Statistical Data Mining Project 5

Question 1

In this problem we had to fit a neural network to the spam data which was available through the package “ElemStatLearn”.

First the spam response variable was converted into the spam as 1 and email as 0 to allow for classification. The spam data was then split into testing and training data and then cross-validation method was used to determine the number of neurons to use in the layer. The test and train error was determined for the layers as shown in figure 1.

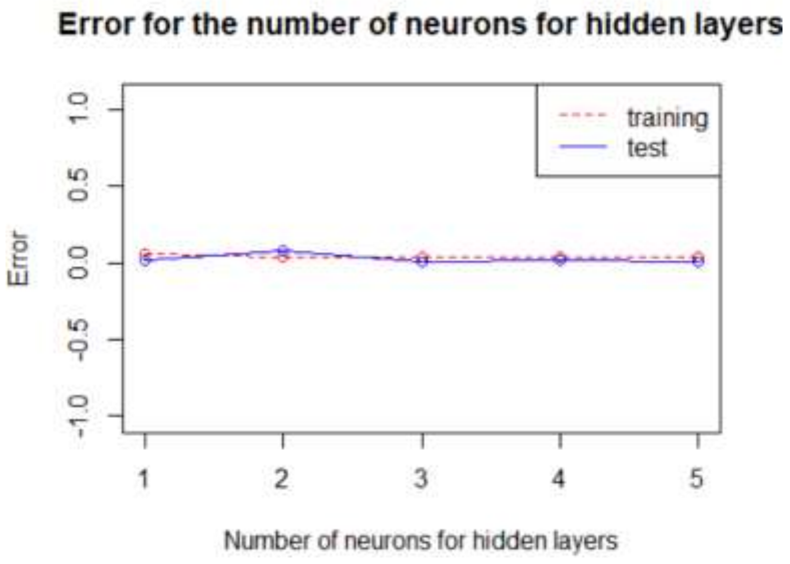


Figure 1: Cross-validation method to determine the number of neurons to use in the layer

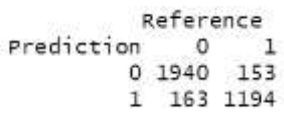
From the above figure we see that 5 hidden layers provides the least error and thus this was fit in a neural network and we obtained the following the classification table:

```
          pred_class
true_class    0    1
         0 1983  120
         1  161 1186
```

- Accuracy = (1983+1186)/3450=0.920
- Recall (sensitivity) =1983/2144=0.925
- Precision=1983/2103=0.943

Afterwards we had to compare your results to those for the additive model i.e. logistic regression.

The test error for the logistic regression was 0.0915942
The train error for the logistic regression was 0.06950478
In comparison for the neural network fit the logistic regressions had the following classification table:


```
              Reference
Prediction      0    1
         0 1940  153
         1  163 1194
```

- Accuracy =0.9084
- Recall (sensitivity) =0.9025
- Precision=0.9269

Considering both the classification performance and interpretability of the final model, we can see that the neural fit has an overall better classification performance with an accuracy i.e. the true positive and true negatives combined is 94% as compared to 92% for the logistic regression. In terms of interpretability, the precision i.e. the ratio of correctly predicted positive observations to the total predicted positive observations, and the recall (sensitivity) i.e. the ratio of correctly predicted positive observations to the all observations in the actual class for the neural fit was higher than the logistic regression by roughly 2%, as shown in the tables above.

Question 2

In this problem we had to take a classification data set and divide it up into a learning set and a test set. I chose a heart disease data set (https://www.kaggle.com/ronitf/heart-disease-uci) and performed a classification on it. The data set had 303 rows and 14 columns. It was split into training set (3/4) and testing set (1/4). It is a binary classification dataset. It has 13 predictors with the response variable being the target (i.e. presence of heart disease=1 and no heart disease=0.

We had to change the value of one observation on one input variable in the learning set so that the value is now a univariate outlier. I chose to change one observation in the Trestbsp variable to 400. This can be shown in the boxplot below.

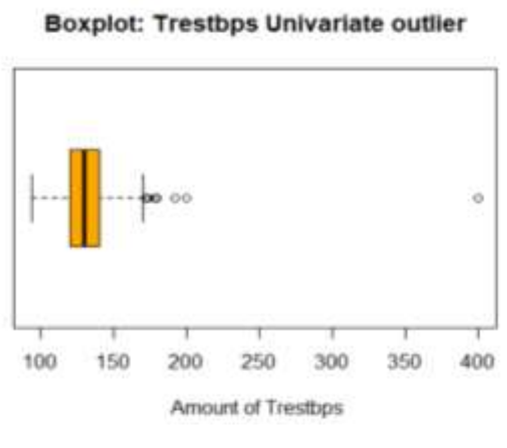


Figure 2: Boxplot of the Trestbps variable with an outlier observation

Afterwards, we had to fit separate single hidden-layer neural networks to the original learning-set data and to the learning set data with the outlier. The 2 different plots of one neuron with single hidden layer for the learning set data with and without the outliers are shown below in figure 3 and 4.

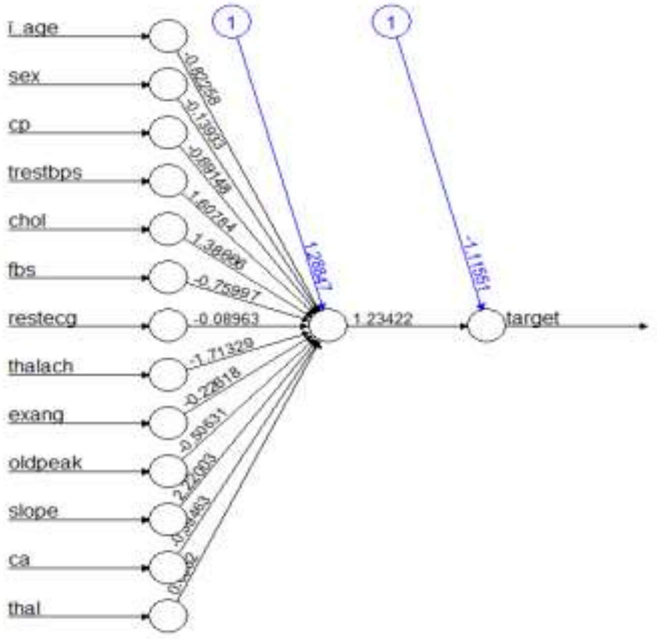


Figure 3: A single hidden-layer neural network using original data

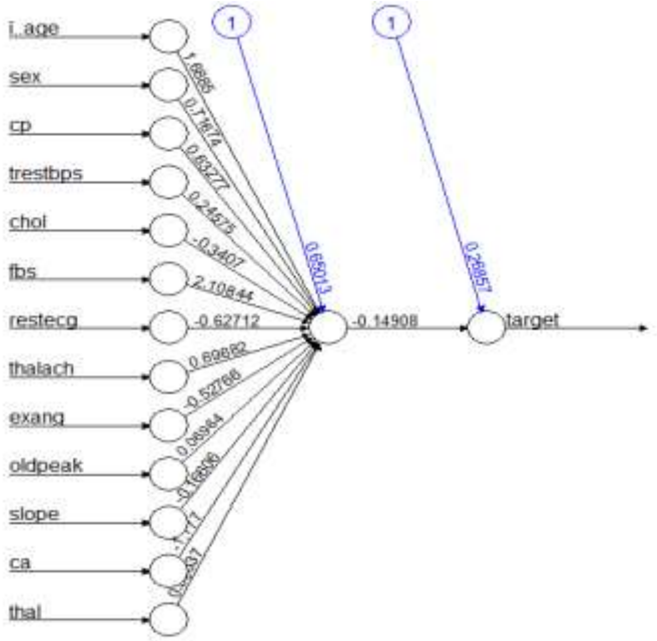


Figure 4: A single hidden-layer neural network using original data with the outlier

Afterwards, we used cross-validation to determine the number of neurons to use in the layer. The error for each corresponding neuron in the hidden layer from 1-5 is shown in figure 5.

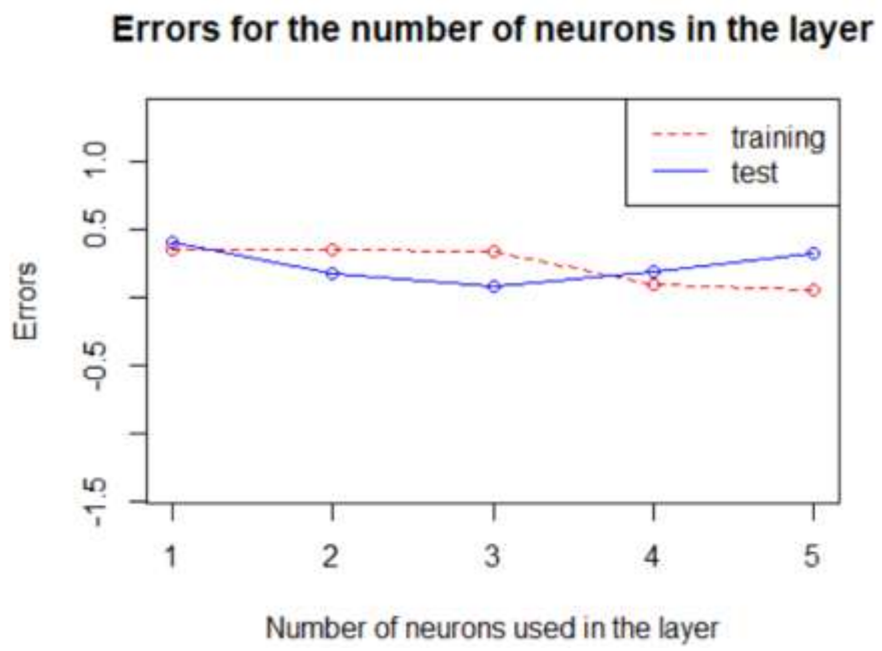


Figure 5: Cross-validation method to determine the number of neurons to use in the layer

From figure 5 we can see that 3 number of neurons for hidden layers provides the least error using the cross validation and thus this was the number that we used to determine the outlier effect on the training data.

Afterwards we had to shrink the value of that outlier toward its original value and evaluate when the effect of the outlier on the neural network fit as it vanishes. I started with a value of 400 Trestbps as mentioned previously and decreased the outlier's value until 180. This gave an accuracy or test error that remained approximately close to the original data value without the outlier i.e. till the error matched the unchanged data.

The test error for the outlier was 0.4078947
The test error for the shrink value of outlier was 0.1447368
The test error for the original learning data was 0.1352632

Thus, the significant changes to the network coefficient estimates occur when the outlier value of 400 d ecreases to 180 Trestbps which is a common value in the Trestbps variable and it not considered an outli er value. The single outlier caused a change because the data set observations were not very large.

Question 3

This problem was had to use the OJ data set in the ISLR package. We used the response variable i.e. we are interested in the prediction of “Purchase”. We had to divide the data into test and training with 2/3 training and 1/3 testing.

We had to fit a support vector classifier with varying cost parameters over the range [0.01, 10], which included 0.01, 0.1,1,5, and 10. Afterwards we had to plot the training and test error across this spectrum of cost parameters, as shown in figure

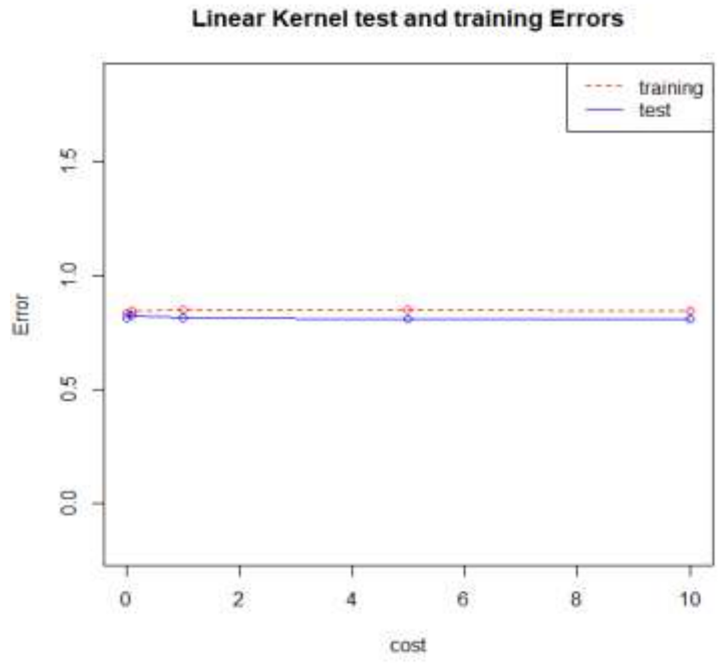


Figure 6: Training and test error across this spectrum of cost parameters for SVM with linear Kernel

From figure 6 we can see that the optimal cost is 5 as it has the least testing and training error.

This was repeated the exercise for a support vector machine with a radial kernel using the default parameter for gamma.

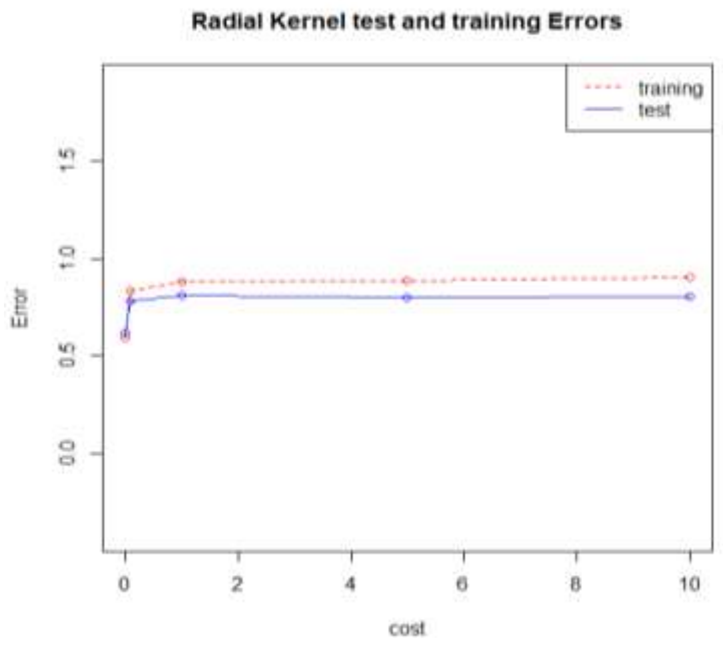


Figure 7: Training and test error across this spectrum of cost parameters for SVM with radial Kernel

From figure 7 we can see that the optimal cost is 0.01 as it has the least testing and training error.

This was also repeated for a support vector machine with a polynomial kernel of degree=2.

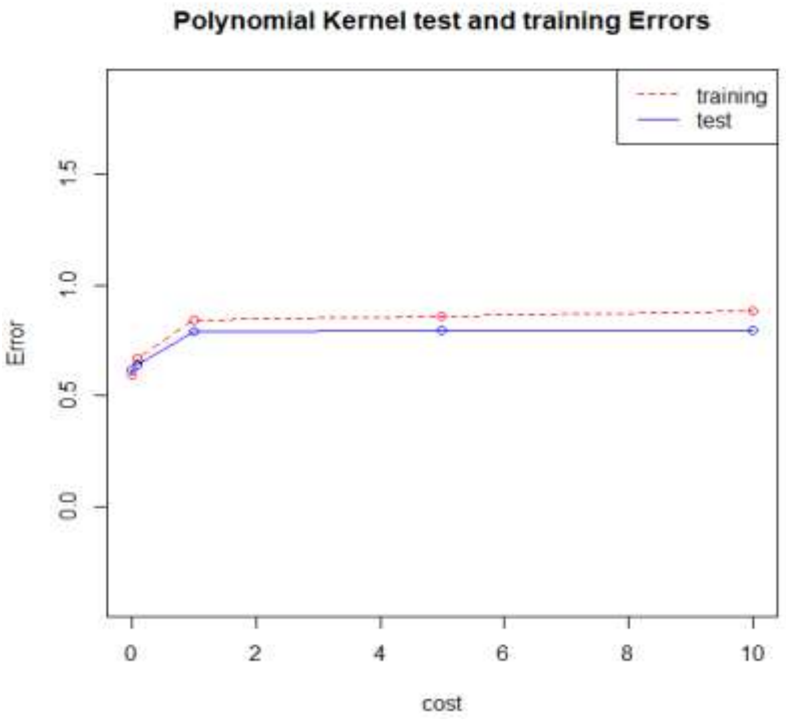


Figure 8: Training and test error across this spectrum of cost parameters for SVM with polynomial kernel of degree 2

From figure 8 we can see that the optimal cost is 0.01 as it has the least testing and training error.

The performance of the support vector classifier, i.e. SVM with a linear kernel we can see that the cost values from 0.01 to 10 are fairly similar in terms of the test and the train error. However with the radial kernel and polynomial kernel degree 2, we can see that 0.01 cost had the lowest test and train error and performs the best. However, the error then increased with cost of 0.1 and remained fairly similar till cost of 10.