1.) Divided the prostate data into test and training data

Training data is 75% of prostate data.

The rest is testing data.

Performed simple linear regression on the data. (“hold out method”).

The test error obtained is 0.3827086.
The train error obtained is 0.476308.

Akaike's an Information Criterion (AIC) for this is 170.9254.
BIC fit is 193.6921

holdout method over a spectrum of complexity parameters:

The model with 3 feature selection is the best for cp

The model with 3 feature selection is the best for bic

The train error values obtained are 1.1837980, 1.1836755, 0.8284743, 0.7114448, 0.7020195, 0.6835290, 0.0000000, 0.0000000(for 8 models)

The test error values obtained are 0.7721009, 0.7709441, 0.5863952, 0.6283031 0.6291550, 0.60 93629 0.0000000, 0.0000000(for 8 models)

bootstrap .632 estimates of prediction error:

The errors obtained are 0.6379466, 0.5684479, 0.5213495, 0.5314069, 0.5258111, 0.5248033, 0.5260925, 0.5289883(for 8 models)

The model with 3 feature selection is best using bootstrap.

10-fold CV for model selection:

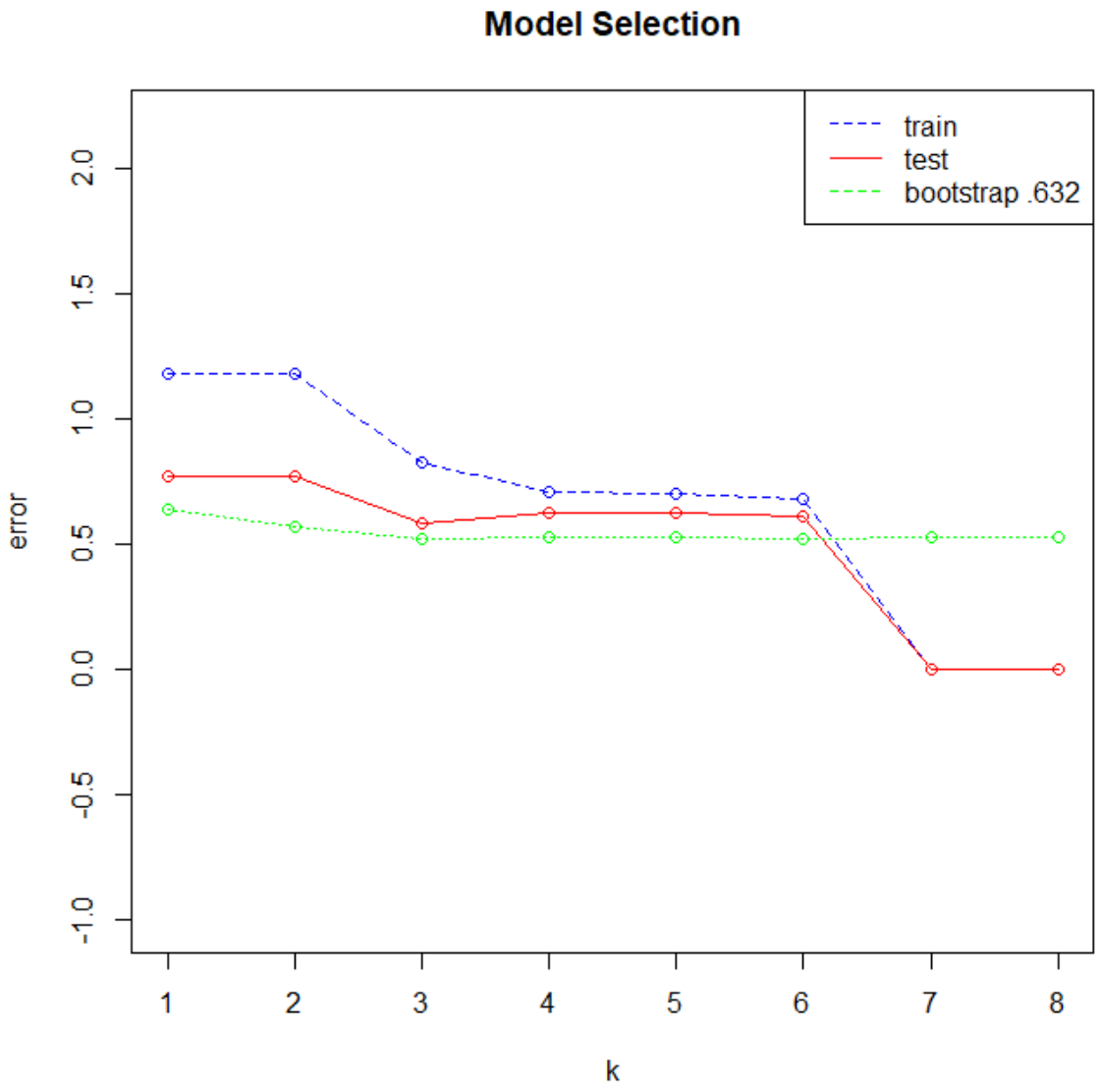
The errors obtained are 0.8070822 0.7626617 0.7488568 0.7893065 0.7672967 0.757221 1 0.7377358 0.7382577 (for 8 models)

The model with 7 feature selection is best using 10-fold CV method.

5-fold CV for model selection:

The errors obtained are 0.7796366, 0.7569648, 0.7330065, 0.7689338, 0.7655337 0.7604827 0.7455869 0.7514670 (for 8 models)

The model with 3 feature selection is best using 5-fold CV method.




The following are the best subset selection obtained for different methods

AIC: 3

BIC: 3

CP: 3

bootstrap .632: 3

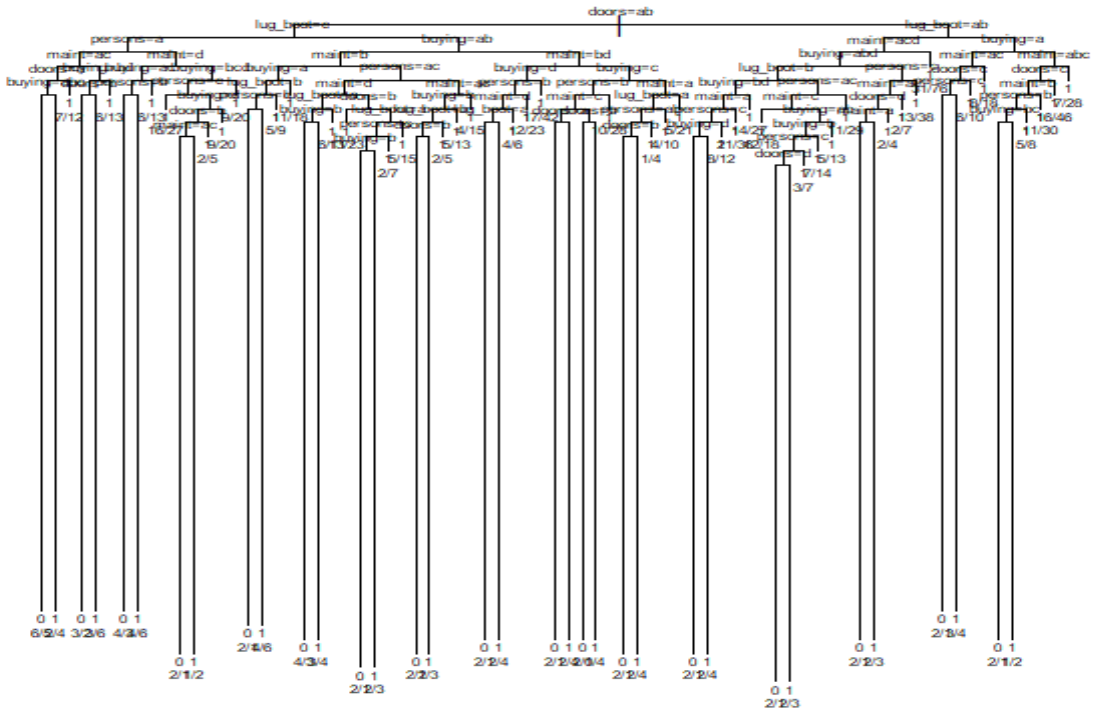
tenfold cross-validation: 7

fivefold cross-validation: 3

From all the above information we can infer that 3 feature subset gives the best results.

3.) I have taken car evaluation dataset from kaggle and applied bagging, boosting, and random forests on this dataset.

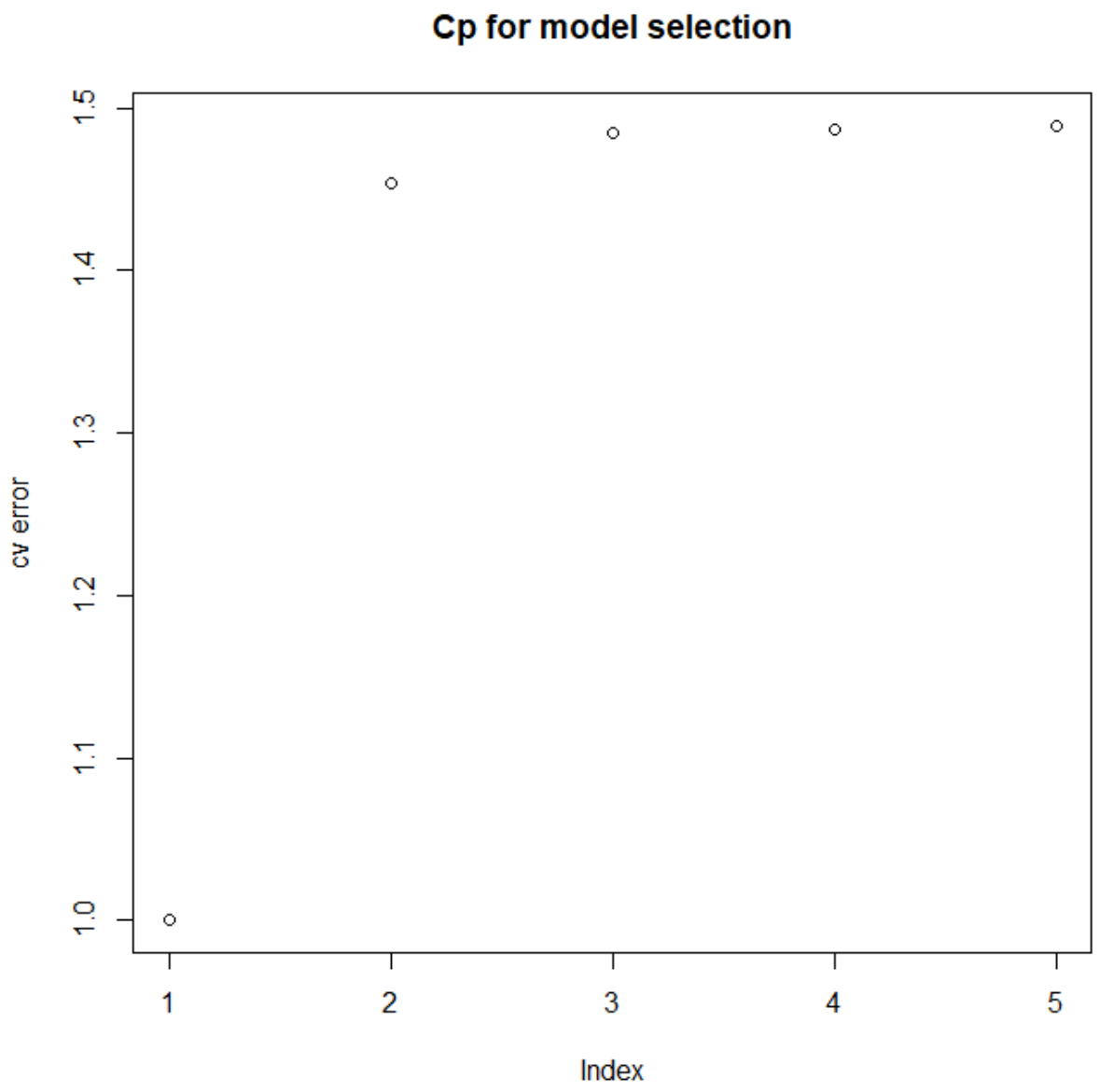
First growing a single tree:




Minimum cp value obtained is for 1

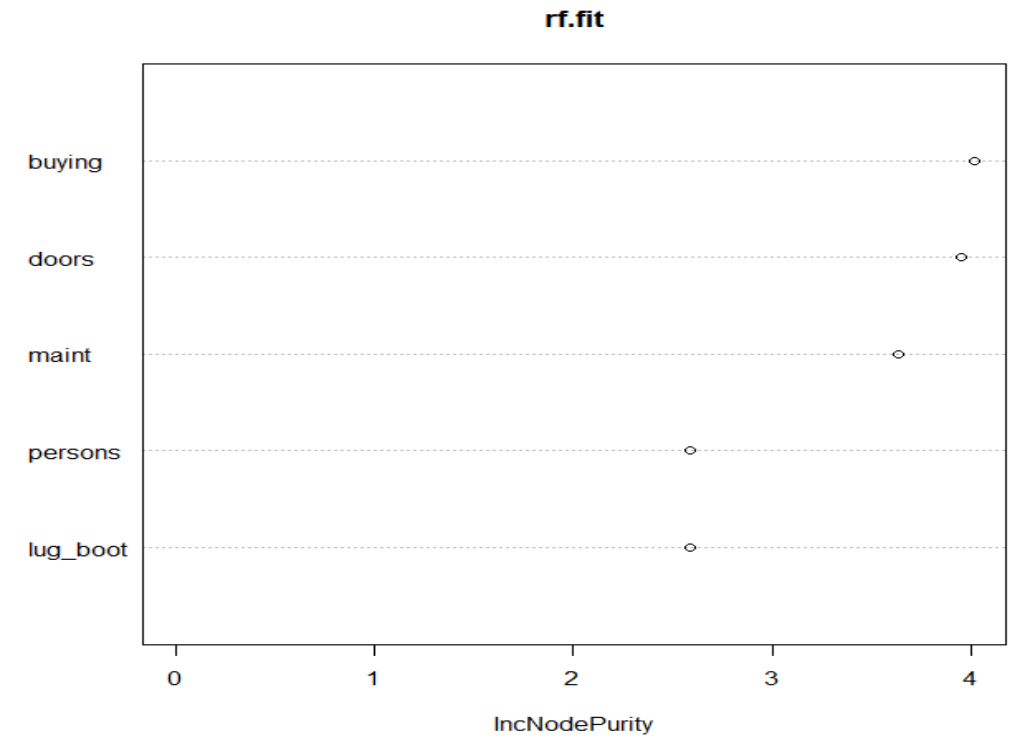
The test error for a single tree obtained is `0.3549884`.

The graph of cv error is:



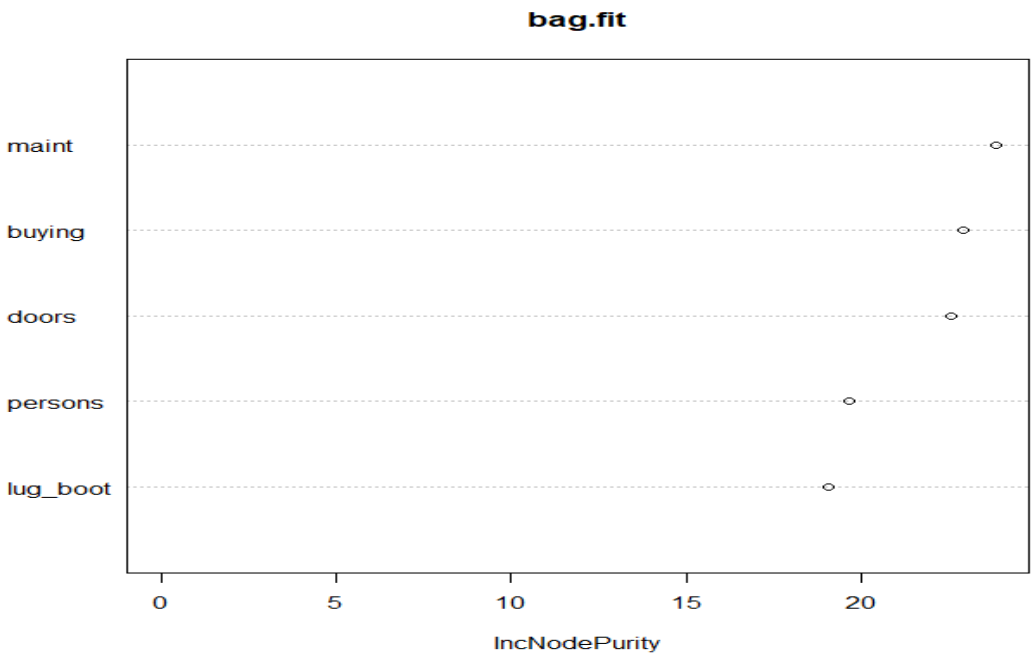

Random forests:

The error obtained by this method is `0.3549884`




Bagging:

The error obtained by this method is `0.3549884`
The corresponding plot for this is



Boosting:

The error obtained for boosting with shrinkage 0.1 is `0.4889897`
The error obtained for boosting with shrinkage 0.6 is `0.5166153`

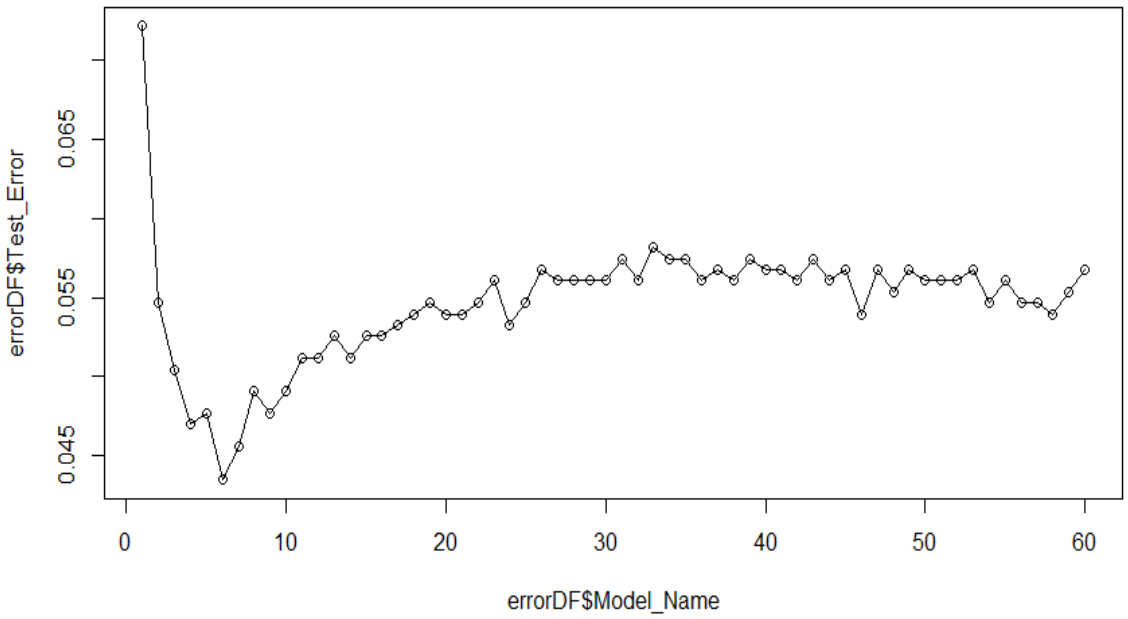
Logistic regression:

The error obtained by this method is `0.4889897.`

It can be clearly observed that the results are more accurate for bagging, boosting, and random forests than a simple logistic regression.

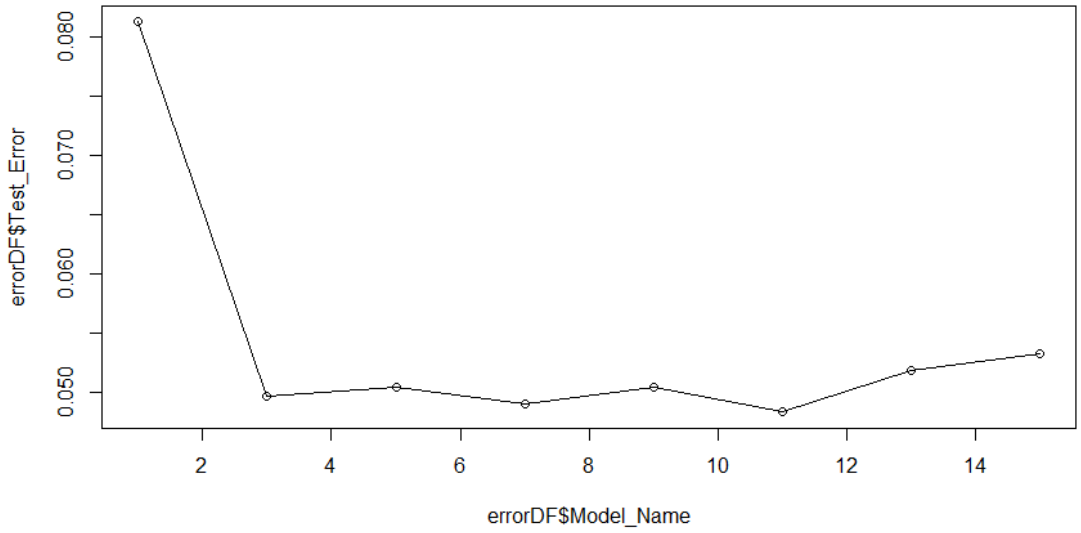
4.) observed that there are no NA values. Now Splitting the data into training set and testing set.

Test error plot for the values of m in the range 1 to 57:




It can be observed that the test error it not changing much for values of m>15.

Test error plot for the values of m in the range 1 to 15 with a step size 2:



Random Forest models have been built using 100 trees. Plotting Out of Bag errors for different values of m gives us the following figure.

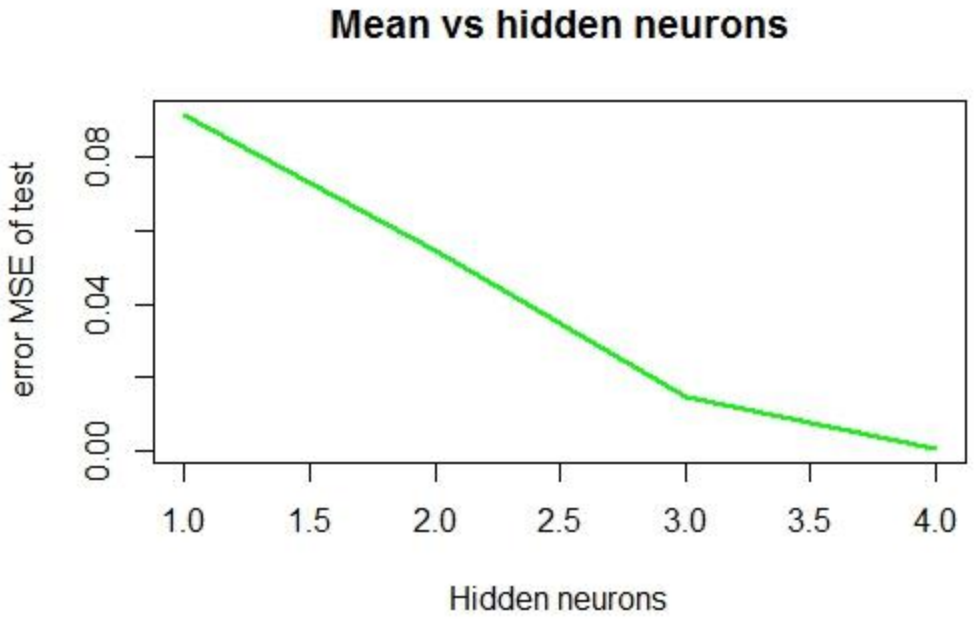

From the above figure we can infer that with the increase in the number of features, error decreases.

5.) fitting a neural network to the spam data.

Split the dataset into training and testing data set.

Run a loop to find the mean of error through cross validation

The plot obtained is:

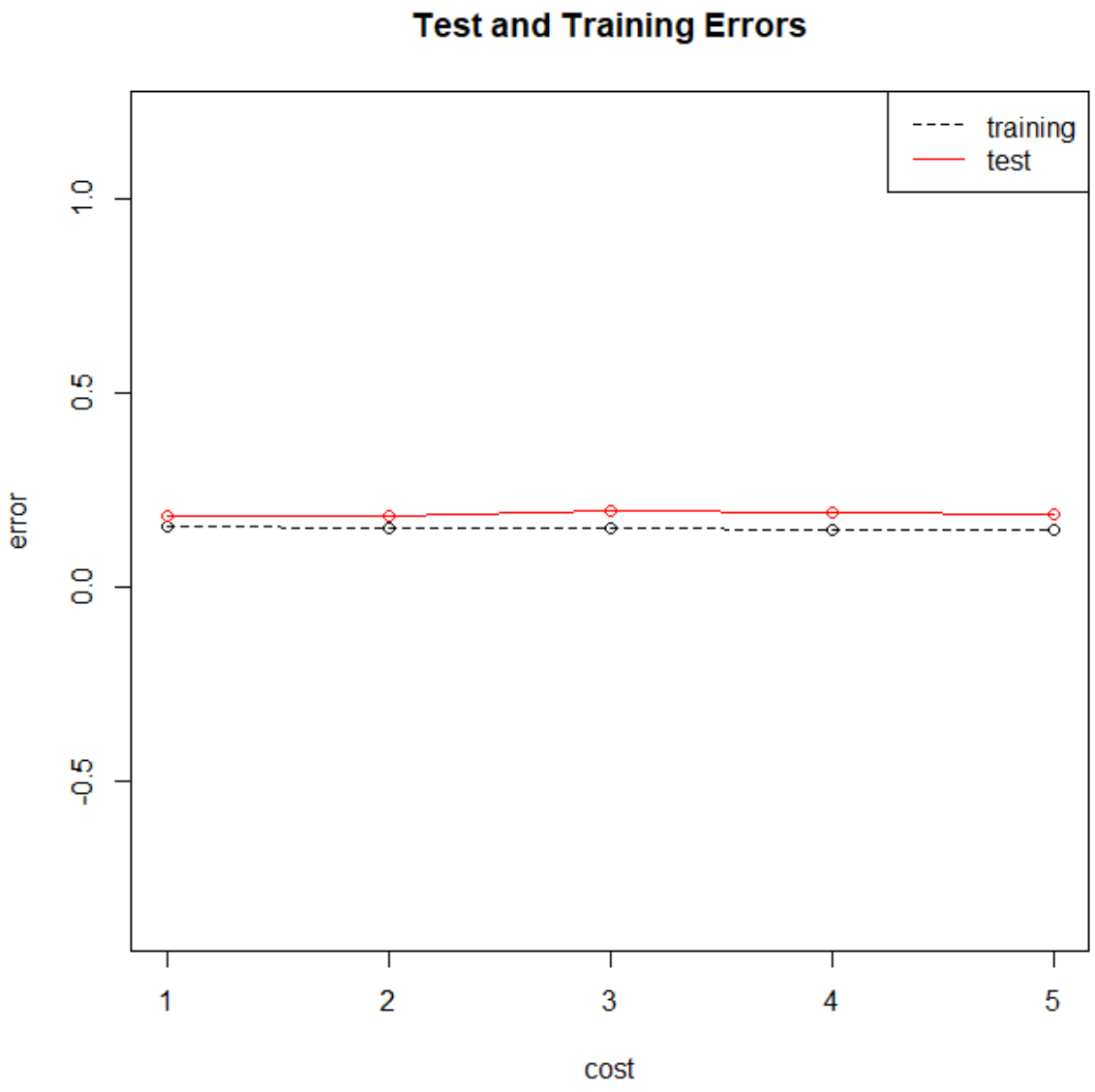



Error obtained from the model of neural network is 26.40

Error obtained from the additive model is 0.037

7.) Splitting the data into training and testing set. Testing set is 35% of the data and the rest is training set.

SVM with a Linear Kernel:



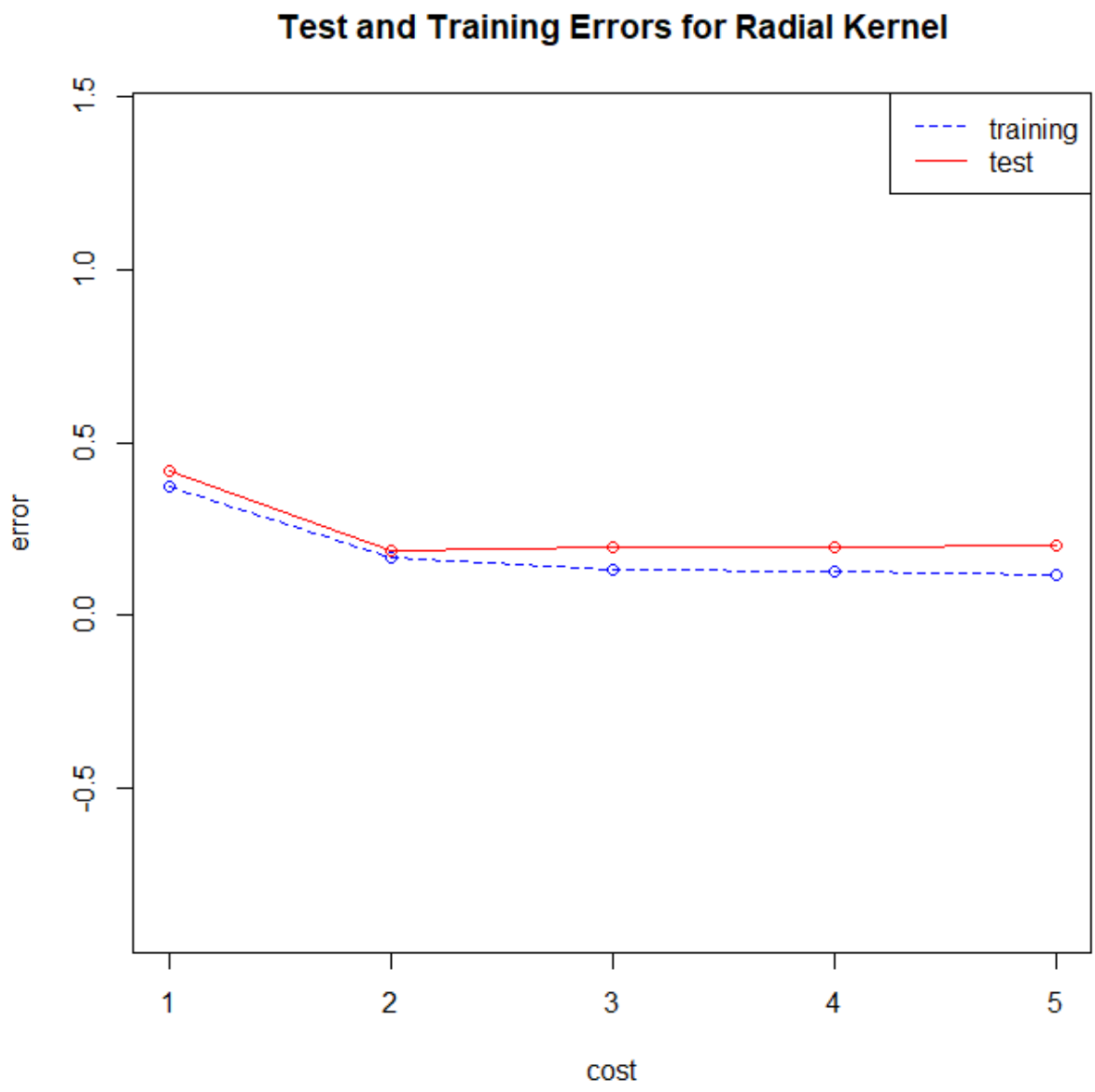

Fitted a support vector classifier with varying cost parameters over the range 0.01 to 10.

A cost of 1 seems to perform best.

We may see that the optimal cost is 0.1.

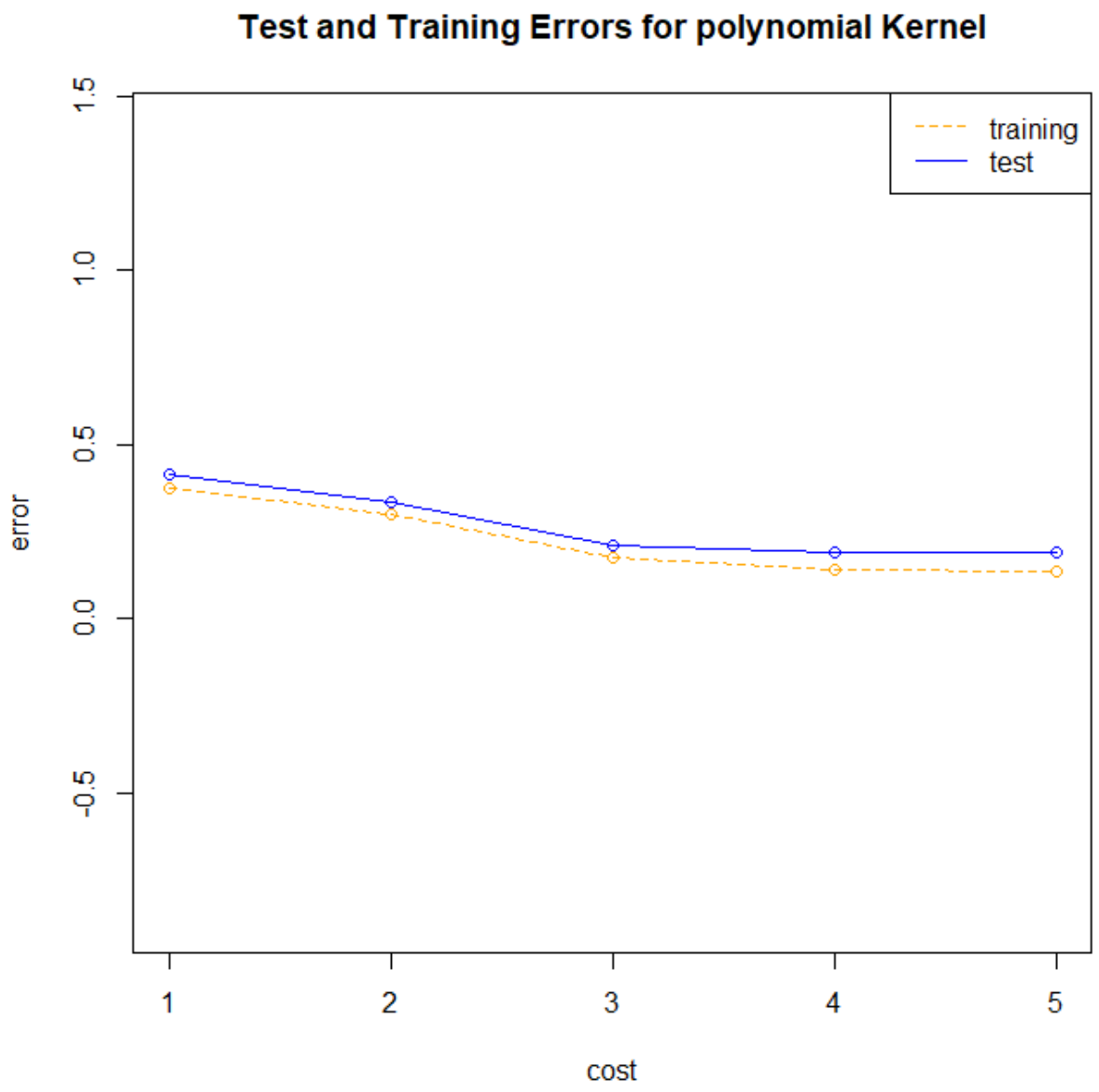
The test errors obtained are `0.1844920 0.1844920 0.1951872 0.1925134 0.1898396`
The train errors obtained are `0.1566092 0.1508621 0.1522989 0.1465517 0.1494253`

SVM with a Radial Kernel:




The test errors obtained are `0.4171123 0.1871658 0.1978610 0.1951872 0.2005348`
The train errors obtained are `0.3750000 0.1695402 0.1321839 0.1264368 0.1192529`

SVM with a Polynomial Kernel:




---

The least error is found at the cost of 3.

The test errors obtained are `0.4171123 0.3342246 0.2112299 0.1898396 0.1898396`
The train errors obtained are `0.3750000 0.2988506 0.1738506 0.1436782 0.1350575`