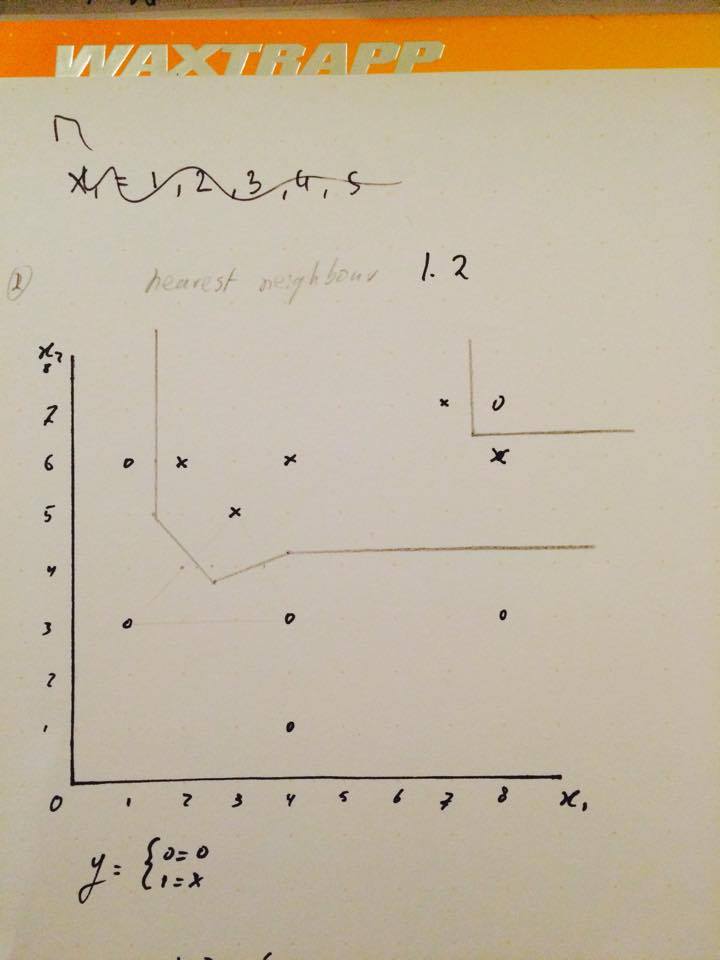
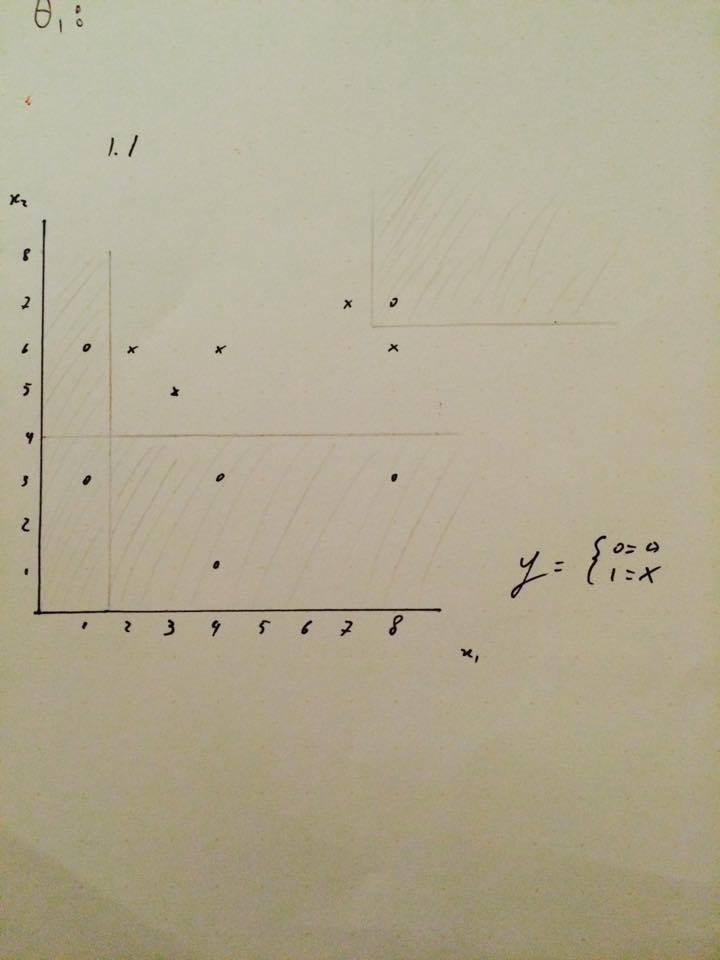
Written assignment 4

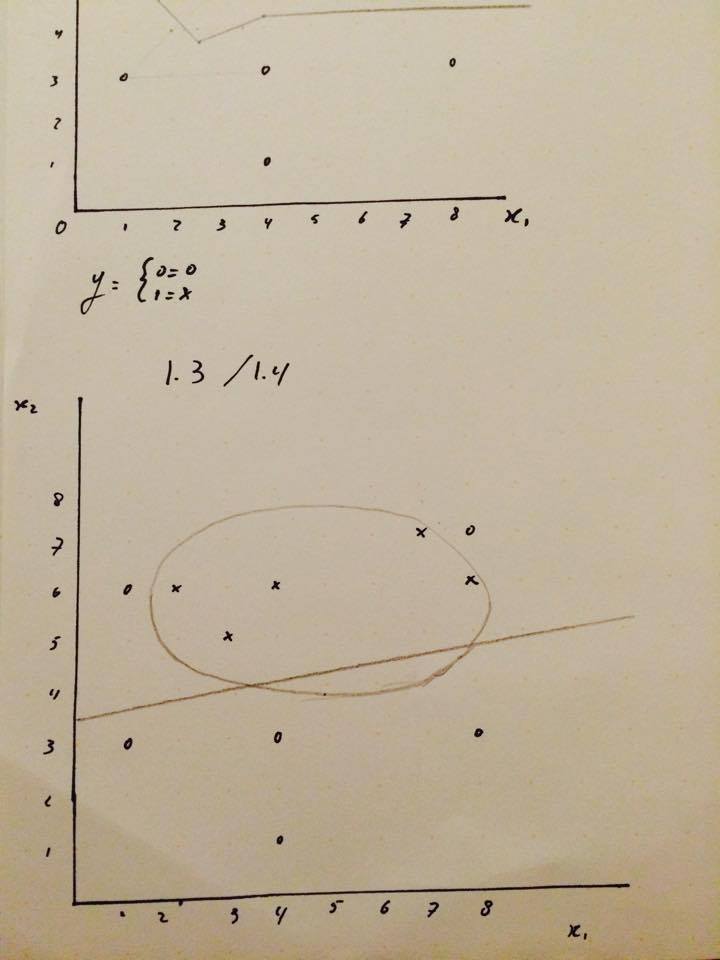
x1 = {1,1,2,3,4,4,4,7,8,8,8}

x2 = {3,6,6,5,1,3,6,7,6,7,3}

x3 = {0,0,1,1,0,0,1,1,1,0,0}

q.1 a)





q.1 b)

The different classifiers are better suited in different situations. Intuitively it seems to be the case that logistic regression using quadrating terms does well in clustering more dense clusters whereas 1-nearest neighbour is very well at making precise bounderies, thus sometimes losing track of the more general form.

I’m not sure how it could be implemented, if it were possible to combine for instance 1-nearest neighbour and logistic regression with quadratic terms, in the way that you use logistic regression to find the general cluster and than find a better fit using the ellipse or circle as a basis to which nearest neighbour is applied.q.2)

k = 3, points of initiation given as {1, 3, 8}

This divides the dataset as follows:

{1, 2}/ {3, 3, 4, 5, 5}/ {7, 10, 11, 13, 14, 15, 17, 20, 21}, note that the 2 is randomly chosen to go with the first cluster, it could also have gone to the second cluster since it is as close to 1 as to 3.

Doing one iteration of k-means, means moving the initiation point to the average of each cluster respectively. This gives:

(1+2)/2 = 1.5

(3+3+4+5+5)/5 = 4

(7+10+11+13+14+15+17+20+21)/ 9 = 14.222

Thus the clusters will be formed using the points {1.5, 4, 14.222}, this gives the following clusters:

{1, 2}/ {3, 3, 4, 5, 5, 7}/ {10, 11, 13, 14, 15, 17, 20, 21}.

The cost of k-means is calculated using mean-squared error; (1/m)\*(sum from i = 1 to m) ||x(i) – uc(i)||^2.

This gives:

1/16\*(0.5 + 4 +169.58) = 10.88