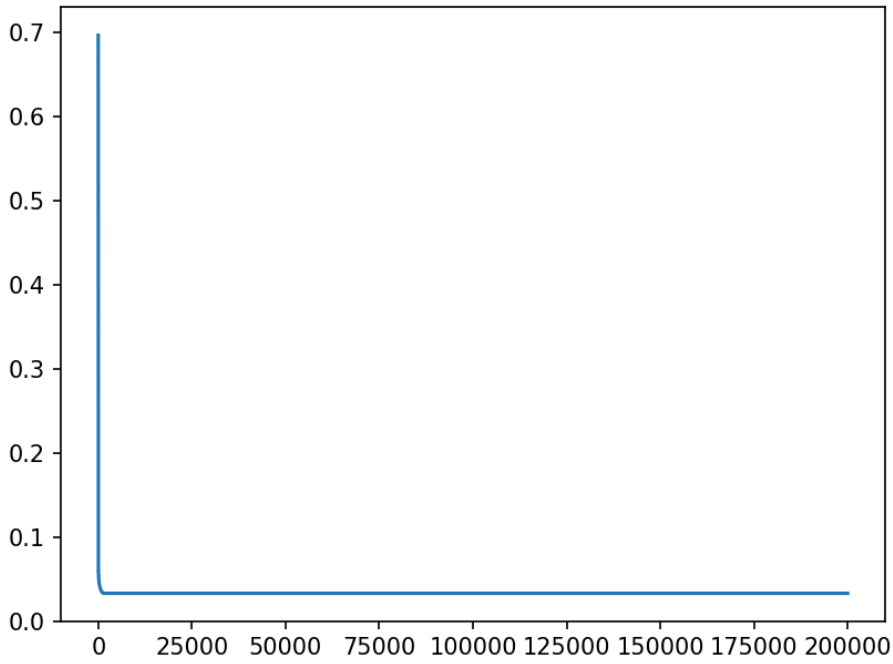
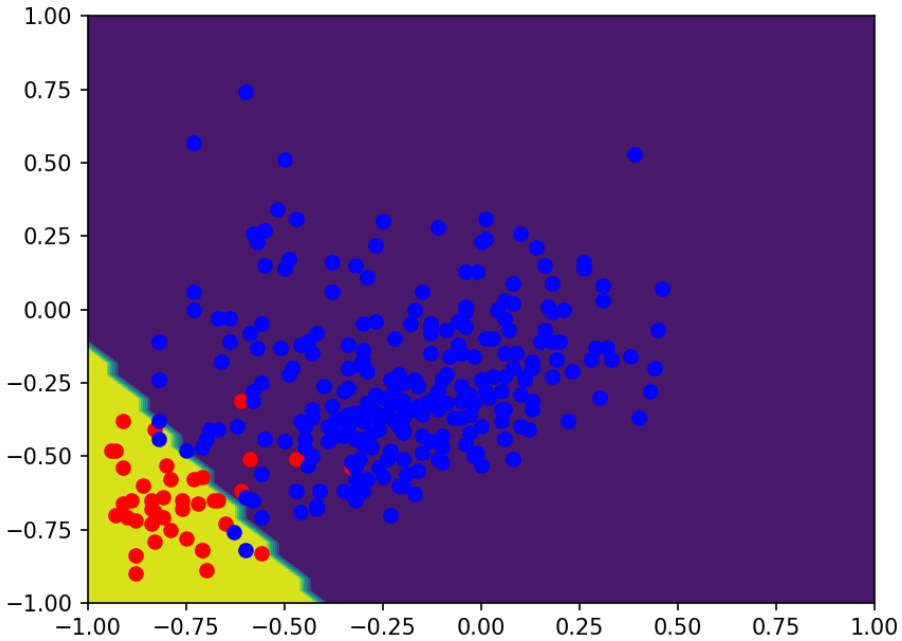
Q1
a:
Identity
[[0.35542488, 0.35542488],
[0.71084977, 0.71084977],
[0.35542488, 0.35542488]]
[2.36949922],
[1.42169953],
[1.42169953]

Tanh
[[0.25257836, 0.25257836],
[0.50515672, 0.50515672],
[0.25257836, 0.25257836]]
[[2.36636925],
[1.27085758],
[1.27085758]]

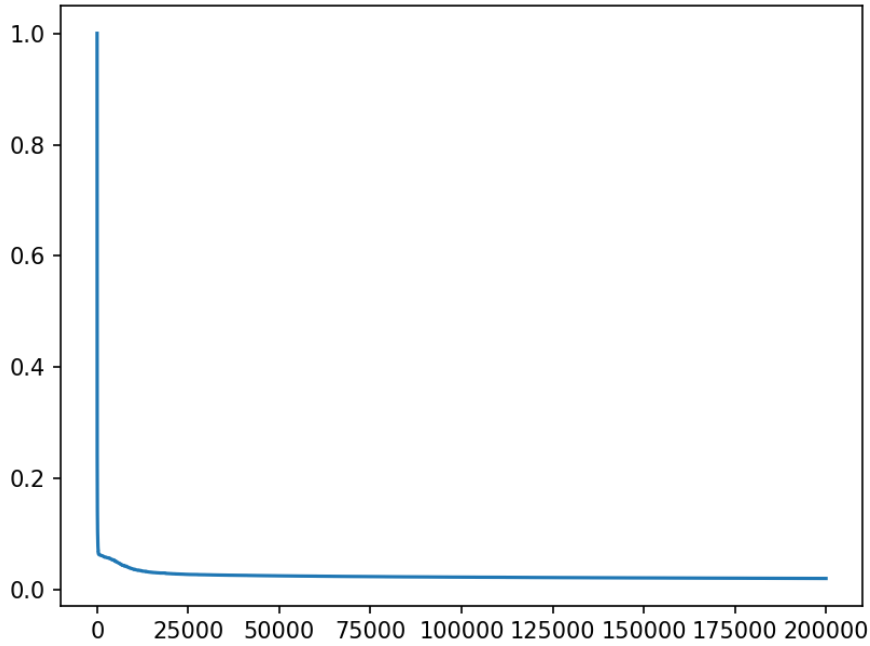
b:
Identity
[[0.35566715, 0.35566715],
[0.71133431, 0.71133431],
[0.35566715, 0.35566715]]
[[2.36953466],
[1.42266861],
[1.42266861]]
Tanh
[0.25251131, 0.25251131],
[0.50502263, 0.50502263],
[0.25251131, 0.25251131]]
[[2.36630271],
[1.27014818],
[1.27014818]]

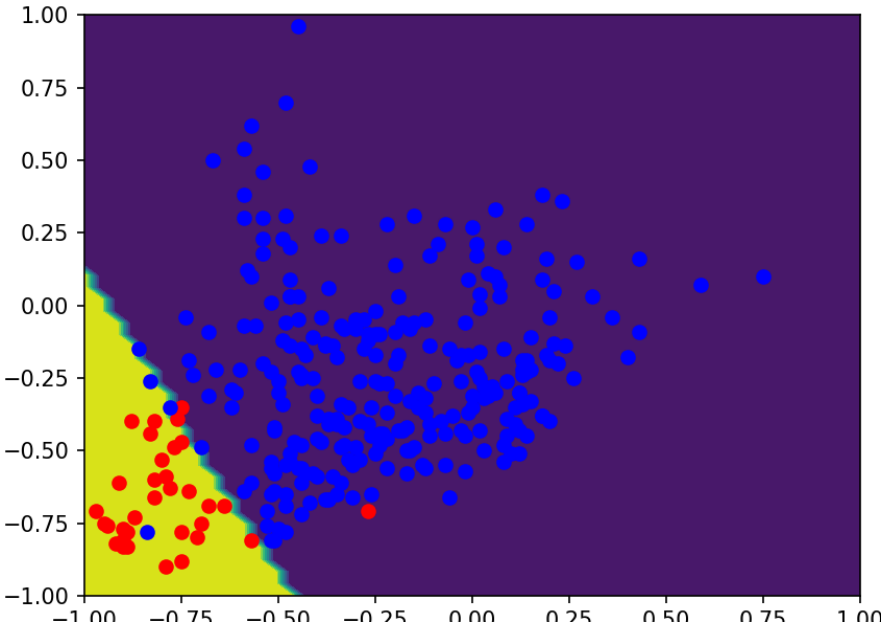
So the results only changed a little.
Q2:
a:


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
25000
50000
75000
100000
125000
150000
175000
200000


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

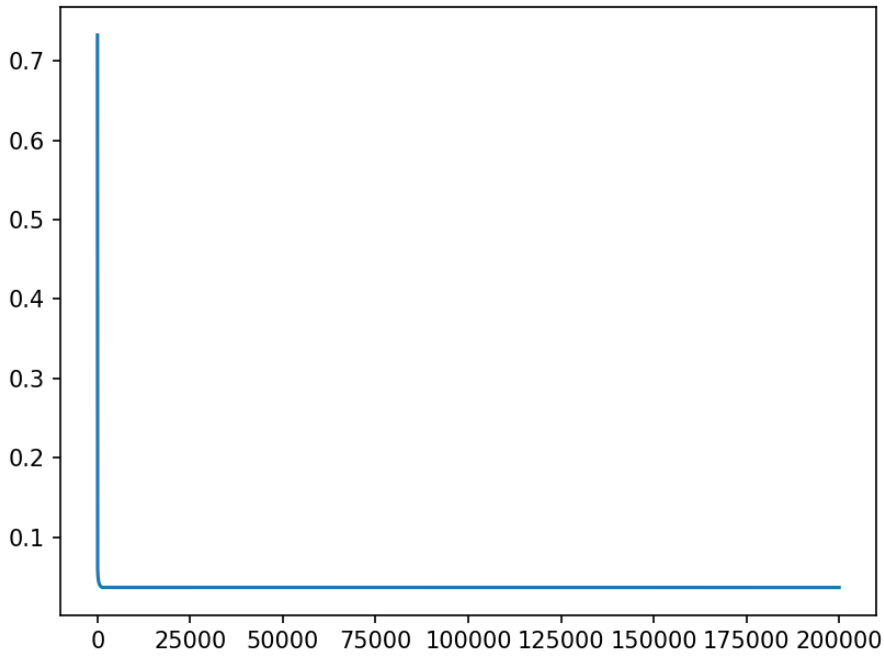
b:

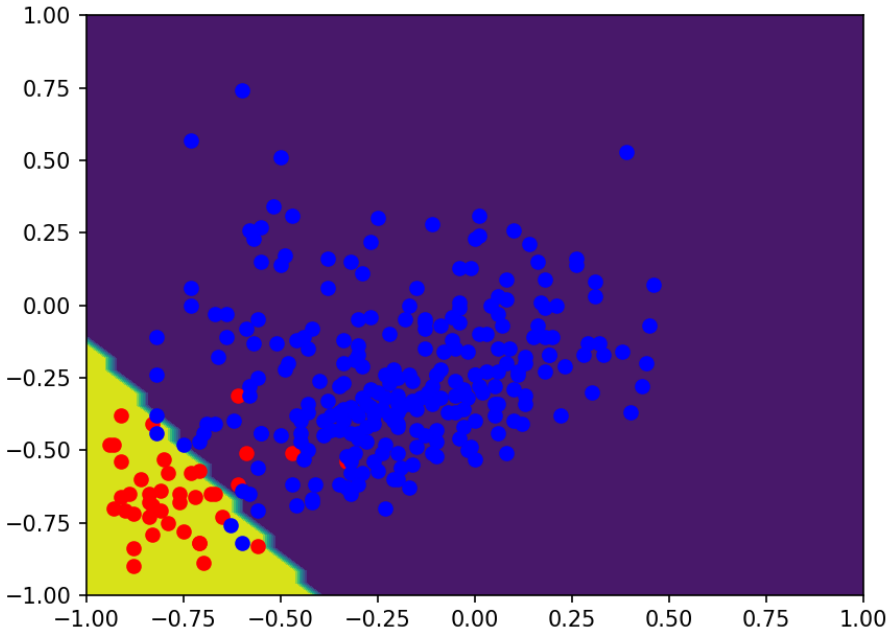




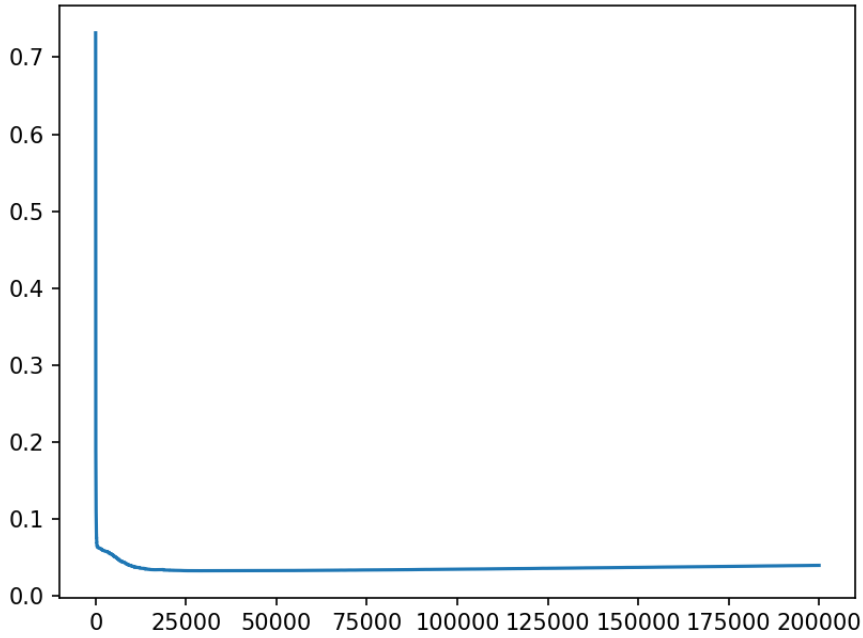
c:

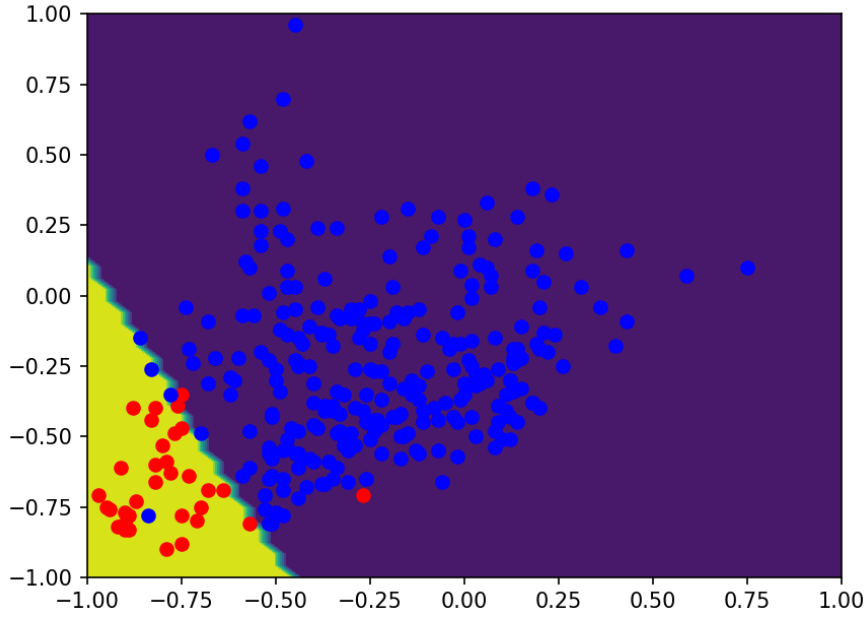
Gradient Descent


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
25000
50000
75000
100000
125000
150000
175000
200000


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

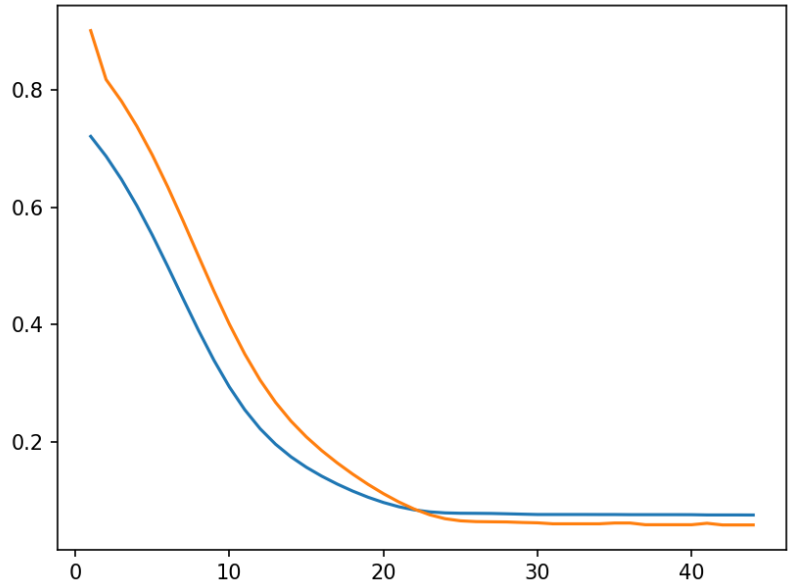
SGD:

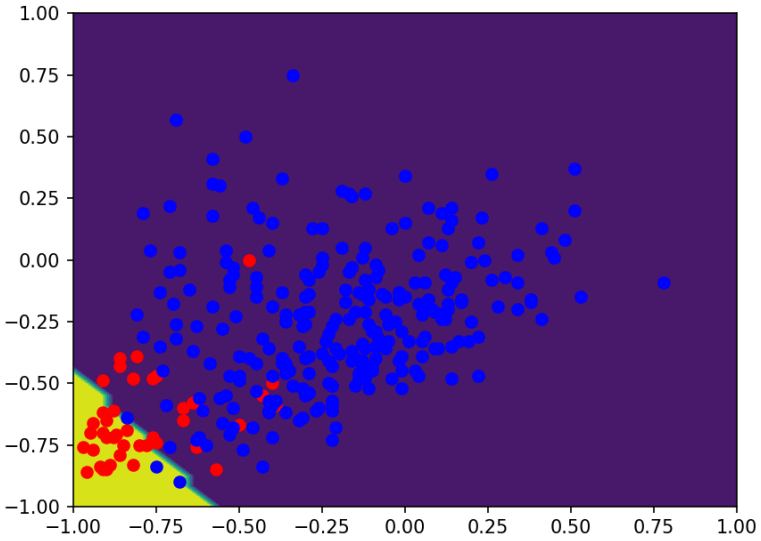




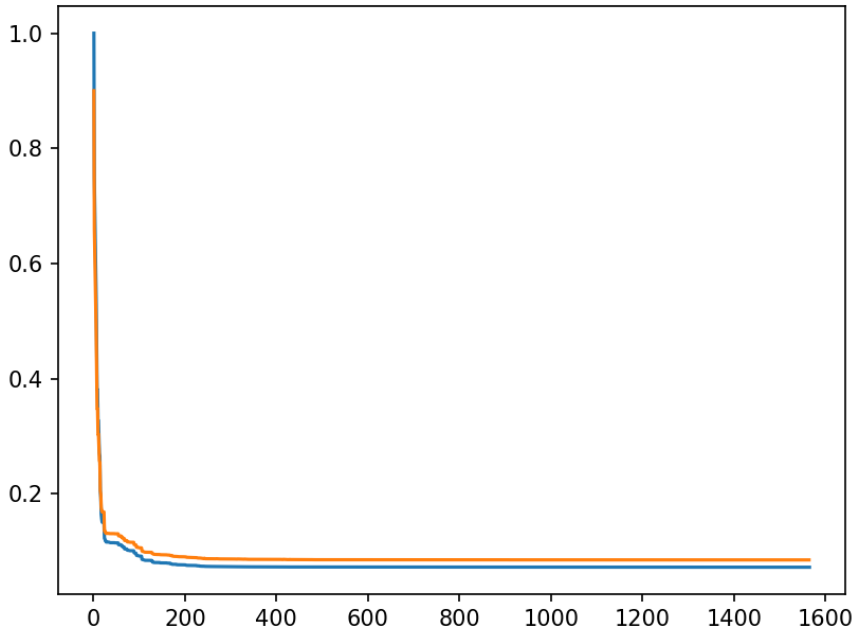
d:

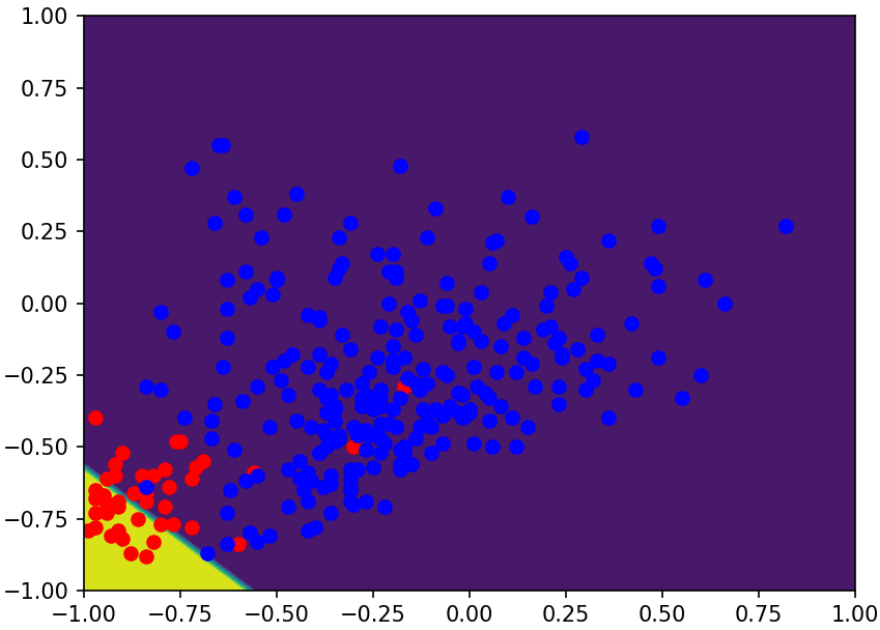
Gradient decent:


0.8
0.6
0.4
0.2
0
10
20
30
40


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

SGD:



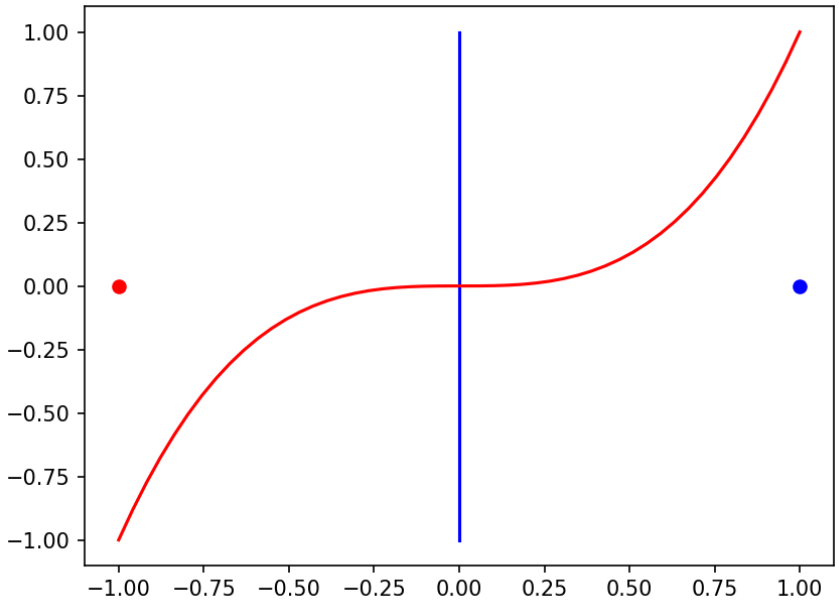


Q3:

a: Since the two points are symmetric about the y axis, the optimal separating hyperplane is just the ‘plane’ that is the perpendicular bisector of the line segment joining the two points. The optimal separating hyperplane equation is g(x) = sign(x1)

b: for the first point, z1 = 1,z2 = 0, for the second point z1 = -1, z2 = 0. Therefore the optimal hyperplane equation g(z) = sign(z1).

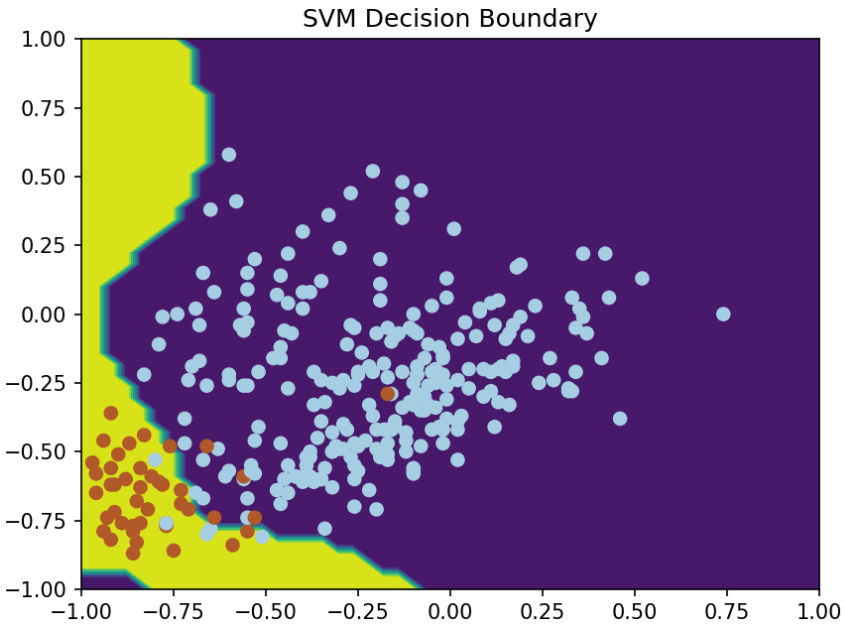
c:



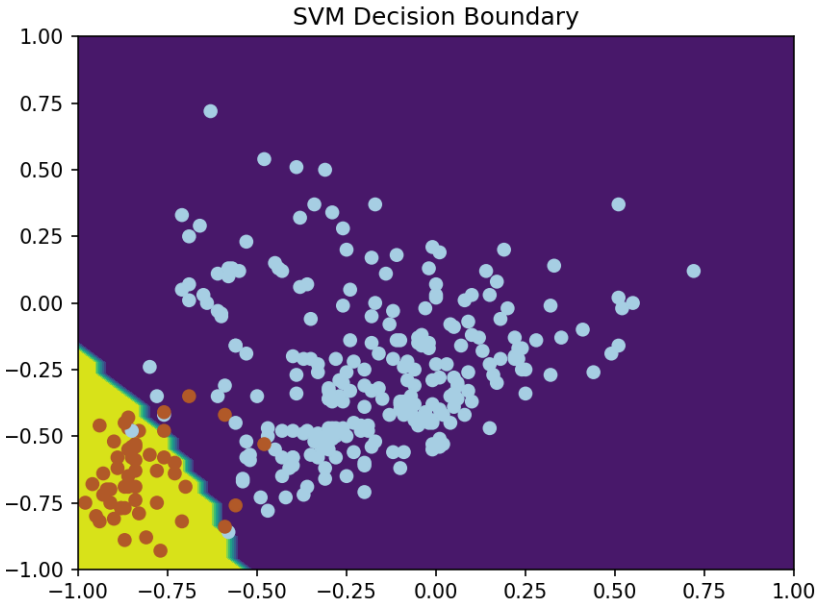
d:
z(x)1 = (x1)^3 - x2 z(x)2 = (x1)*(x2)
z(y)1 = (y1)^3 - y2 z(x)2 = (y1)*(y2)
z(x) * z(y) = ((x1)^3 - x2) * (y1)^3 - y2 z(x)2 + (x1)*(x2) * (y1)*(y2)
= (x1)^3 * (y1)^3 - x2 * (y1)^3 - z(x)2 * (x1)^3 + (y1)^3 * (x1)^3 + (x1)*(x2) * (y1)*(y2)

e:
g(x) = sign((x1)^3 - x2) for any point (x1, x2)

Q4:
a:
C = 100

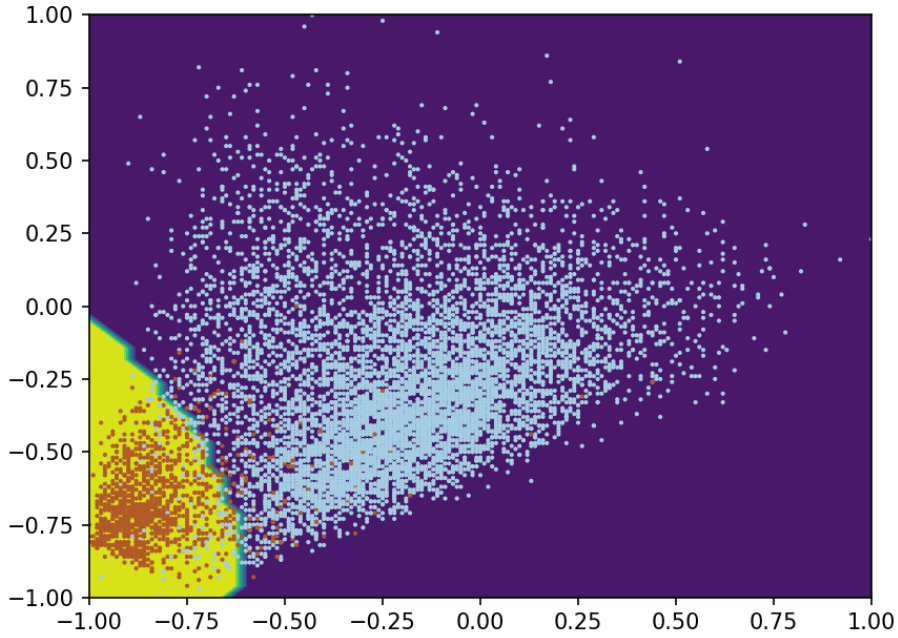


C = 0.01



b: As we can see from the graph, as C gets bigger the decision boundary is getting more complex.

c: C = 0.01



Etest = 0.030117803956434763

Q5: From homework 11, we know:
Etest for Linear model with 8 order polynomial transform is: 0.0316737052678373
Etest for KNN is: 0.030228939764392088
Etest for RBF network is: 0.02967326072460547
From this homework, we know:
Etest for SVM is: 0.030117803956434763
After check etest part on Q2, I got:
Etest for Neural network is: 0.0310086
As we can see all those models have very similar Etest, the highest Etest is only less than 0.002 larger than the lowest Etest. Therefore, I think all of those models perform pretty ok.