Abdoulaye Niamou
Data mining Homework 5

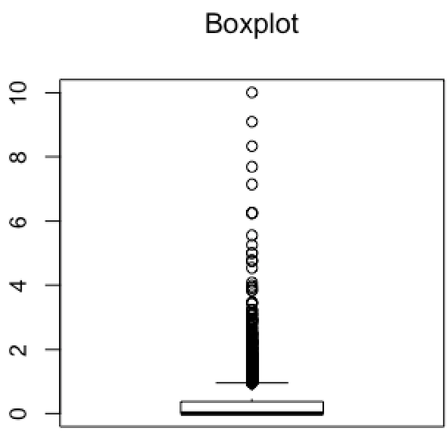
```
> results<-cbind(m,error_train,error_test)
> results
       m   error_train    error_test
 [1,]  2 0.07934782609 0.08695652174
 [2,]  4 0.07554347826 0.08369565217
 [3,]  6 0.06983695652 0.08478260870
 [4,]  8 0.06929347826 0.08043478261
 [5,] 10 0.06875000000 0.08260869565
 [6,] 12 0.06956521739 0.08478260870
 [7,] 14 0.06956521739 0.08695652174
 [8,] 16 0.06711956522 0.08152173913
 [9,] 18 0.06684782609 0.07826086957
[10,] 20 0.06548913043 0.08152173913
> min(error_train)
[1] 0.06548913043
> min(error_test)
[1] 0.07826086957
```

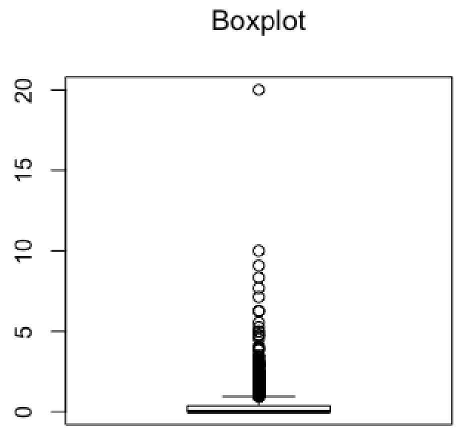
When m = 20, the model has the minimum training error of 0.065 and when m = 18, the model has the minimum test errors of 0.078.

```
> NN_error                 > additive_error
[1] 0.06956521739          [1] 0.9282608696
```

The final test errors for the neural network is 0.0696, and the final test errors for the additive model is 0.928. The neural networks and the hold out give lower error and better results than the additive model.

2.






```
> origin_error_train
 [1] 0.08152173913 0.11548913043 0.10625000000 0.07119565217 0.07717391304
 [6] 0.06983695652 0.06956521739 0.07173913043 0.06983695652 0.06657608696
[11] 0.06956521739 0.06657608696 0.06793478261 0.07038043478 0.06548913043
[16] 0.07010869565 0.06820652174 0.06657608696 0.06820652174 0.06576086957
> origin_error_test
 [1] 0.09021739130 0.12500000000 0.11739130435 0.08478260870 0.09021739130
 [6] 0.08478260870 0.08260869565 0.08369565217 0.08260869565 0.07826086957
[11] 0.08043478261 0.08152173913 0.08152173913 0.08260869565 0.07826086957
[16] 0.08478260870 0.08369565217 0.08152173913 0.08043478261 0.07826086957
> outlier_error_train
 [1] 0.08722826087 0.13641304348 0.11739130435 0.08940217391 0.08722826087
 [6] 0.07798913043 0.06684782609 0.07146739130 0.07173913043 0.07173913043
[11] 0.07010869565 0.06739130435 0.06684782609 0.06929347826 0.06739130435
[16] 0.06711956522 0.06603260870 0.06576086957 0.06657608696 0.06630434783
> outlier_error_test
 [1] 0.10108695652 0.15108695652 0.12608695652 0.09891304348 0.09891304348
 [6] 0.08695652174 0.07826086957 0.08478260870 0.08152173913 0.08043478261
[11] 0.08369565217 0.07826086957 0.07500000000 0.07608695652 0.08043478261
[16] 0.07826086957 0.07608695652 0.07500000000 0.07934782609 0.07608695652
```

The minimized training errors for the original training data is 0.0655 which occurs when m = 15.

The minimum testing errors for the original testing data is 0.0782 which occurs when m = 20

The minimum training errors for the training data with an outlier is 0.0658 which occurs when m = 18.

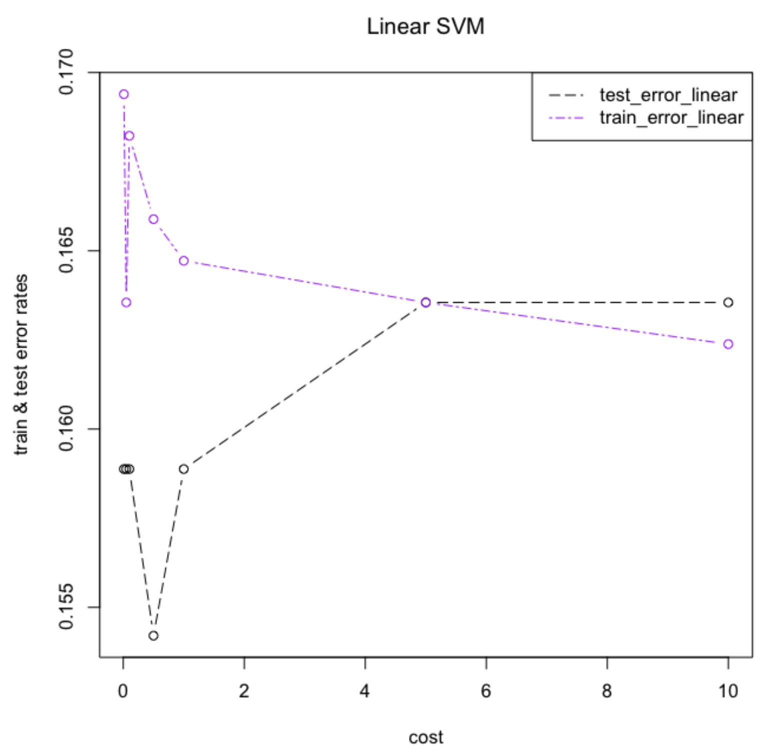
The minimum testing errors for the testing data with an outlier is 0.075 which occurs when m = 13 or 18.

We can see that the outlier point barely have any effects when fitting the data to the classifier model or when classifying the test data.

```
> error_test
 [1] 0.08043478261 0.08043478261 0.07717391304 0.07934782609 0.08043478261
 [6] 0.07934782609 0.07717391304 0.07826086957 0.07934782609 0.07934782609
[11] 0.07717391304
```

By shrinking the value of the outlier from 20 to 0, we can see that when the difference is less and equal to 6, they would have the similar (within a reasonable range of 0.001) classification errors rates on the test dataset.

4.

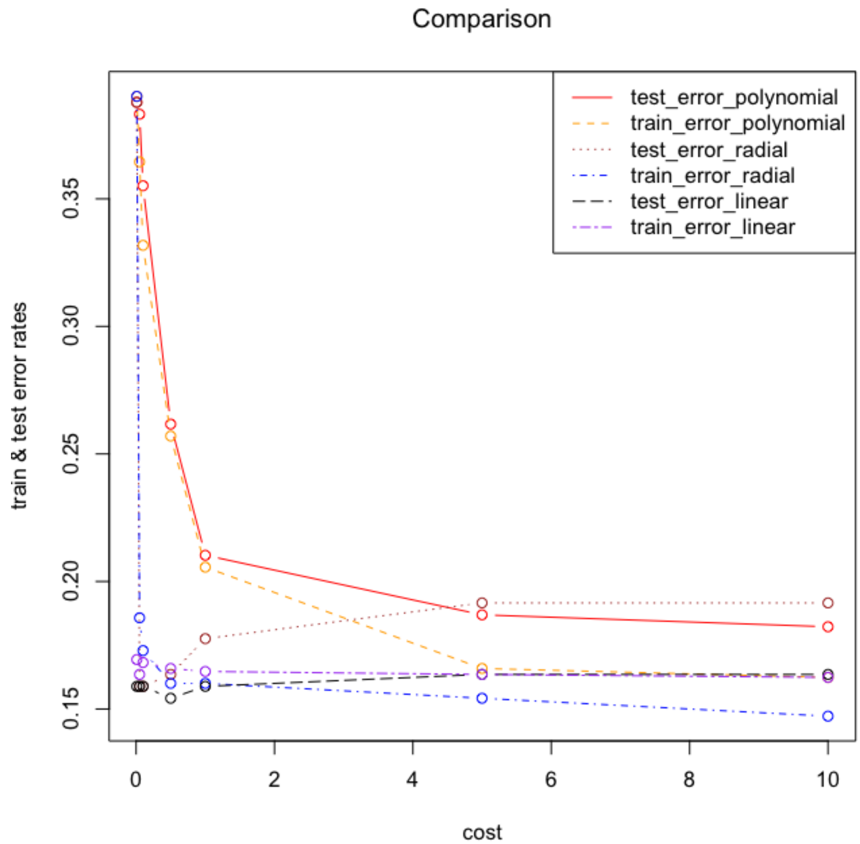


```
> linear<-cbind(cost,OJ_train_errors,OJ_test_errors)
> linear
      cost OJ_train_errors OJ_test_errors
[1,]  0.01     0.1693925234   0.1588785047
[2,]  0.05     0.1635514019   0.1588785047
[3,]  0.10     0.1682242991   0.1588785047
[4,]  0.50     0.1658878505   0.1542056075
[5,]  1.00     0.1647196262   0.1588785047
[6,]  5.00     0.1635514019   0.1635514019
[7,] 10.00     0.1623831776   0.1635514019
```

Using SVM with linear kernel, we can see that the optimal cost for training error is 10, and the optimal cost for testing error is 0.5.

When cost = 5, the training error and the test error are the same. This can probability be selected as the optimal value for cost parameter.

```
> radial<-cbind(cost,OJ_train_errors1,OJ_test_errors1)
> radial
      cost OJ_train_errors1 OJ_test_errors1
[1,]  0.01      0.3901869159    0.3878504673
[2,]  0.05      0.1857476636    0.1588785047
[3,]  0.10      0.1728971963    0.1588785047
[4,]  0.50      0.1600467290    0.1635514019
[5,]  1.00      0.1600467290    0.1775700935
[6,]  5.00      0.1542056075    0.1915887850
[7,] 10.00      0.1471962617    0.1915887850
> polynomial<-cbind(cost,OJ_train_errors2,OJ_test_errors2)
> polynomial
      cost OJ_train_errors2 OJ_test_errors2
[1,]  0.01      0.3901869159    0.3878504673
[2,]  0.05      0.3644859813    0.3831775701
[3,]  0.10      0.3317757009    0.3551401869
[4,]  0.50      0.2570093458    0.2616822430
[5,]  1.00      0.2056074766    0.2102803738
[6,]  5.00      0.1658878505    0.1869158879
[7,] 10.00      0.1623831776    0.1822429907
```



From the above graph, we can conclude that SVM with linear kernel performs better than SVM with polynomial kernel classifier for classifying this dataset.