1) We use the hold-out method to train the neural network and determine the total number of neurons: The following are the train and test error for the various number of neurons for spam dataset. The train test split has been set at 0.75%

Model	Train Error	Test Error
logistic_accuracy	0.90785279629093	0.919130434782609
1	0.605911330049261	0.606086956521739
2	0.938858301941466	0.940869565217391
3	0.942915097073312	0.934782608695652
4	0.944943494639235	0.934782608695652
5	0.951608229498696	0.940869565217391

We can see that the accuracy of the neural network with more than 1 neuron is quite better than an additive model such as logistic regression but as the neural network is a black box we will not be able to interpret the model result and take actions at a dependent variable level as to which ones might affect the result the most. So a neural network will give us a better solution with no interpretation. But logistic regression will give a good solution with the interpretation of variables. As spam is a standalone classifier in which we don't need to understand parameters, neural network is the better solution.

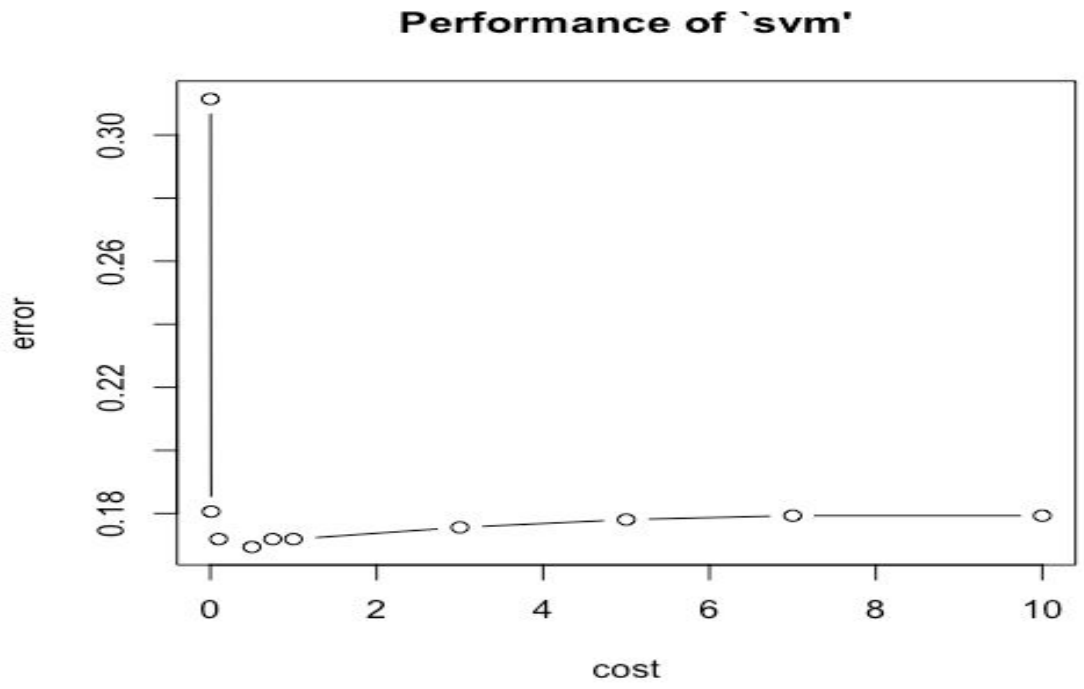
2) We can clearly see that the accuracy for the dataset without the outlier is much better as minimum test error is about 22% while for outlier it is 27% at higher complexity of the model. The following are the tables for errors of number of neurons and the train and test error for each of the dataset. The initial outlier was about 100 times the maximum value of the column. At each iteration, we can see that we have decreased it by value by ten times. We can see that when the outlier is less than 2 times the max, it's effect is not felt.

	Data without Outlier		Data with Outlier	
#Neurons	Train Error	Test Error	Train Error	Test Error
1	0.2204301	0.2580645	0.3172043	0.3870968
2	0.2258065	0.2580645	0.2956989	0.3709677
3	0.2150538	0.2741935	0.1881720	0.3387097
4	0.2096774	0.2258065	0.1612903	0.3548387
5	0.3064516	0.3548387	0.1881720	0.2741935

#Size of outlier w.r.t. the max of the column	Train error	Test Error
10	0.2043011	0.2580645
5	0.1989247	0.3225806
3.3	0.1881720	0.3064516
2.5	0.2311828	0.2419355
2	0.2096774	0.3387097
1.66	0.2043011	0.2419355
1.4	0.2096774	0.2580645
1.25	0.2150538	0.2580645
1.1	0.2150538	0.2741935
1	0.2096774	0.2903226

3) a) The dataset has been divided into test and train split. The split has been taken at 0.75

b) The following are the plots of the cost parameters for the linear kernel or the support vector classifier. As we can see the best c value is : 0.5



c) The following are the error values for linear, radial and degree 2 polynomial for various cost values:

Cost	Error-Radial	Error-Linear	Error-2deg poly
0.001	0.389892	0.3114198	0.3897685
0.01	0.389892	0.1806019	0.3674383
0.1	0.195571	0.1718673	0.311358
0.5	0.1831327	0.1693673	0.2079475
0.75	0.1844136	0.1718673	0.210463
1	0.1843981	0.1718673	0.2129321
3	0.1893673	0.1755864	0.1993056
5	0.1918827	0.1780556	0.1980864
7	0.1956327	0.1793056	0.1980556
10	0.1906481	0.1793056	0.1955864

The accuracy is highest for a radial kernel at 0.8614 while it is lowest for linear kernel with 0.8352. The 2 degree poly has an accuracy of 0.8576