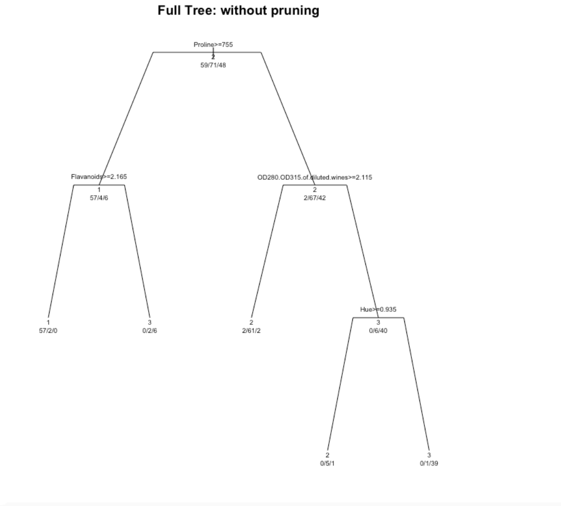
STATISTICAL DATA MINING
HOMEWORK 4

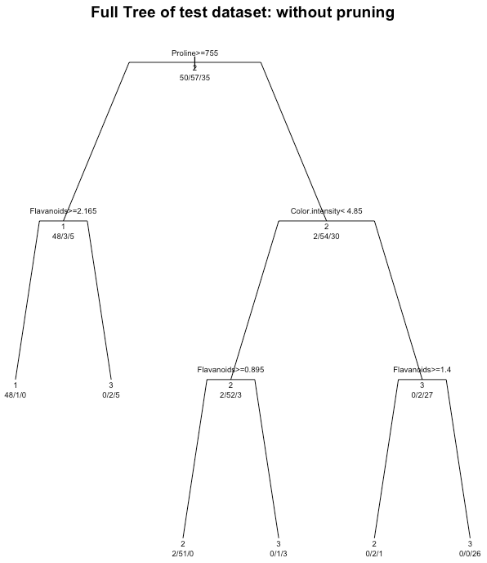
NAME: PRIYA MURTHY
UB PERSON NUMBER:50248887
CLASS NUMBER:53

ANSWER 2

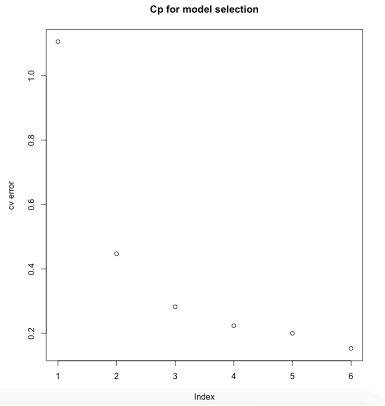
The figure below represents the classification tree for the whole wine dataset.
To interpret the data, we look at the values in the figure.

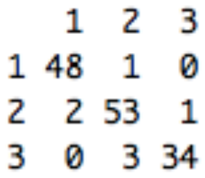


- The value a/b/c below each node represents the number of variables of each class which lies in that node. We have three classes in our wine dataset which represents the 3 cultivators namely: Barolo, Grignolino, and Barbera.
- The value at node 1 is 57/4/6 which means we have 57 cases of Barabera, 4 cases of Barolo and 6 cases of Grignolino under that category.
- The figure below shows the tree for the test dataset. The interpretation of the tree is done in the same way as described above.



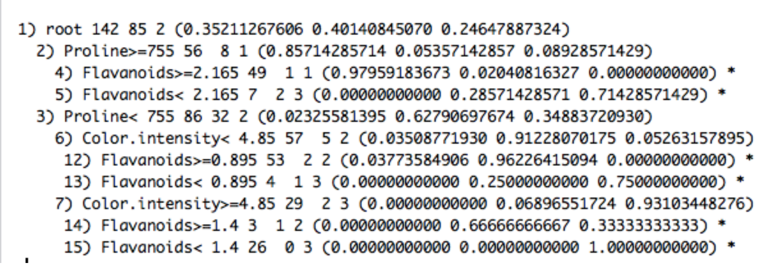
- The error for the complexity parameter for the training model is represented below:





	1	2	3
1	48	1	0
2	2	53	1
3	0	3	34

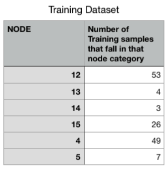
- <- Table represents how the tree has categorized the test dataset.
- The recursive partitioning of the training data resulted in a tree with the nodes as shown below:



```
1) root 142 85 2 (0.35211267606 0.40140845070 0.24647887324)
  2) Proline>=755 56  8 1 (0.85714285714 0.05357142857 0.08928571429)
    4) Flavanoids>=2.165 49  1 1 (0.97959183673 0.02040816327 0.00000000000) *
    5) Flavanoids< 2.165 7  2 3 (0.00000000000 0.28571428571 0.71428571429) *
  3) Proline< 755 86 32 2 (0.02325581395 0.62790697674 0.34883720930)
    6) Color.intensity< 4.85 57  5 2 (0.03508771930 0.91228070175 0.05263157895)
     12) Flavanoids>=0.895 53  2 2 (0.03773584906 0.96226415094 0.00000000000) *
     13) Flavanoids< 0.895 4  1 3 (0.00000000000 0.25000000000 0.75000000000) *
    7) Color.intensity>=4.85 29  2 3 (0.00000000000 0.06896551724 0.93103448276)
     14) Flavanoids>=1.4 3  1 2 (0.00000000000 0.66666666667 0.33333333333) *
     15) Flavanoids< 1.4 26  0 3 (0.00000000000 0.00000000000 1.00000000000) *
```

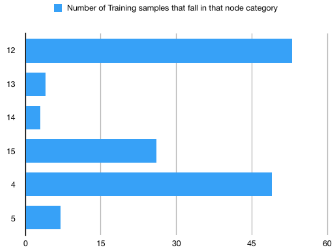
- Out of all the nodes, the test and training data samples are assigned different nodes.

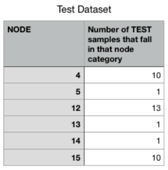
- The figure below shows a pictorial representation of the training and test samples that fall under different nodes of the classification tree. The approach used is the same as described in the steps above: We created a tree using the training data and predicted the categories in which the test samples will fall into using the trained model.



Training Dataset

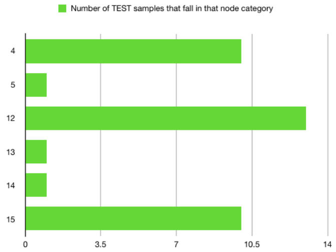
NODE	Number of Training samples that fall in that node category
12	53
13	4
14	3
15	26
4	49
5	7





Test Dataset

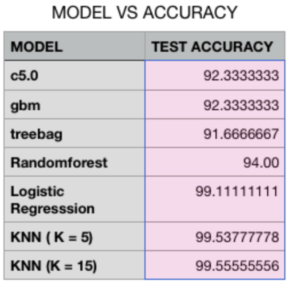
NODE	Number of TEST samples that fall in that node category
4	10
5	1
12	13
13	1
14	1
15	10



ANSWER 3

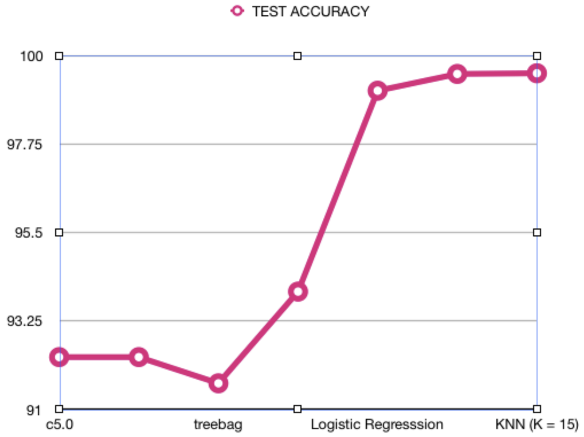
Dataset chose: IRIS

- Divided the dataset into training and test set
- On application of various models on the training dataset and predicting test set, the test accuracy for the various models is shown below.



MODEL VS ACCURACY

MODEL	TEST ACCURACY
c5.0	92.3333333
gbm	92.3333333
treebag	91.6666667
Randomforest	94.00
Logistic Regresssion	99.11111111
KNN (K = 5)	99.53777778
KNN (K = 15)	99.55555556



- **Observation:**
  1) Simplistic (non-ensemble) methods (Logistic regression and KNN) perform better than bagging, boosting, and random forests.

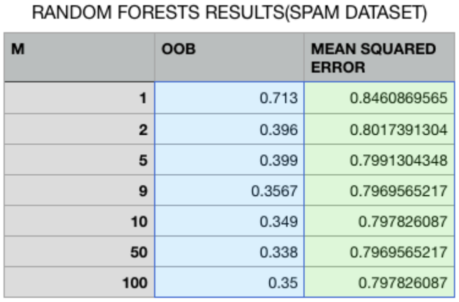
2) KNN with K = 15 has the best accuracy among all the models and C5.0 has the least accuracy.

- Advantages and Disadvantages of Committee machines:
  1. Over training risk is minimized with Committee machines.
  2. The effects of one more more experts training to local minima is reduced.
  3. The error can be reduced by averaging.
  4. One problem could be that these models increase linearly as the data size increases multifold which might cause the result only to be obtained at a later stage.

ANSWER 4

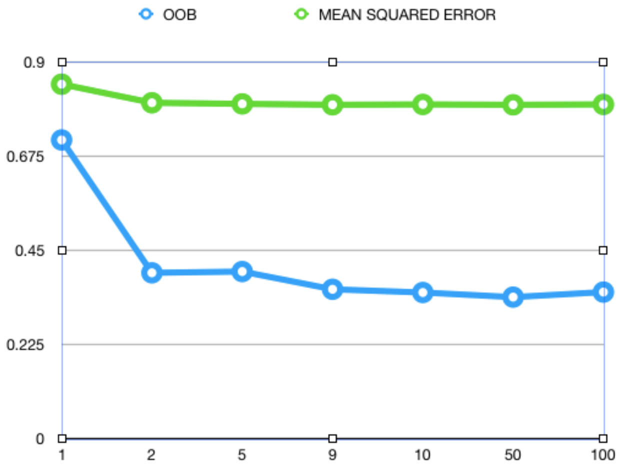
In this question, we applied random forest on train dataset, and found the OOB(training error) and test error for m = 1,2,5,9,10,50,100 where m denotes the number of randomly selected inputs for each tree.

- Created data set and applied random forests and calculated the test and training error for each value of m.
- In the below result table, we can see that the error (both OOB and MSE) are less for higher values of m.

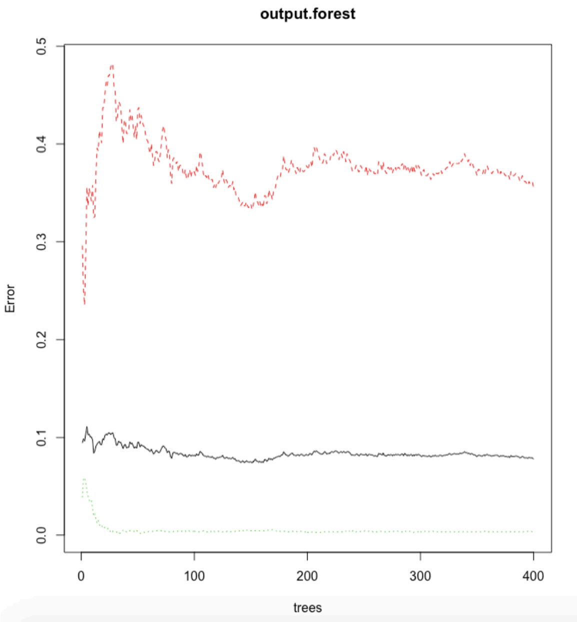


RANDOM FORESTS RESULTS(SPAM DATASET)

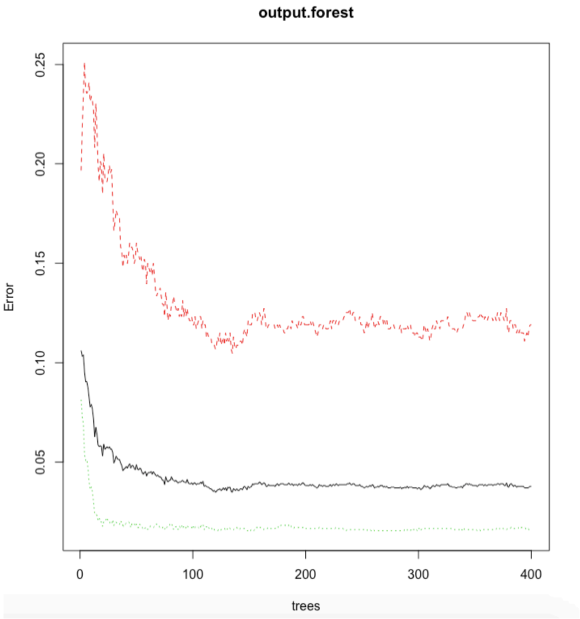
M	OOB	MEAN SQUARED ERROR
1	0.713	0.8460869565
2	0.396	0.8017391304
5	0.399	0.7991304348
9	0.3567	0.7969565217
10	0.349	0.797826087
50	0.338	0.7969565217
100	0.35	0.797826087



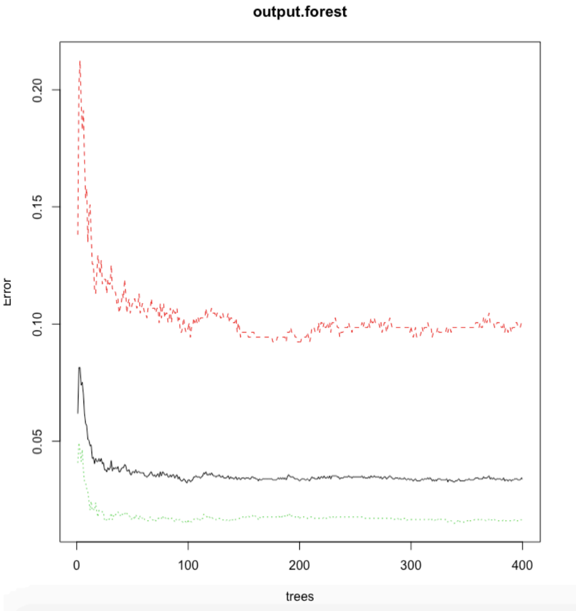
- **Fig: OOB and Mean squared error**



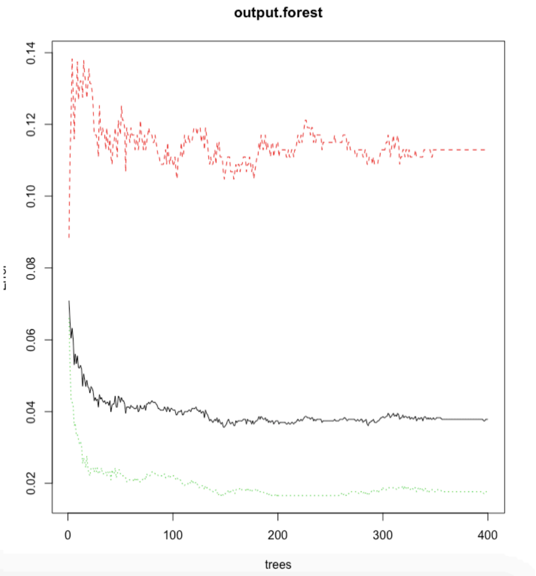
m = 1



m = 2



m = 7

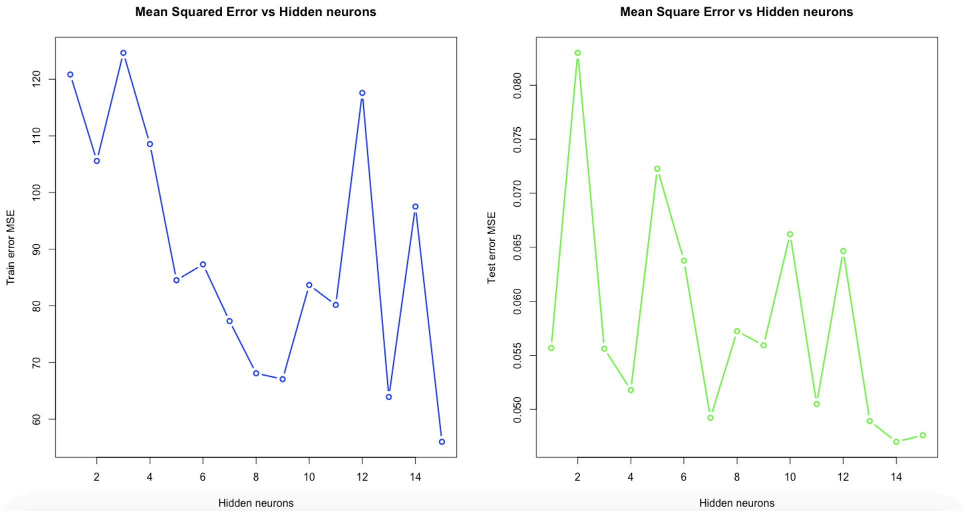


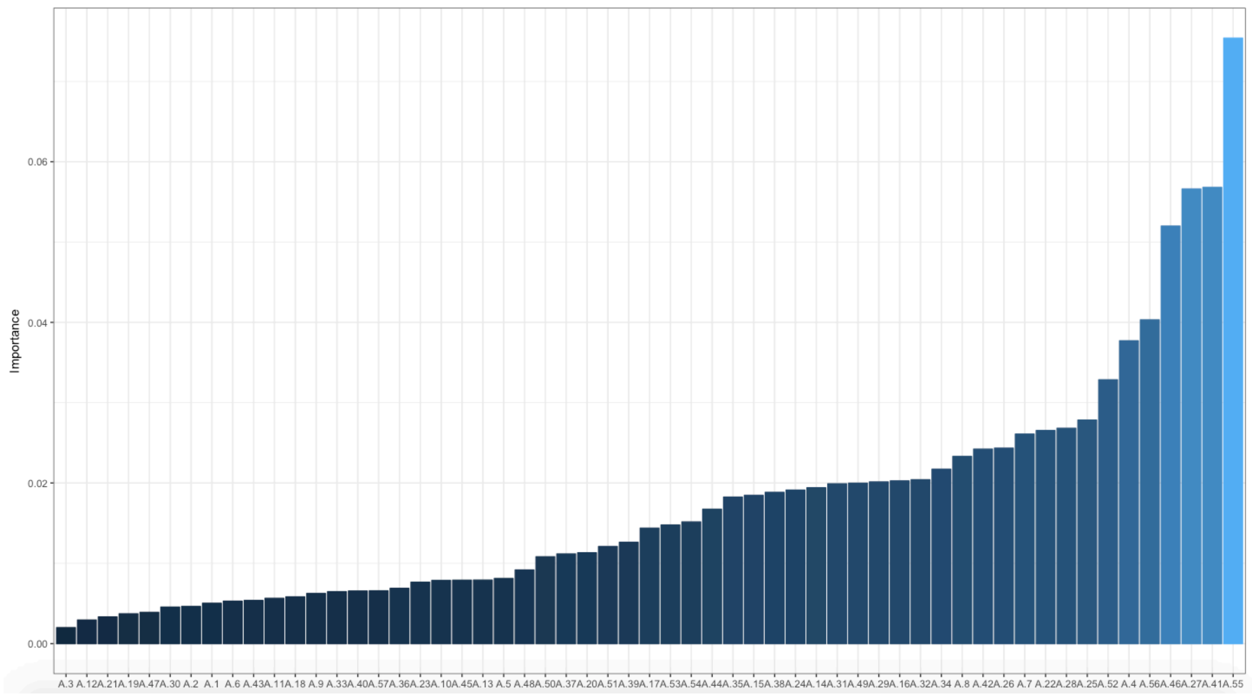
m = 50

The above 4 plots show the result of the random forest vs Error, for different values of m.

ANSWER 5

- In this question, we fit a neural network for spam data available through packet “ElemStatLearn”.
- Hidden variables were chosen by cross validation.
- Figure bellows shows the value of training and test error vs Hidden neurons.
- The train error and test error is least when the value of Hidden neurons is 15.



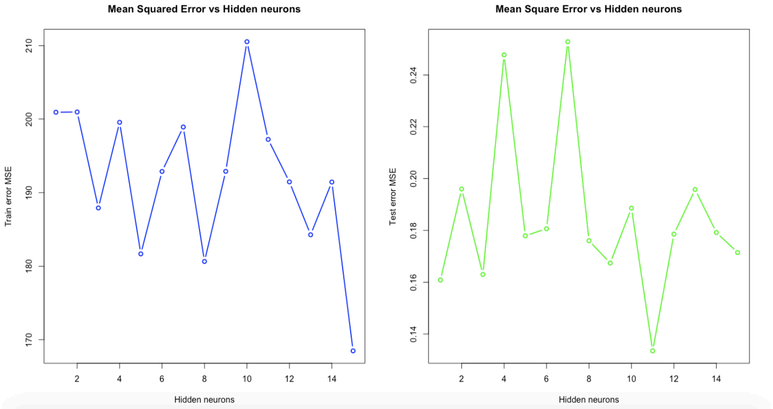


- From the above figure, we can see that the most significant variable according to the model is “capitalAve”.

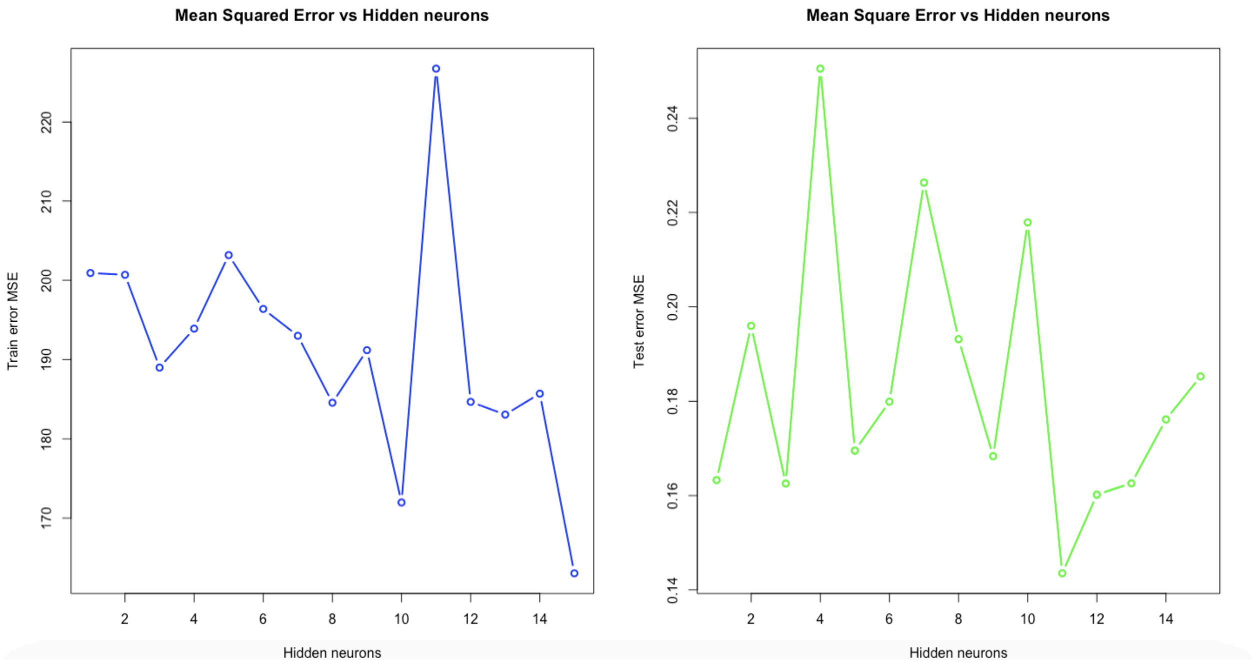
- From the results above we can see that, test error decreases as the hidden neurons are increased but the train error shows variations with respect to hidden neurons. Looking at **'Table 10.1' in section 9.1.2**, we can say that Neural network is good to extract linear combinations from the dataset and is good for prediction.

ANSWER 6

- For this question, the dataset taken is **Heart Disease Data Set** from the machine learning repository.
- The figure below shows the graph for mean squared error vs Hidden neurons for traning and test set.

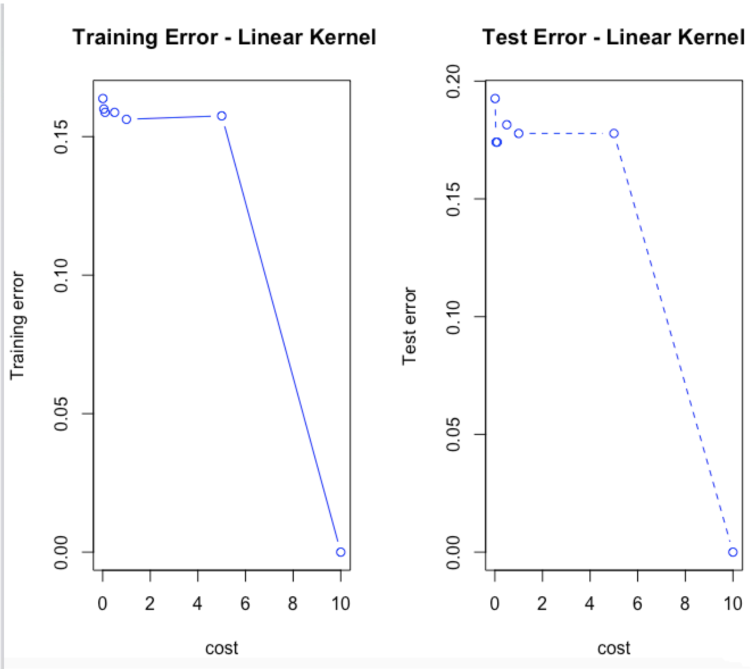


- The test error was minimum for Hidden neuron 11 and train error for 15.
- Now we spike the dataset and check when the effect of the outlier on the fit vanishes.
- I increased the value of column 40 to a large value and the errors obtained are as shown below. The increased value had to be somewhere around 900,000 to get significant change in results obtained.

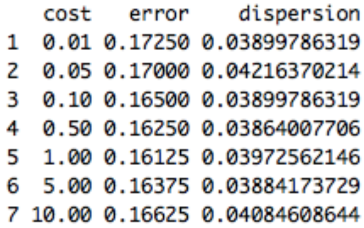


- In this case the minimum test error value increases than what was obtained before.
- Thus, after a few observations and value changes, it was found that as the value of the spiked data is brought closer to the original value, the value of error also decreases.

ANSWER 7(A)

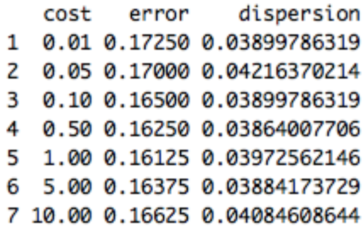


Detailed Analysis for Training error: SVM linear kernel



```
   cost   error    dispersion
1  0.01 0.17250 0.03899786319
2  0.05 0.17000 0.04216370214
3  0.10 0.16500 0.03899786319
4  0.50 0.16250 0.03864007706
5  1.00 0.16125 0.03972562146
6  5.00 0.16375 0.03884173729
7 10.00 0.16625 0.04084608644
```

Detailed Analysis for Test error: SVM linear kernel

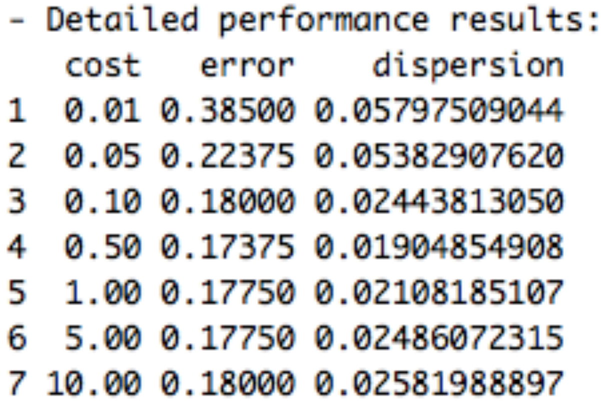


```
   cost   error    dispersion
1  0.01 0.17250 0.03899786319
2  0.05 0.17000 0.04216370214
3  0.10 0.16500 0.03899786319
4  0.50 0.16250 0.03864007706
5  1.00 0.16125 0.03972562146
6  5.00 0.16375 0.03884173729
7 10.00 0.16625 0.04084608644
```

The above plot and Analysis shows the training and test error for a Support vector classifier with varying cost parameters over the range [0.01, 10] and linear kernel.
We can see that the test error is minimum when the cost is 1 and is maximum when the cost is 0.01.

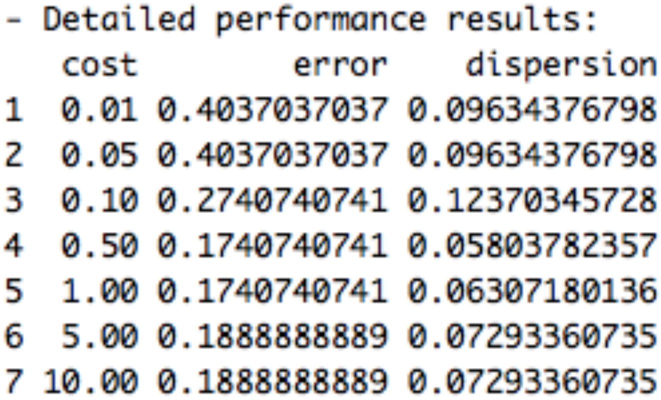
ANSWER 7(B)

Detailed Analysis for Training error: SVM radial kernel

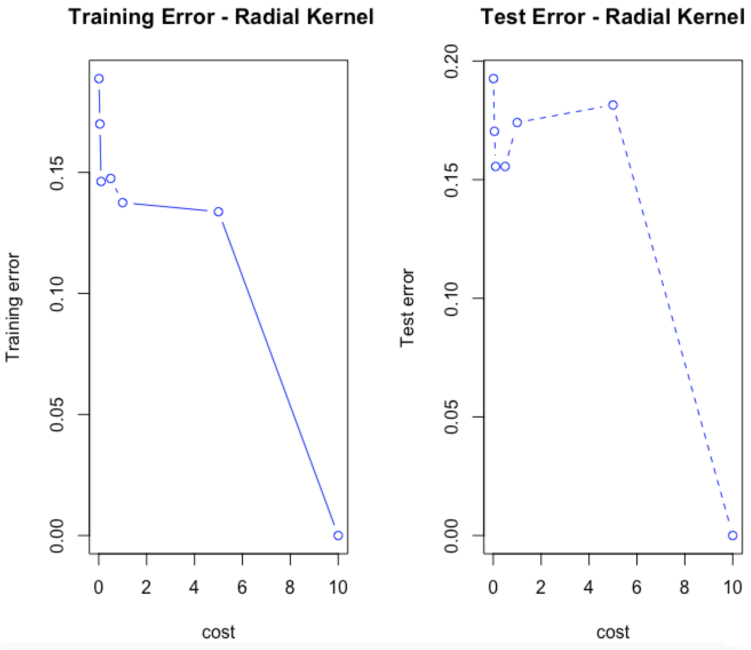


```
- Detailed performance results:
   cost   error    dispersion
1  0.01 0.38500 0.05797509044
2  0.05 0.22375 0.05382907620
3  0.10 0.18000 0.02443813050
4  0.50 0.17375 0.01904854908
5  1.00 0.17750 0.02108185107
6  5.00 0.17750 0.02486072315
7 10.00 0.18000 0.02581988897
```

Detailed Analysis for Test error: SVM radial kernel



```
- Detailed performance results:
   cost         error    dispersion
1  0.01 0.4037037037 0.09634376798
2  0.05 0.4037037037 0.09634376798
3  0.10 0.2740740741 0.12370345728
4  0.50 0.1740740741 0.05803782357
5  1.00 0.1740740741 0.06307180136
6  5.00 0.1888888889 0.07293360735
7 10.00 0.1888888889 0.07293360735
```



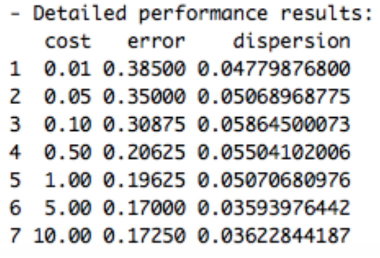
The above plot and Analysis shows the training and test error for a Support vector machine with varying cost parameters over the range [0.01, 10] with radial kernel.

We can see that the test error is minimum when the cost is 1 and 0.5 and is maximum when the cost is 0.01.

This almost has similar behavior to the Support vector machine with linear kernel. Error decreases as the cost increases.

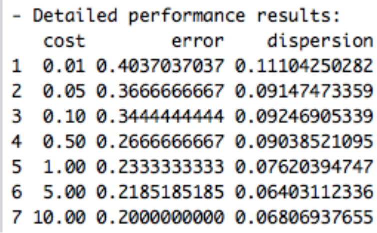
POLYNOMIAL KERNEL OF DEGREE 2

Detailed Analysis for Training error: SVM polynomial kernel

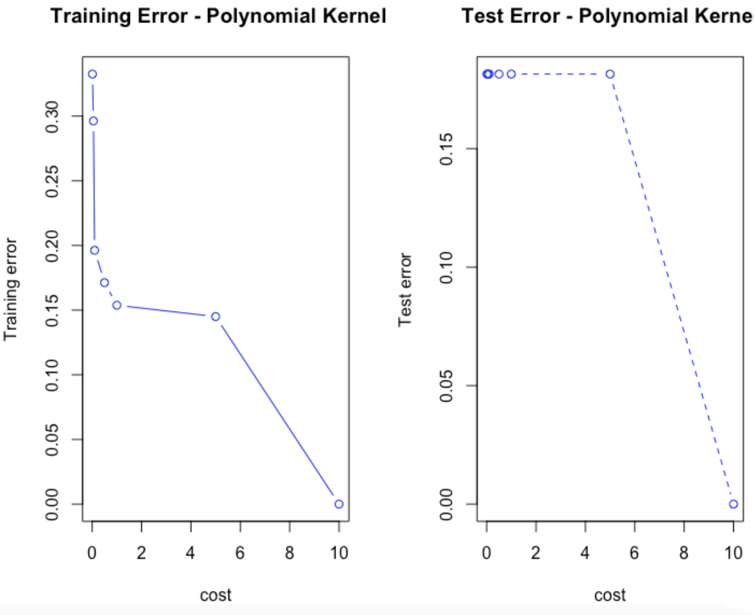


```
- Detailed performance results:
   cost   error   dispersion
1  0.01 0.38500 0.04779876800
2  0.05 0.35000 0.05068968775
3  0.10 0.30875 0.05864500073
4  0.50 0.20625 0.05504102006
5  1.00 0.19625 0.05070680976
6  5.00 0.17000 0.03593976442
7 10.00 0.17250 0.03622844187
```

Detailed Analysis for Test error: SVM polynomial kernel



```
- Detailed performance results:
   cost        error   dispersion
1  0.01 0.4037037037 0.11104250282
2  0.05 0.3666666667 0.09147473359
3  0.10 0.3444444444 0.09246905339
4  0.50 0.2666666667 0.09038521095
5  1.00 0.2333333333 0.07620394747
6  5.00 0.2185185185 0.06403112336
7 10.00 0.2000000000 0.06806937655
```



The above plot and Analysis shows the training and test error for a Support vector machine with varying cost parameters over the range [0.01, 10] with polynomial kernel of degree 2.

We can see that the test error is minimum when the cost is 10 and is maximum when the cost is 0.01.

This almost has similar behavior to the Support vector machine with polynomial kernel of kernel 2. Error decreases as the cost increases.