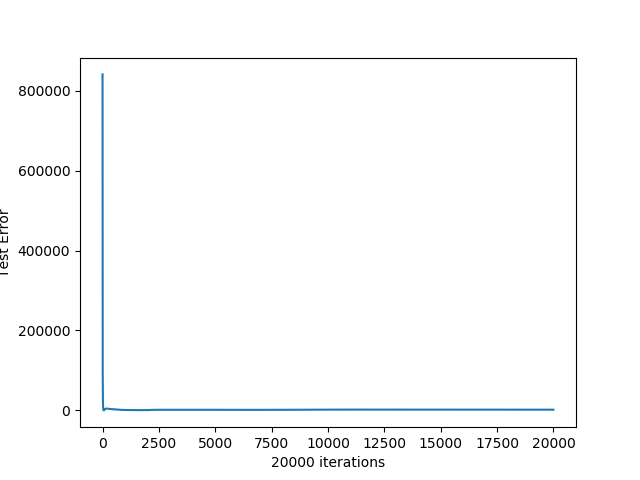
**1.3 Recursive feature elimination**

Datasets may contain many feature sets to train regression algorithms. Not all features may contribute to the prediction of the output variable. Hence, some feature variables are removed for better accuracy and to avoid overfitting.

As the name suggestd, recursive elimination recursively removes the unnecessary features until we obtain the optimal number of features. In this assignment, we are removing features until we get models with 6 and 5 input features selected using the RFE.

7-train error  
  
 7 test error

6 train error  
  
6 test error  
 5 train error

  
 5 test error

**1.4 4-layer network**

We trained the 4-layer and 5-layer networks with the same learning rate and using drop-outs with a keep probability of 0.8 (as given in the code).