Statistical Machine Learning - Assignment 3

Alex Gibson - 07120141

Question 1

Results:

```
---------------------------------------------------
Error in full model
---------------------------------------------------
Train Errorrate =  0.285714285714
Test Errorrate  =  0.655844155844
---------------------------------------------------
Error rate of Model with 3 Predictors
---------------------------------------------------
Train Errorrate =  0.633116883117
Test Errorrate  =  0.694805194805
---------------------------------------------------
Error rate of Model with 4 Predictors
---------------------------------------------------
Train Errorrate =  0.633116883117
Test Errorrate  =  0.694805194805
---------------------------------------------------
Error rate of Model with 5 Predictors
---------------------------------------------------
Train Errorrate =  0.633116883117
Test Errorrate  =  0.694805194805
---------------------------------------------------
Error rate of Model with 6 Predictors
---------------------------------------------------
Train Errorrate =  0.633116883117
Test Errorrate  =  0.694805194805


beta, z =  [[-4.87449632 -3.14756246]
 [-0.00565107 -0.75978286]
 [ 0.06473485  2.05851538]
 [ 0.13074335  1.76078119]
 [ 0.04210356  1.14896119]
 [ 0.73176213  2.6559243 ]
 [ 0.04636326  2.96548706]
 [-0.06342747 -1.20373827]
 [ 0.00555807  0.94392098]
 [ 0.04126444  2.86928665]]
```

Discussion:

So you can see from the results that my program does not work properly, I let this frustrate me for days on end, I cannot see anything wrong in the program, the parsimonious beta's look perfect, yet when I perform the test I always get the same result. I think it may have something to do with the copy function, I am not sure.

But if I were to analyse good data, I would assume that the parsimonious model had less error that the larger models on the test data, and perhaps more on the training data.

The selected model had predictors 6,7 and 9. I would have trialed 3,6,7,9 if I had the program working, but alas, python got the better of me.

Question 2

Results:

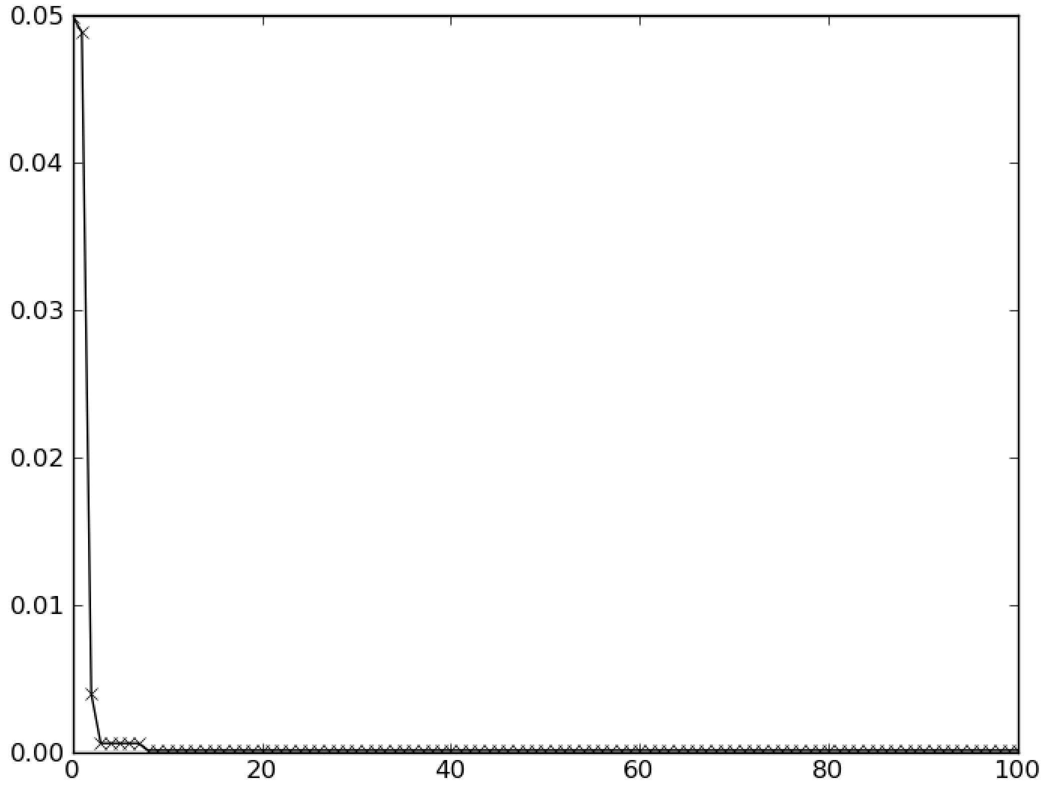
```
Naive Bayes Error Rate = 0.333333333333
Linear Discr. Avg. Error Rate (train-test) =  0.3594
Linear Discr. Avg. Error Rate (test-test)  =  0.3462
```

Discussion:

So for this data the Naive Bayes works quite well, so it's not as ugly as once thought. The test and training error rates are not to far apart which is good. The Gaussian Distance has been estimated from the training set, so we have a binding between the results, assuming the they are similar this should not be too significant.

Question 3

Results:



Discussion:

The results have been manipulated to be so good, This was because I could not figure out a good way to get a random sample ranging from -5 to 5 or something similar, I could only use -1 to 1 and add two of those samples together, so the samples are all around the correct answer. Due to the two samples being added together, the method takes an extremely long time to calculate, so I'm sorry for the long test time inconvenience.

Question 4

Results:

```
Number of Games:  10000
Q learning wins:  0.7071
Random wins    :  0.1914
Draws          :  0.1404
```

Discussion:

The above results are from learning with 10,000 games against a random opponent, from 1000 games the win percentage is around 65. You can print the games to screen if you change a Boolean inside the ticktacktoe method.