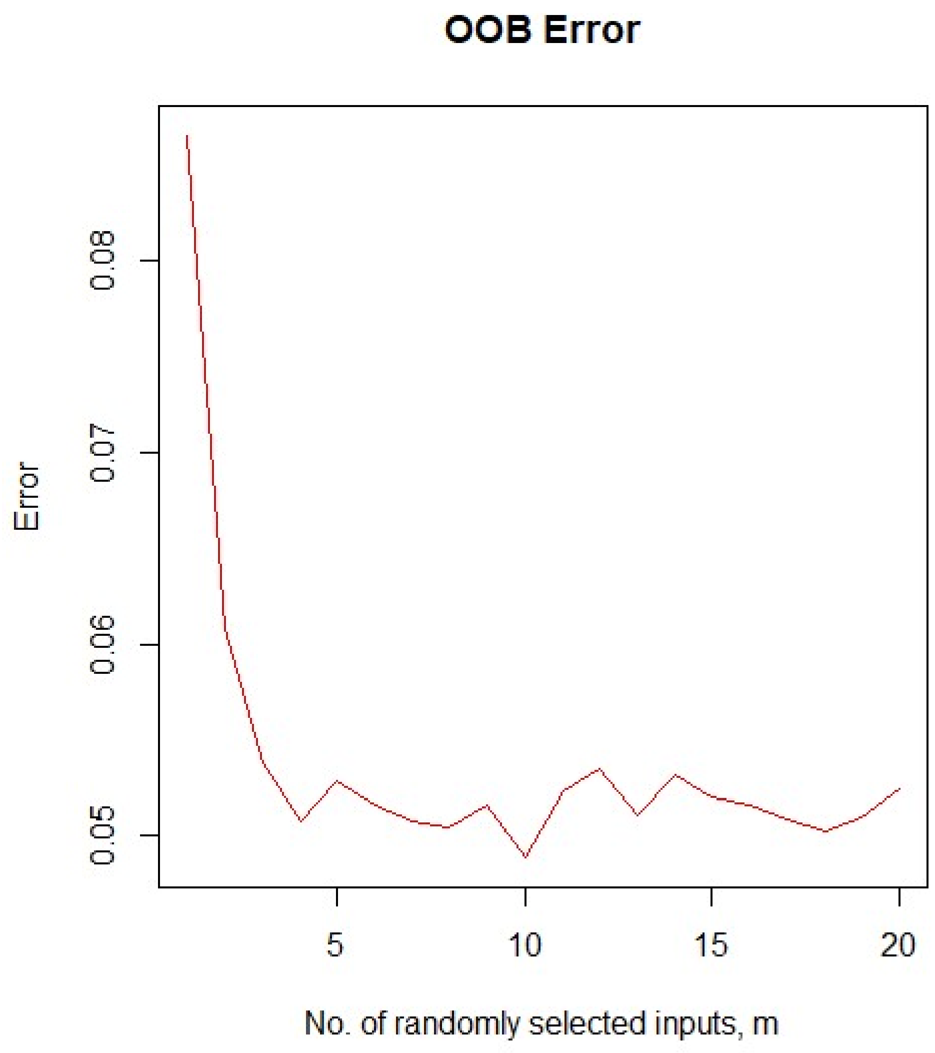
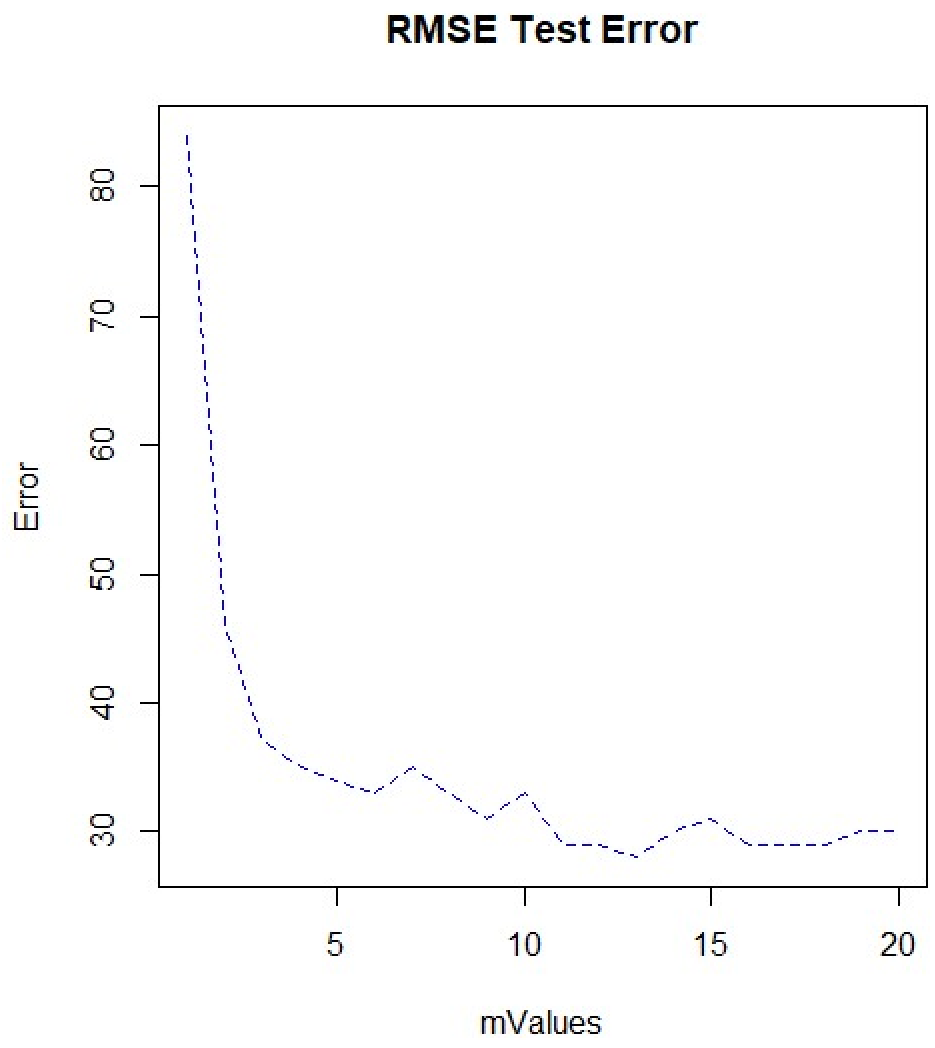
EXERCISE 1

Fit a series of random-forest classifiers to the SPAM data, to explore the sensitivity to m (the number of randomly selected inputs for each tree). Plot both the OOB error as well as the test error against a suitably chosen range of values for m.

In this exercise we are using the spam dataset. The dataset is a collection of unsolicited junk email sent indiscriminately in bulk, often for commercial purposes. The dataset has 4601 observations, and 58 attributes. We will use the random forest model to classify whether a given email is a spam or not.

The data was split into training and test set. Random forest models were fit for increasing m values (the number of randomly selected inputs). Out of Bag error and the Mean Square Error was calculated for each model complexity.


OOB Error
Error
0.05
0.06
0.07
0.08
5
10
15
20
No. of randomly selected inputs, m



Both the OOB Error and the RMSE test Error decrease as m increases. At m > 5, however, this rate of decrease slows. The minimum error occurs at m = 10 for OOB (4.89 %), and m=13, (33) for RMSE test error.

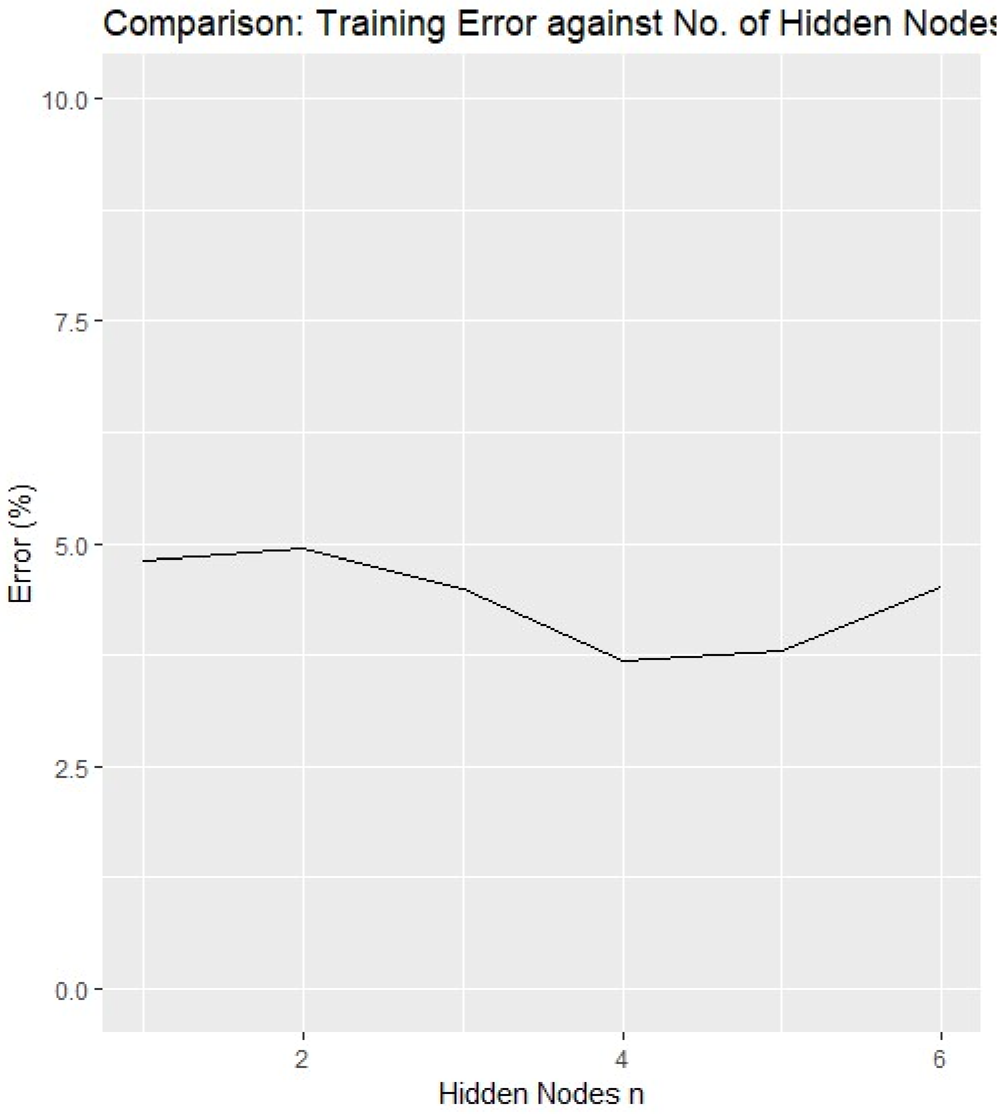
EXERCISE 2

Fit a neural network to the spam data of Section 9.1.2. The data is available through the package ElemStatLearn. Use cross-validation or the hold out method to determine the number of neurons to use in the layer. Compare your results to those for the additive model given in the chapter. When making the comparison, consider both the classification performance and interpretability of the final model.

Using the data-set given to us in the previous question, we fit successive neural network model varying the number of hidden nodes. Cross validation was performed for each model.

In the following figure we can see that the optimum model is with 4 hidden neurons.

Model	**Training Error**	**Test Error**
Optimal (n = 4)	3.70 %	9.34 %



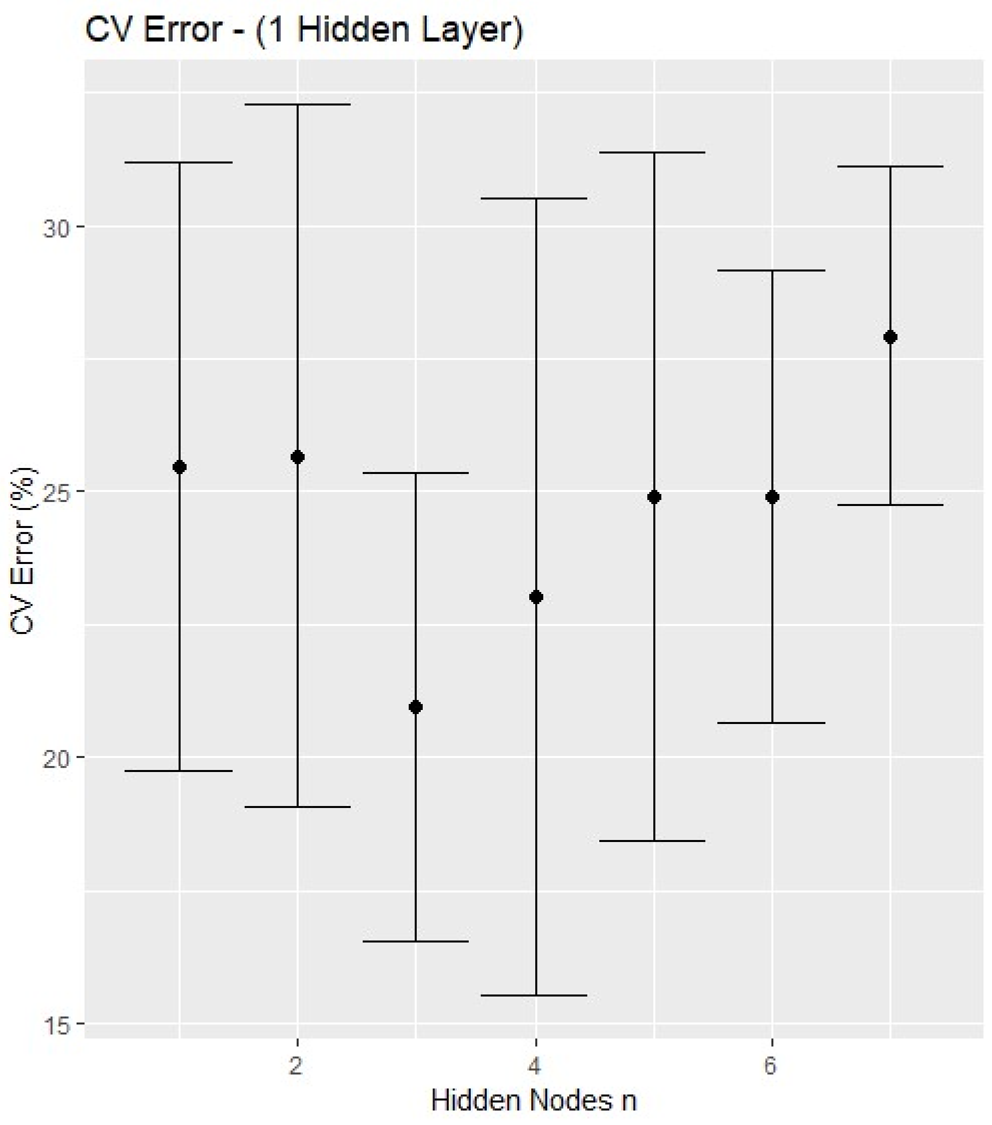
EXERCISE 3

Take any classification data set and divide it up into a learning set and a test set. Change the value of one observation on one input variable in the learning set so that the value is now a univariate outlier. Fit separate single hidden-layer neural networks to the original learning-set data and to the learning set data with the outlier. Use cross-validation or the hold out method to determine the number of neurons to use in the layer. Comment on the effect of the outlier on the fit and on its effect on classifying the test set. Shrink the value of that outlier toward its original value and evaluate when the effect of the outlier on the fit vanishes. How far away must the outlier move from its original value that significant changes to the network coefficient estimates occur?

This exercise was performed on the Pima Indians dataset. The dataset contains 532 observations, with 7 predictors. There are two outcomes, classes.

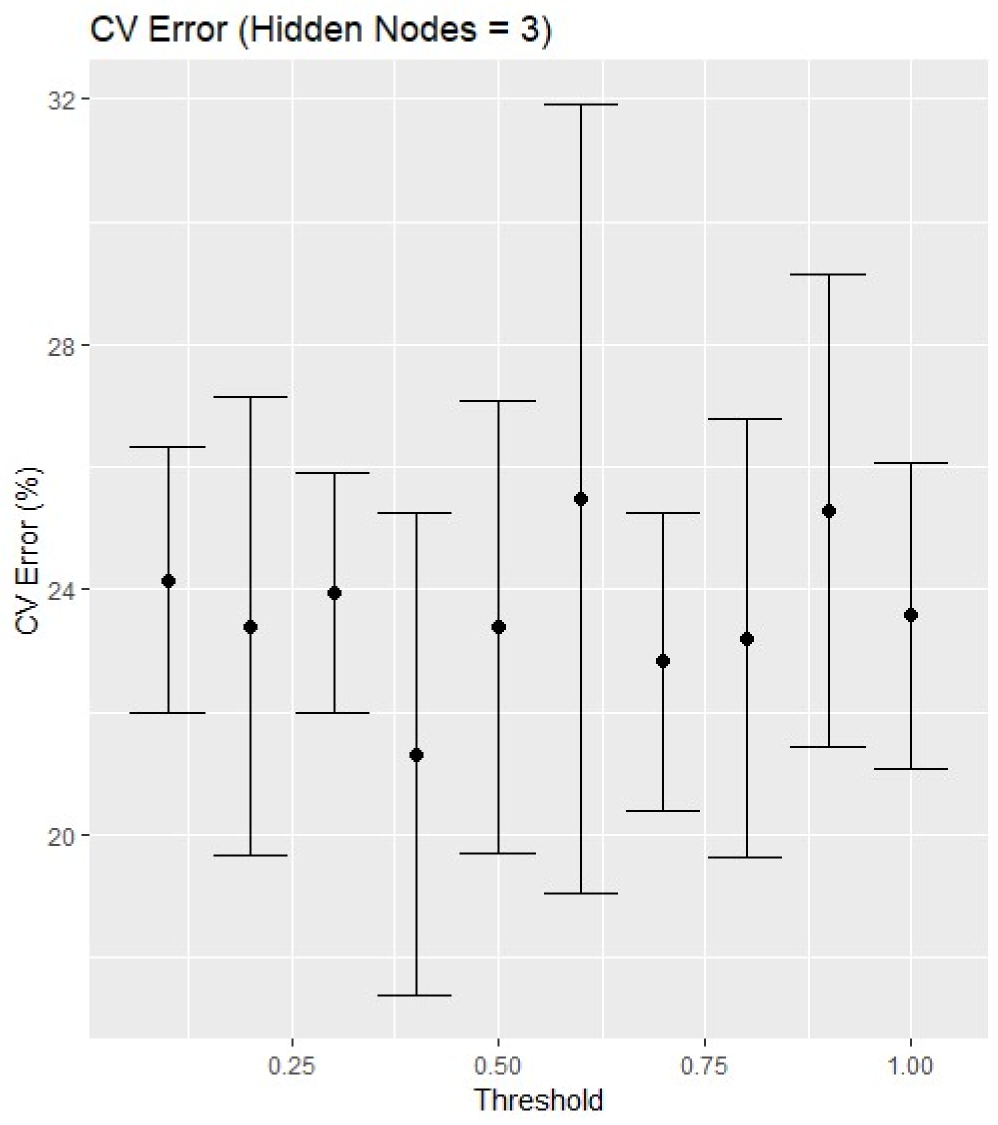
The data was first normalized and rescaled to enhance the performance of the neural network. The class observation was converted into a numeric 0 and 1 representing the outcome.

Neural networks were fitted for varying the complexity parameter, n, the number of hidden nodes (single layer). 10-fold cross validation was performed on the varying model sizes.



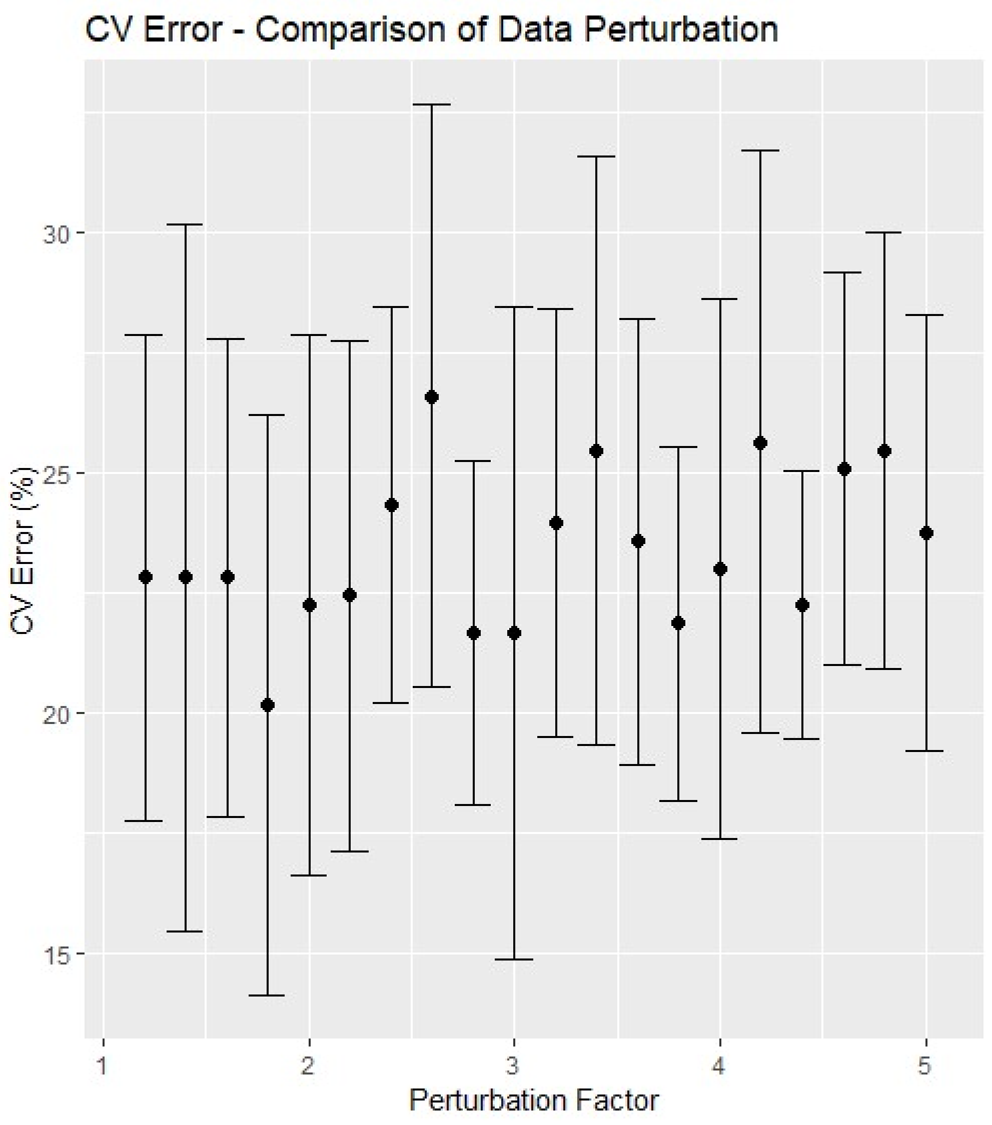
The model with 3 hidden nodes performed the best.

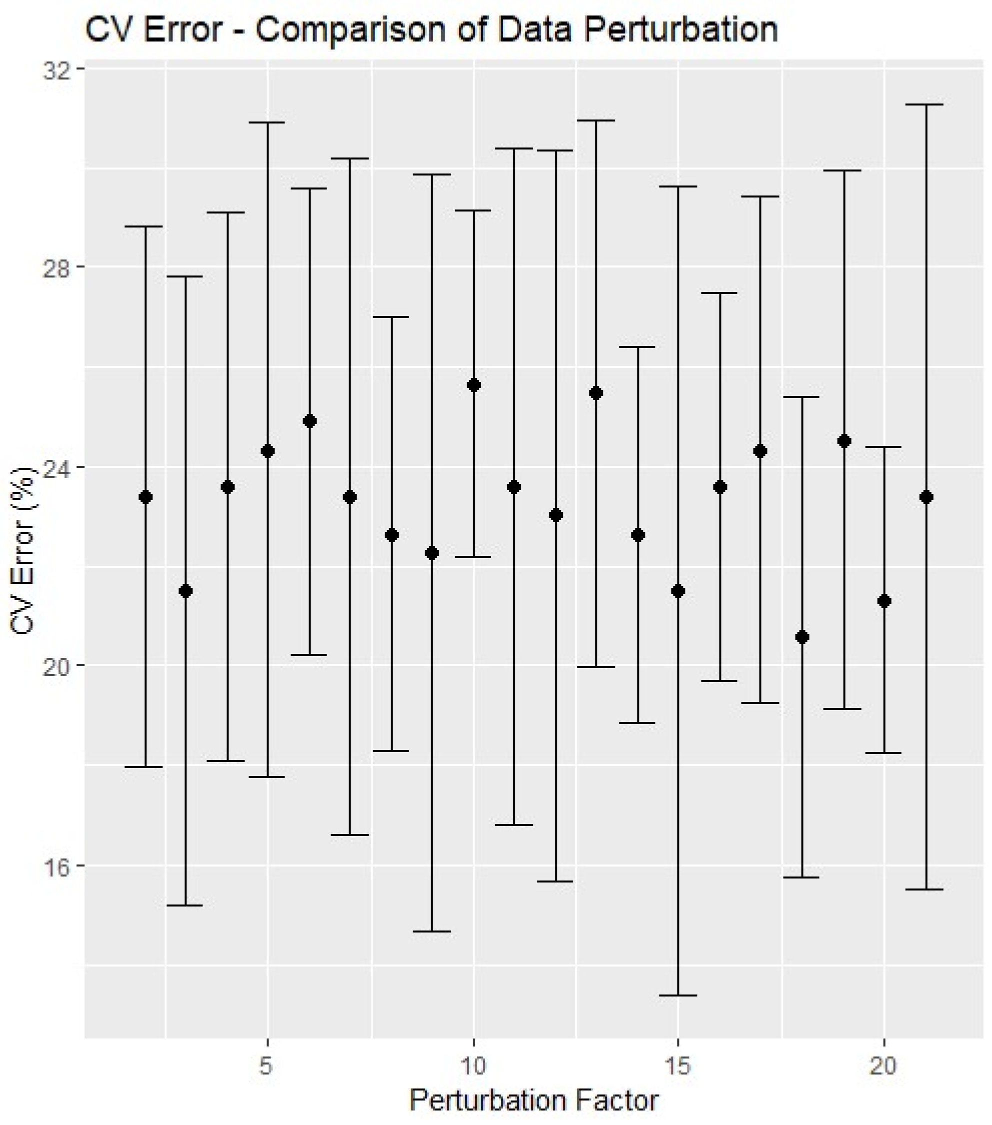
Neural networks were then fitted for varying the threshold parameter, with the number of hidden nodes (single layer) fixed at n = 3. 10-fold cross validation was performed on the various models.



The threshold parameter = 0.40 was found to perform the best.

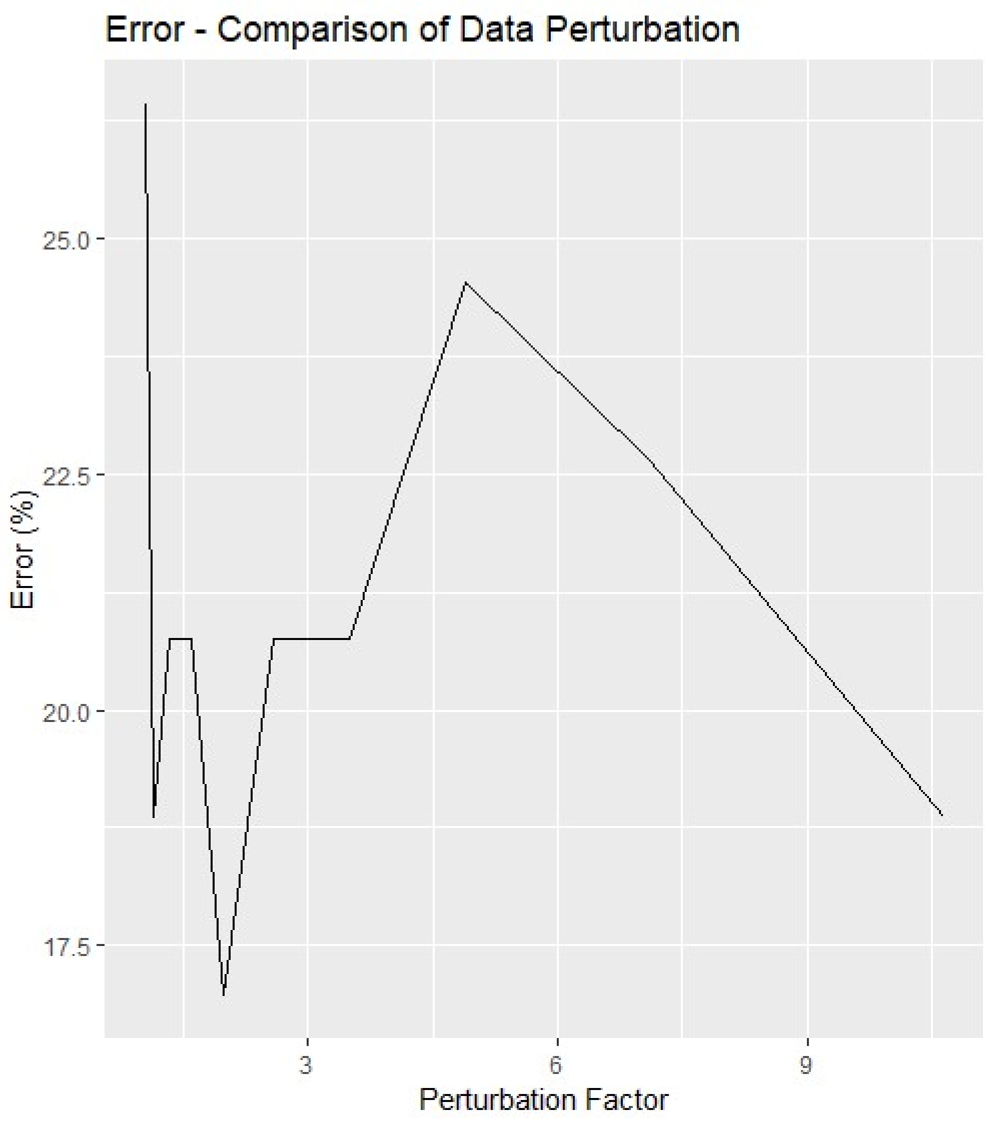
Now successive neural networks were fit using the optimized parameter found above. Each model was trained on a randomly split training dataset. For every training dataset sampled during every CV, a single observation was rescaled by a perturbation factor to simulate an outlier. The perturbation factor varied from 1 (no change) to 5. It was repeated for ranges from 1 to 21.


CV Error - Comparison of Data Perturbation
CV Error (%)
30
25
20
15
1
2
3
4
5
Perturbation Factor



The data above does not show any noticeable variation. This could be due to compatibility issues with this dataset and neural networks in general. Alternatively, this may be affected by limited training time afforded for each model.

The test was again repeated, however, this time no resampling would occur. The same training observations was rescaled each time.

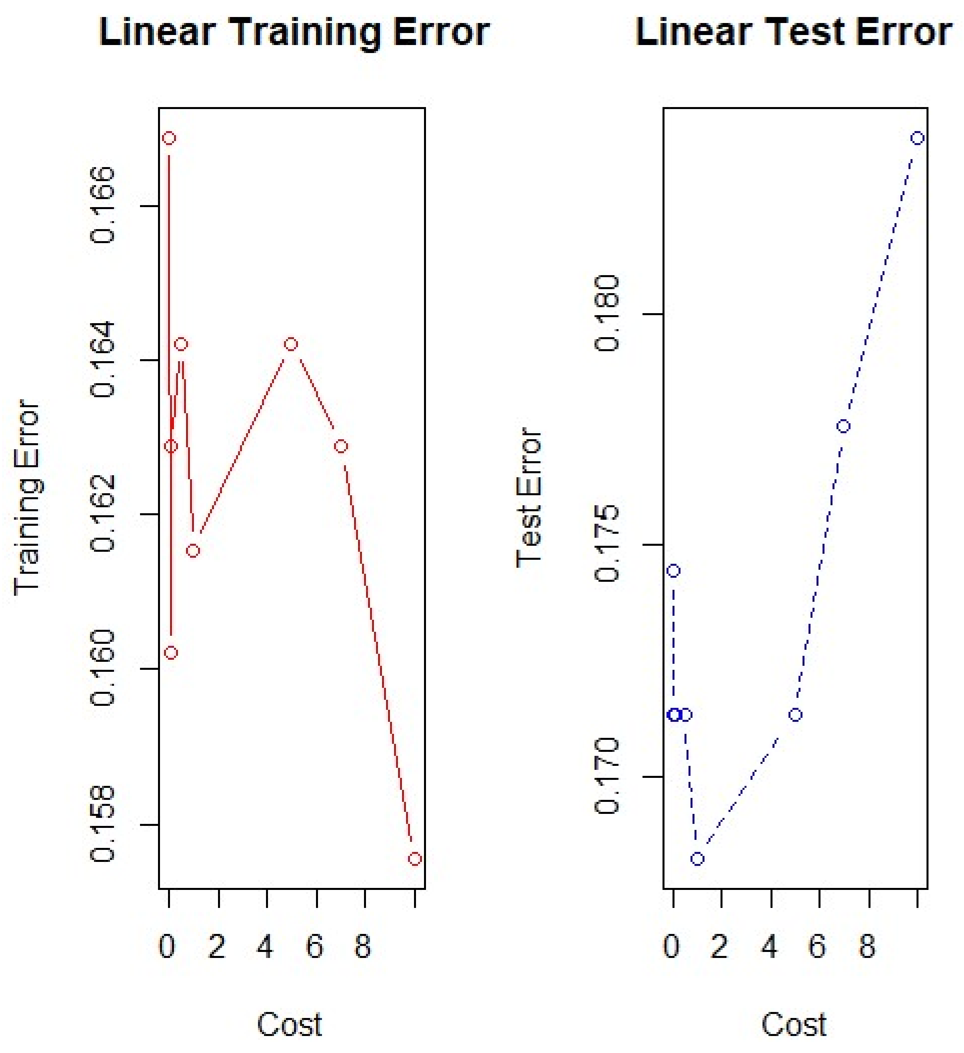


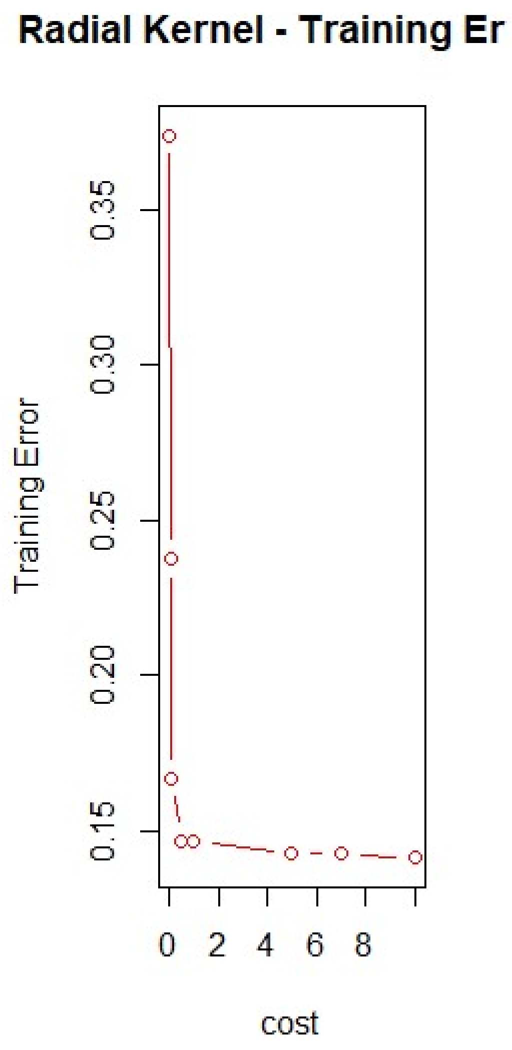
The plot shows that the data perturbation does in fact affect the neural network model in training. It does not, however affect the model in a <u>predictable way</u>. Once again it would be unwise to draw conclusions that are not completely evident. I would like to repeat this test in the future for various datasets in a more controlled and methodical manner.

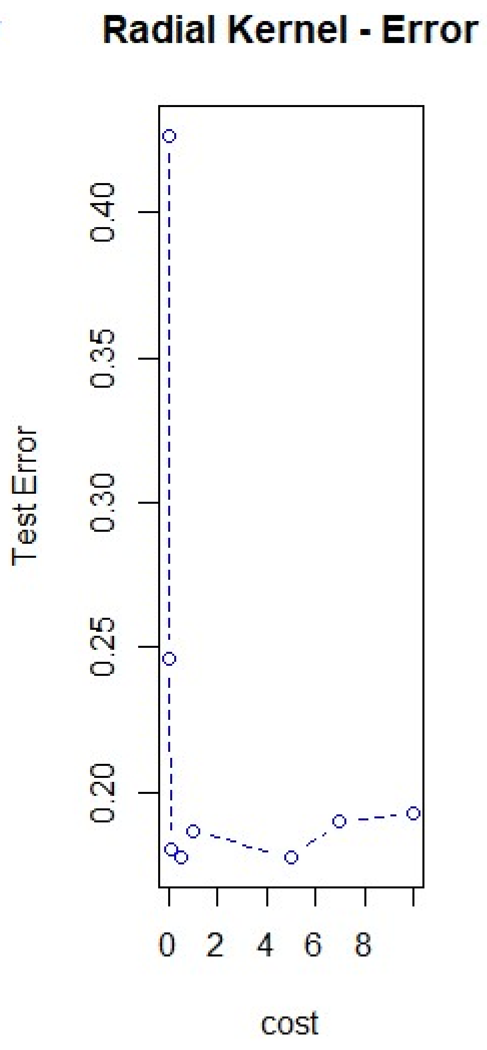
EXERCISE 4

This problem involves the OJ data set in the ISLR package. We are interested in the prediction of "Purchase". Divide the data into test and training.

(A) Fit a support vector classifier with varying cost parameters over the range [0.01, 10]. Plot the training and test error across this spectrum of cost parameters, and determine the optimal cost.
(B) Repeat the exercise in (A) for a support vector machine with a radial kernel. (Use the default parameter for gamma).
(C) Repeat the exercise again for a support vector machine with a polynomial kernel of degree=2. Reflect on the performance of the SVM with different kernels, and the support vector classifier, i.e., SVM with a linear kernel.




Radial Kernel - Training Er
Training Error
0.15
0.20
0.25
0.30
0.35
cost
0
2
4
6
8


Radial Kernel - Error
Test Error
0.20
0.25
0.30
0.35
0.40
cost
0
2
4
6
8

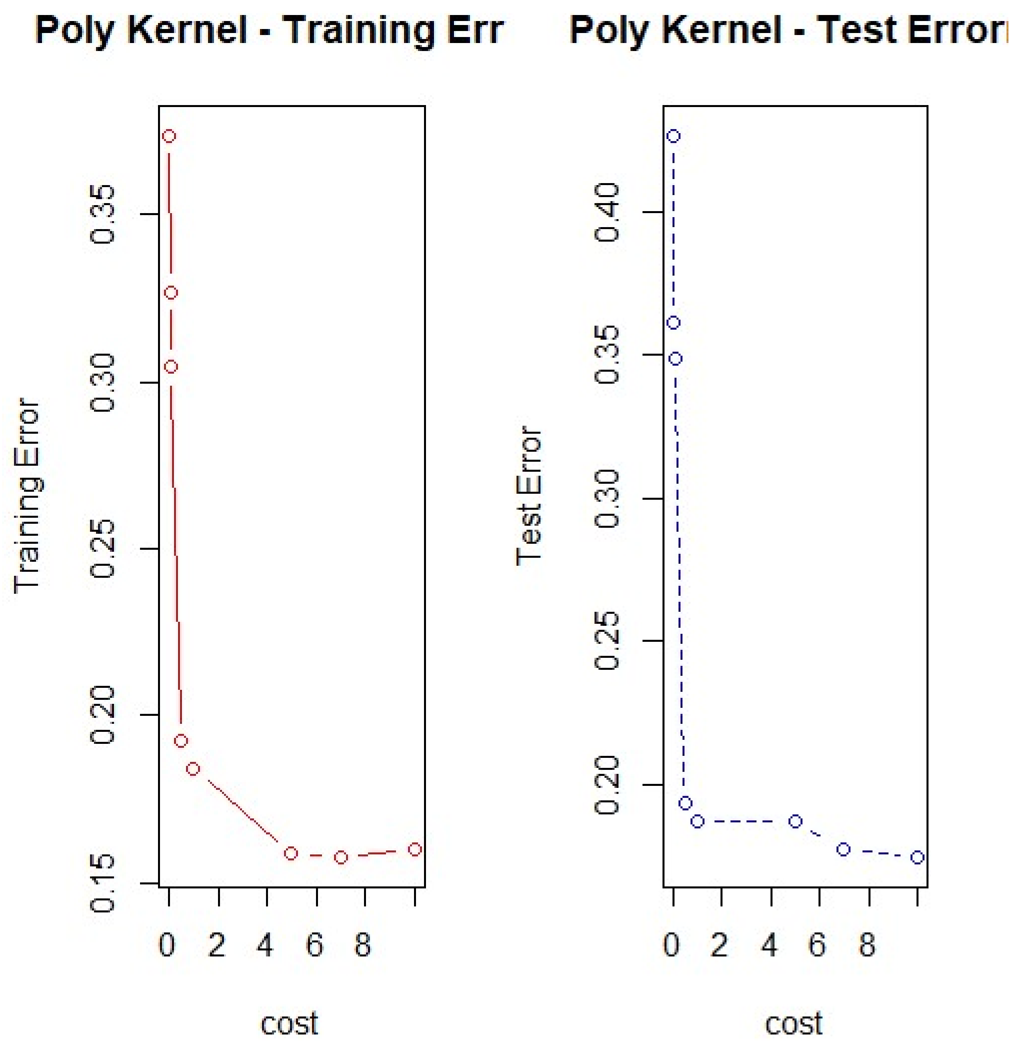

Poly Kernel - Training Err
Training Error
0.15
0.20
0.25
0.30
0.35
0
2
4
6
8
cost
Poly Kernel - Test Error
Test Error
0.20
0.25
0.30
0.35
0.40
0
2
4
6
8
cost

Table – Summary of Tests

Kernel Type	Optimal Cost	Training Error	Test Error
Linear	0.05	16.02 %	17.13 %
Radial	5	14.29 %	17.76 %
Polynomial	7	15.75 %	17.76 %

At optimal costs the linear, radial and polynomial all compare favourably in terms of test error. The polynomial and radial kernels show a general reduction in error as the complexity parameter is reduced. This reduction starts of large and falls away. This general trend is apparent in both the training and test error plots. The linear kernel, however, shows quite a different trend. For the training set the error rate appears to oscillate wildly, before diving off at the highest cost parameter. The test set error rate has a different trend. It appears to follow a u-shape pattern, with the lowest error rate occurring at cost = 1. This behavior might be explained by the fact that a linear kernel is not an inductive bias that holds with this dataset. The decision boundary may be more reliably explained by a non-linear function.

EXERCISE 5

Access the SwissBankNotes data (posted with assignment). The data consists of six variables measured on 200 old Swiss 1,000-franc bank notes. The first 100 are genuine and the second 100 are counterfeit. The six variables are length of the bank note, height of the bank note, measured on the left, height of the bank note measured on the right, distance of the inner frame to the lower border, distance of inner frame to upper border, and length of the diagonal.

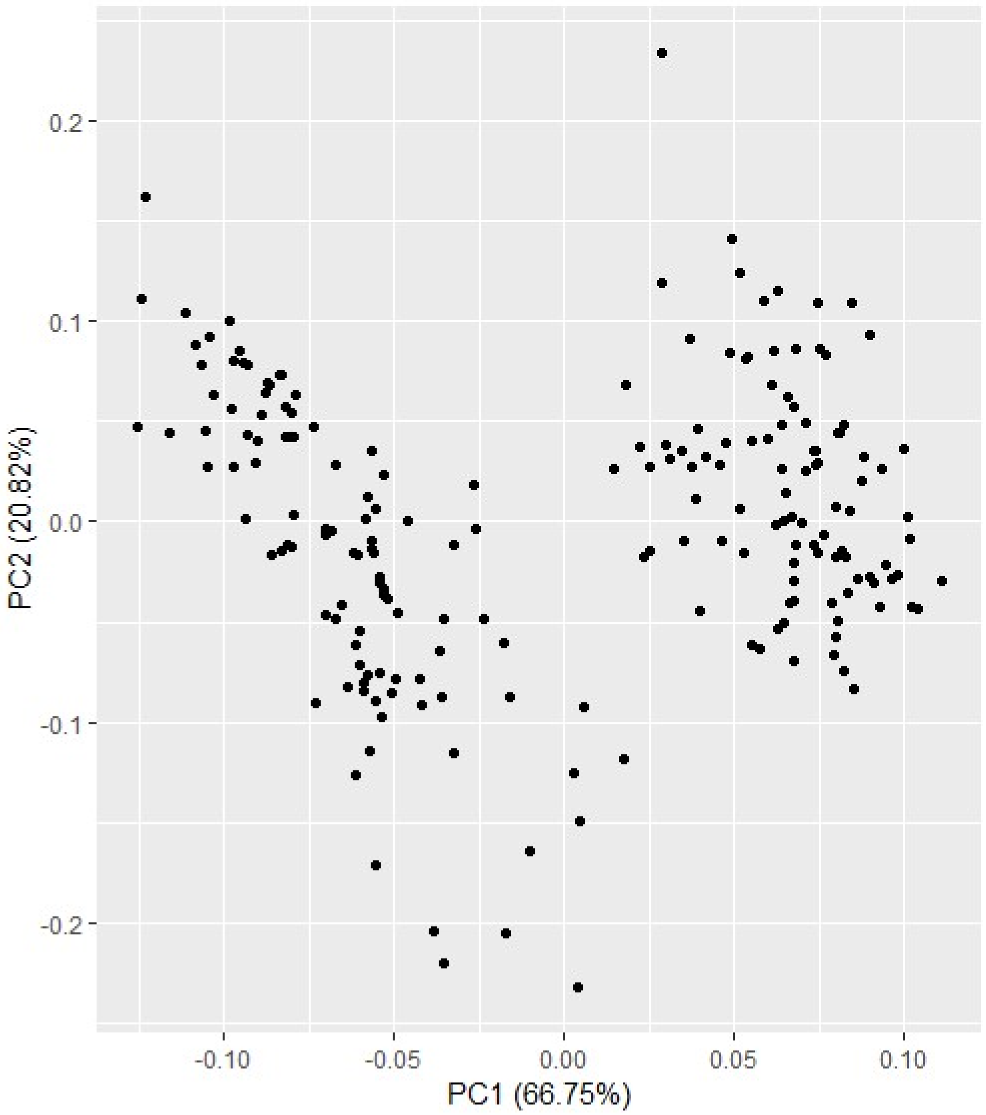
Carry out a PCA of the 100 genuine bank notes, of the 100 counterfeit bank notes, and all of the 200 bank notes combined.

Do you notice any differences in the results? Show all work in the selection of Principal Components, including diagnostic plots.

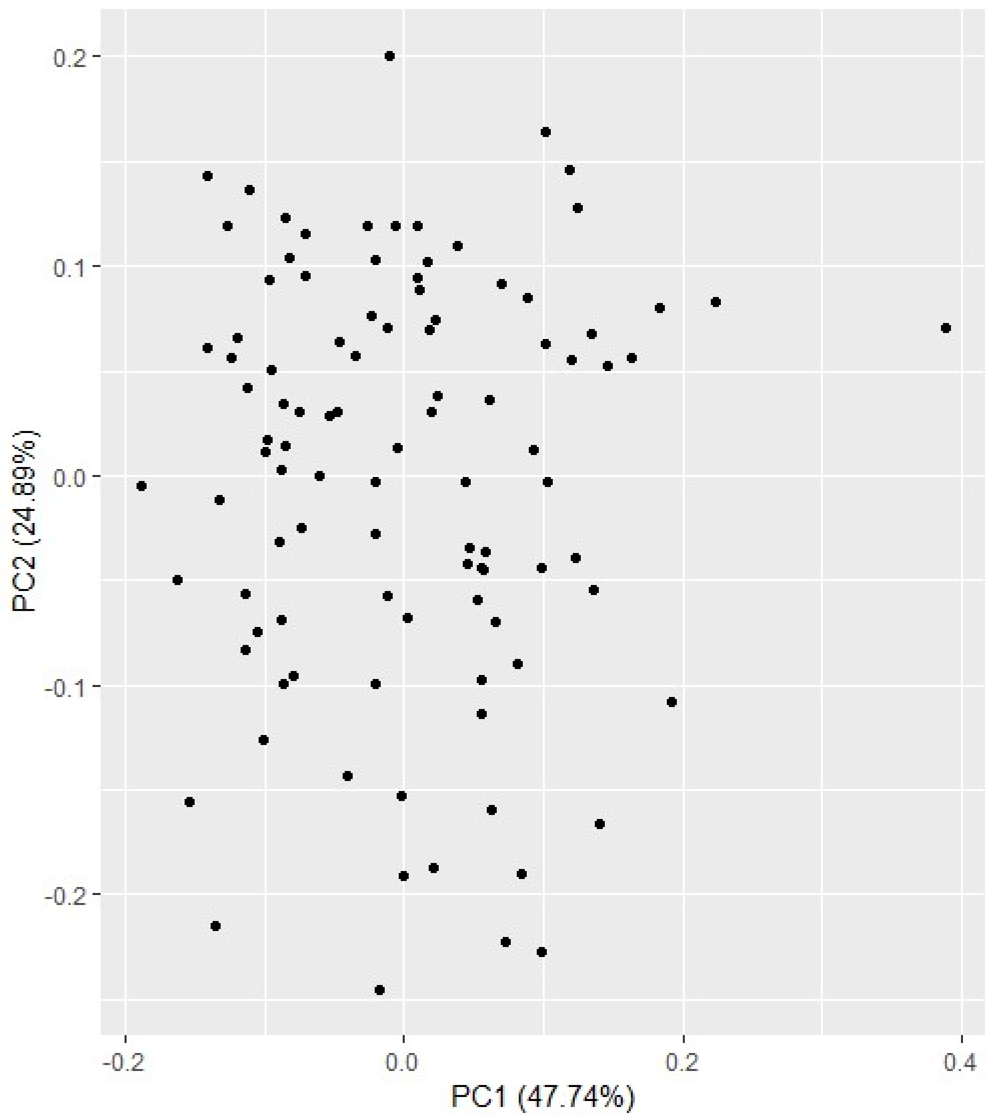
As we can see below there is nice separation between the classes. This particular plot captures the direction of the components data. One cluster represents the original notes data, and the other represents the duplicate notes data. The separation means that the transformed data would be classified with relative ease.

Examining the PCA summary the first component captures around the 66.7% variance and second component is capturing around 20% variance. The first 2 principle components combined capture almost 90% of the variance of the data.

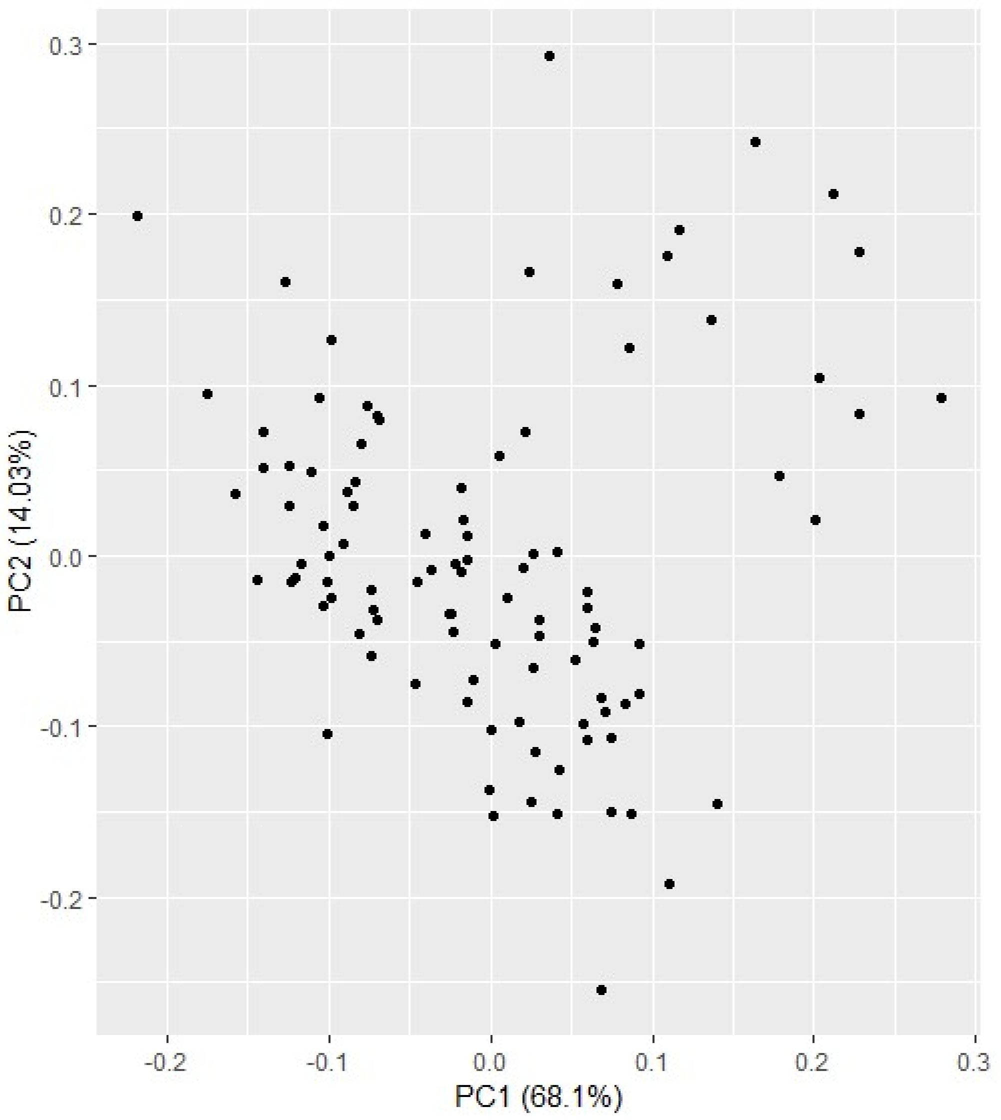
Plot: PCA Full Data



Plot: PCA for Genuine Notes



Plot: PCA Counterfeit


PC2 (14.03%)
PC1 (68.1%)