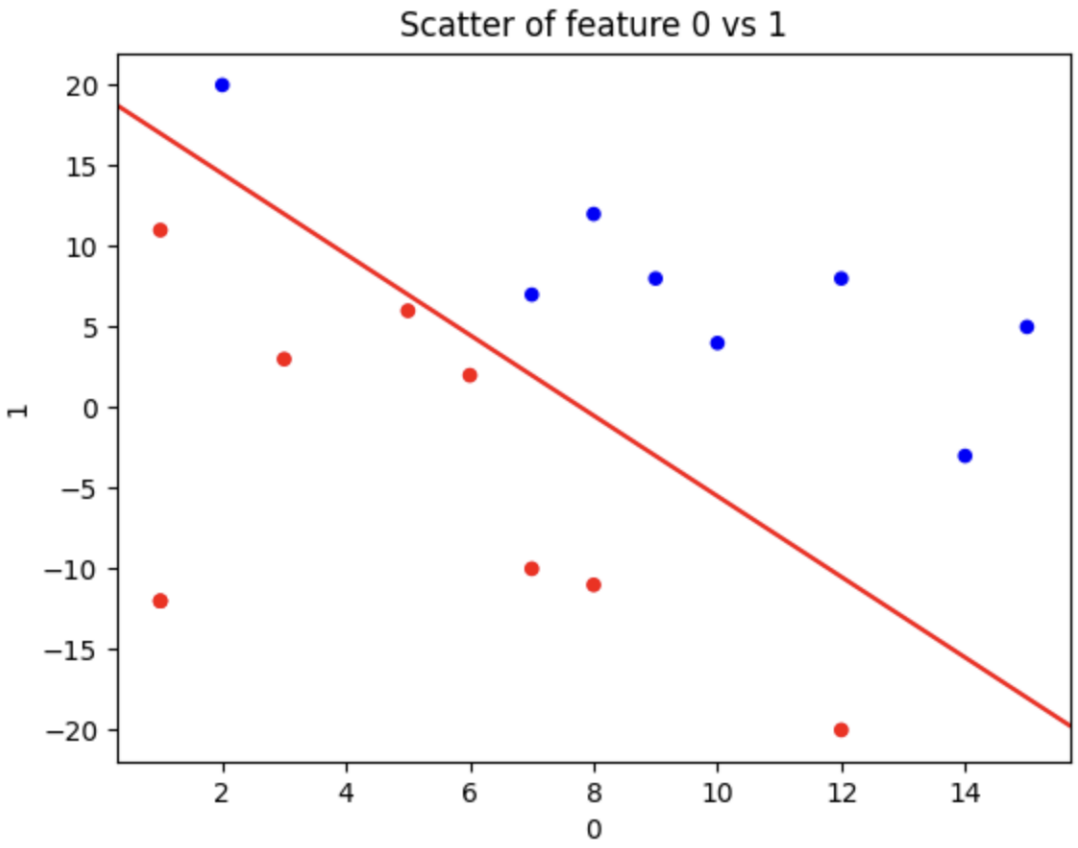
README for COMS 4701 Artificial Intelligence Homework 4 Coding

1. Perceptron

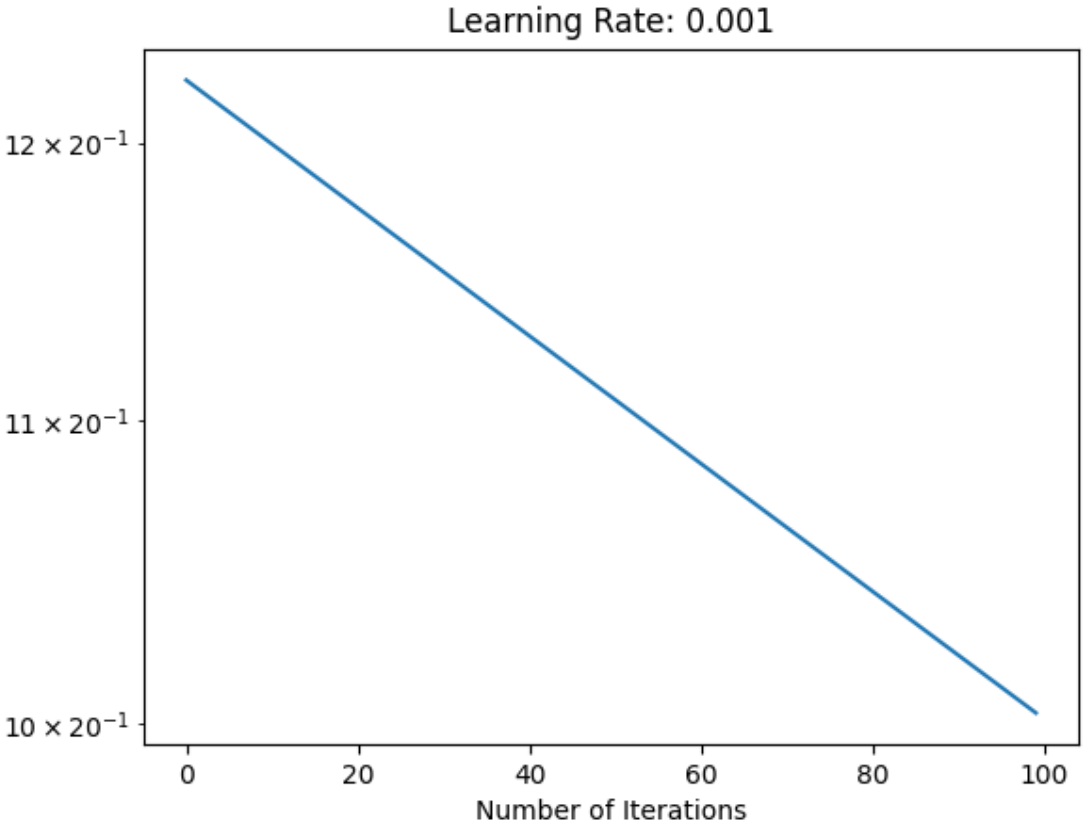
The following figure demonstrates the decision boundary of the final result.

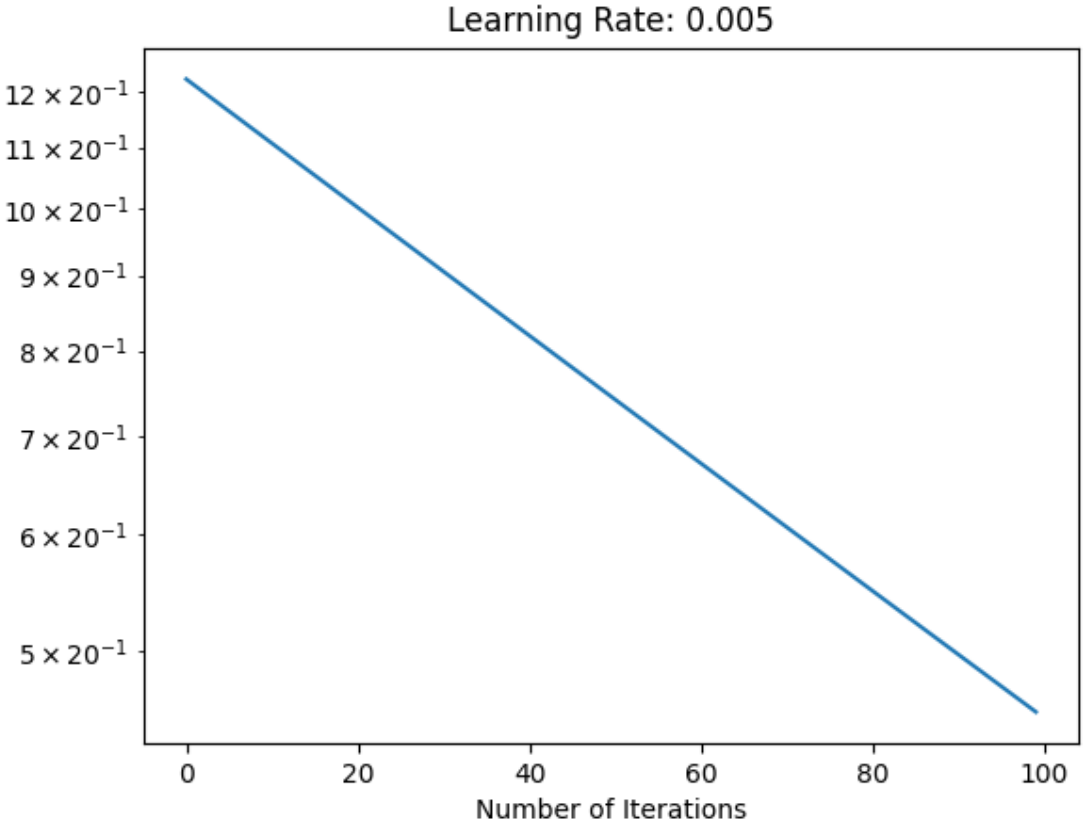


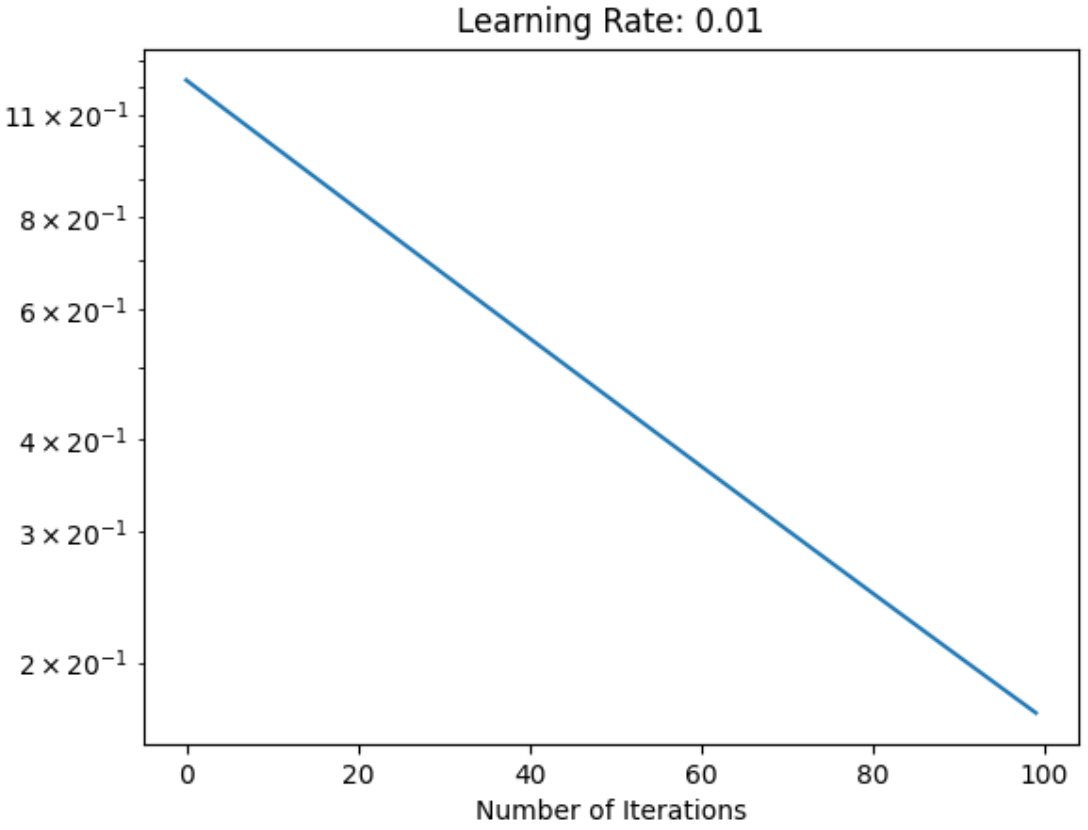
The decision boundary of this figure implies that the dataset is linearly separable.

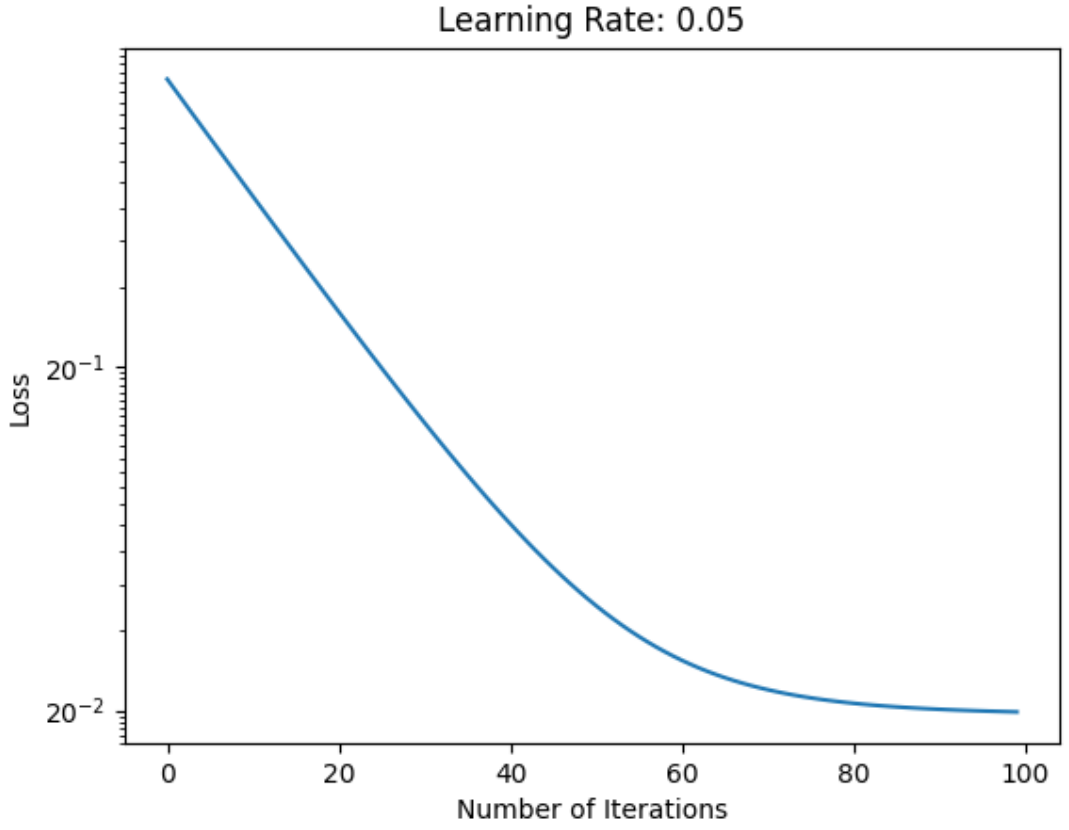
2. Linear Regression

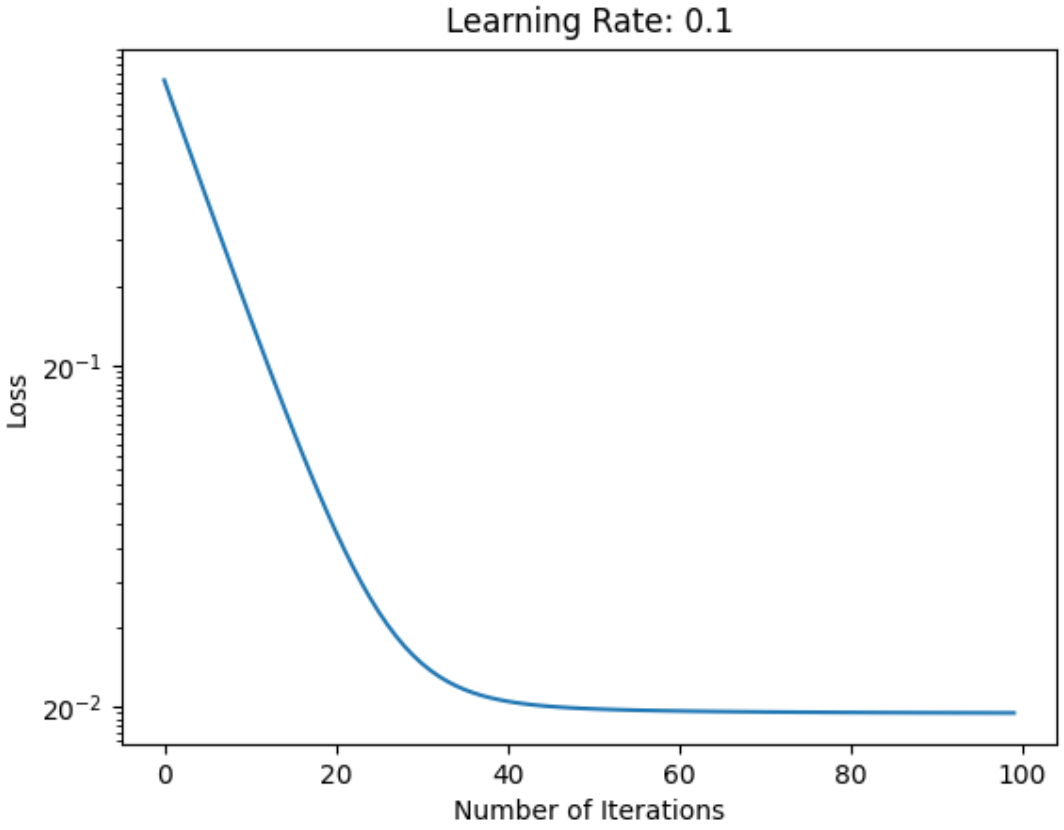
Here I plot the loss function (y) over iterations (x) for various learning rates. It is clear that for the learning rates 0.001, 0.005, 0.01, 0.05 and 0.1, the loss function could not converge well over 100 iterations, whereas for the learning rates 5 and 10, the learning rates are too large that void the algorithm entirely. A learning rate of 0.5 or 1 did a great job in this dataset, so some value in between should be perfect as a result of fine-tuning. I chose 0.8 as the learning rate and decided to early stop the programme by setting the number of iterations to 30 since more iterations would introduce trivial improvements (See the last figure).

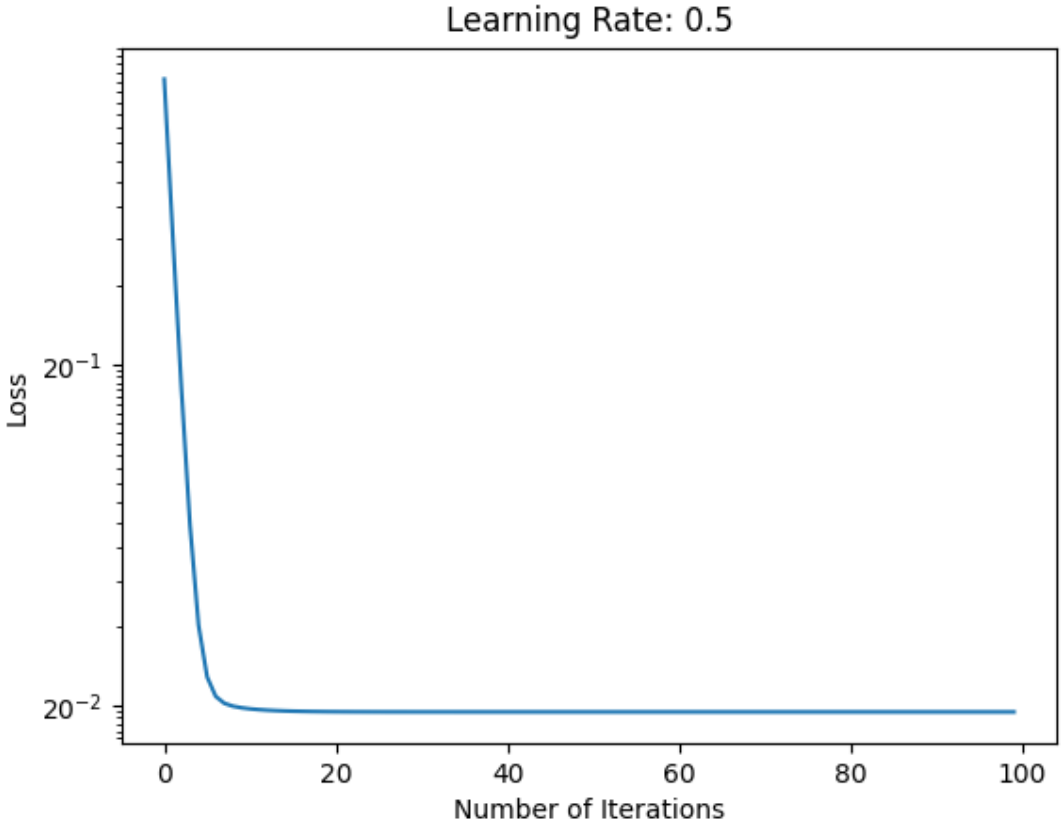

Learning Rate: 0.001
12×20^{-1}
11×20^{-1}
10×20^{-1}
0
20
40
60
80
100
Number of Iterations

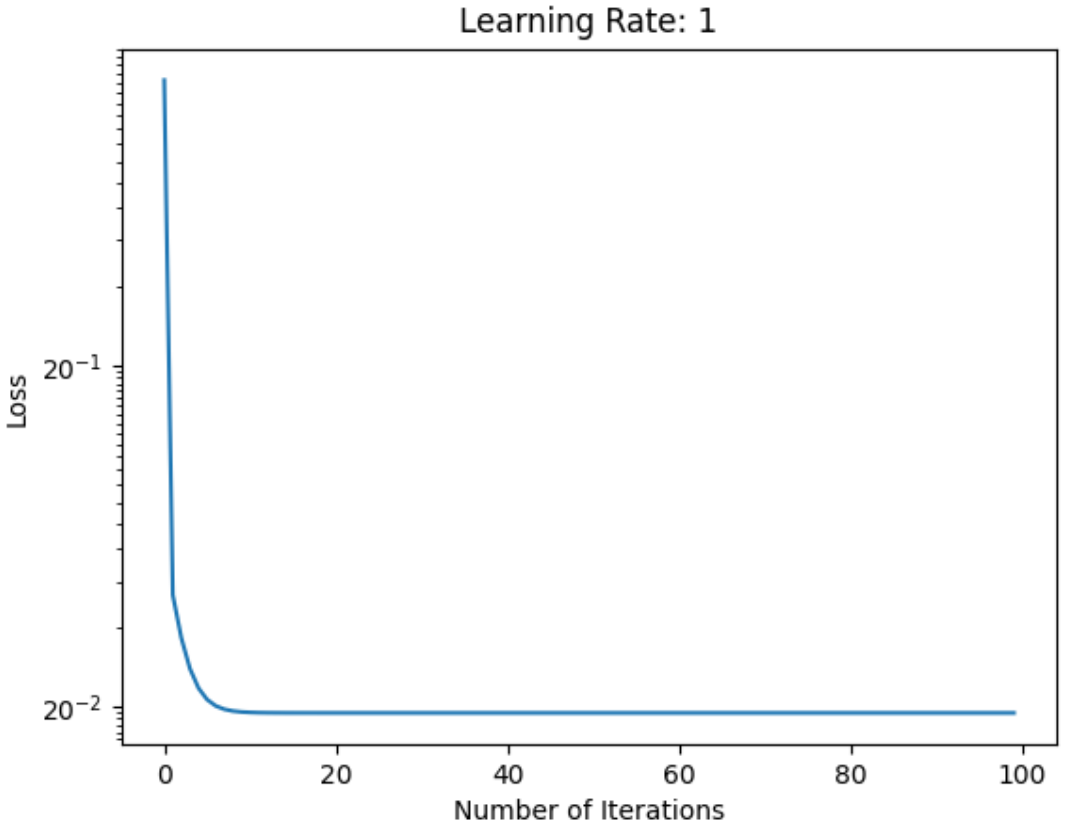

Learning Rate: 0.005
12×20^{-1}
11×20^{-1}
10×20^{-1}
9×20^{-1}
8×20^{-1}
7×20^{-1}
6×20^{-1}
5×20^{-1}
0
20
40
60
80
100
Number of Iterations

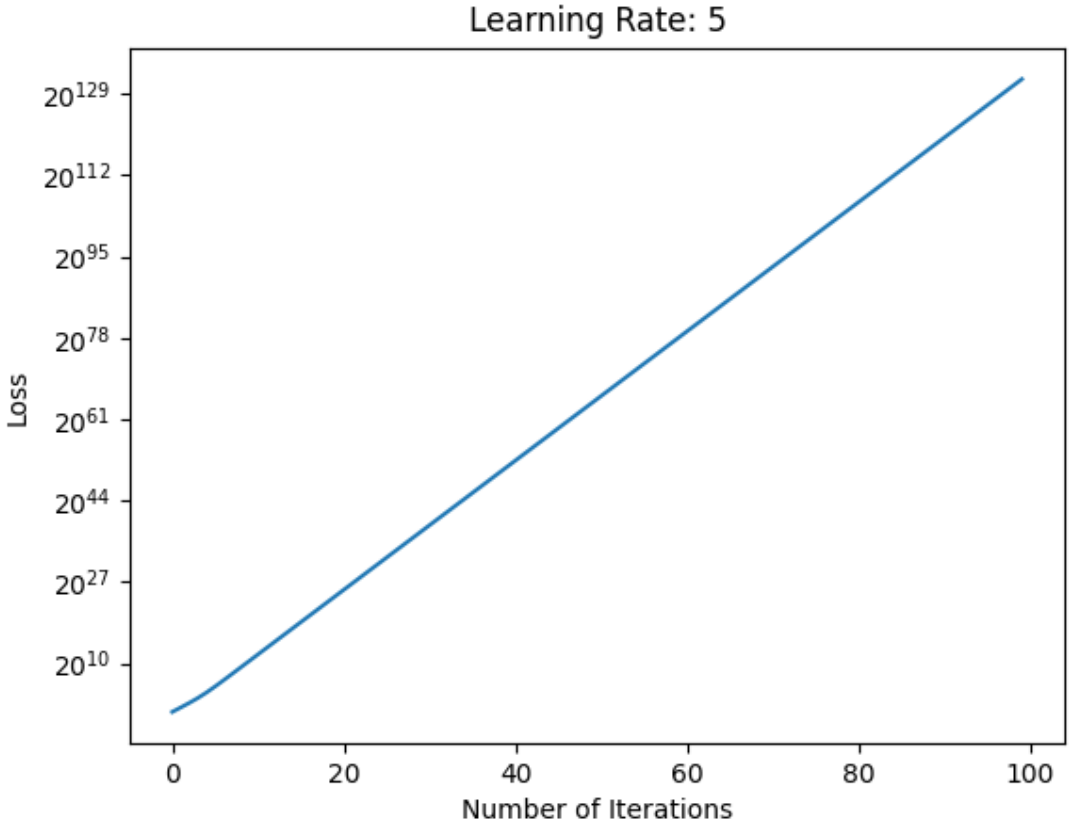

Learning Rate: 0.01
11×20^{-1}
8×20^{-1}
6×20^{-1}
4×20^{-1}
3×20^{-1}
2×20^{-1}
0
20
40
60
80
100
Number of Iterations

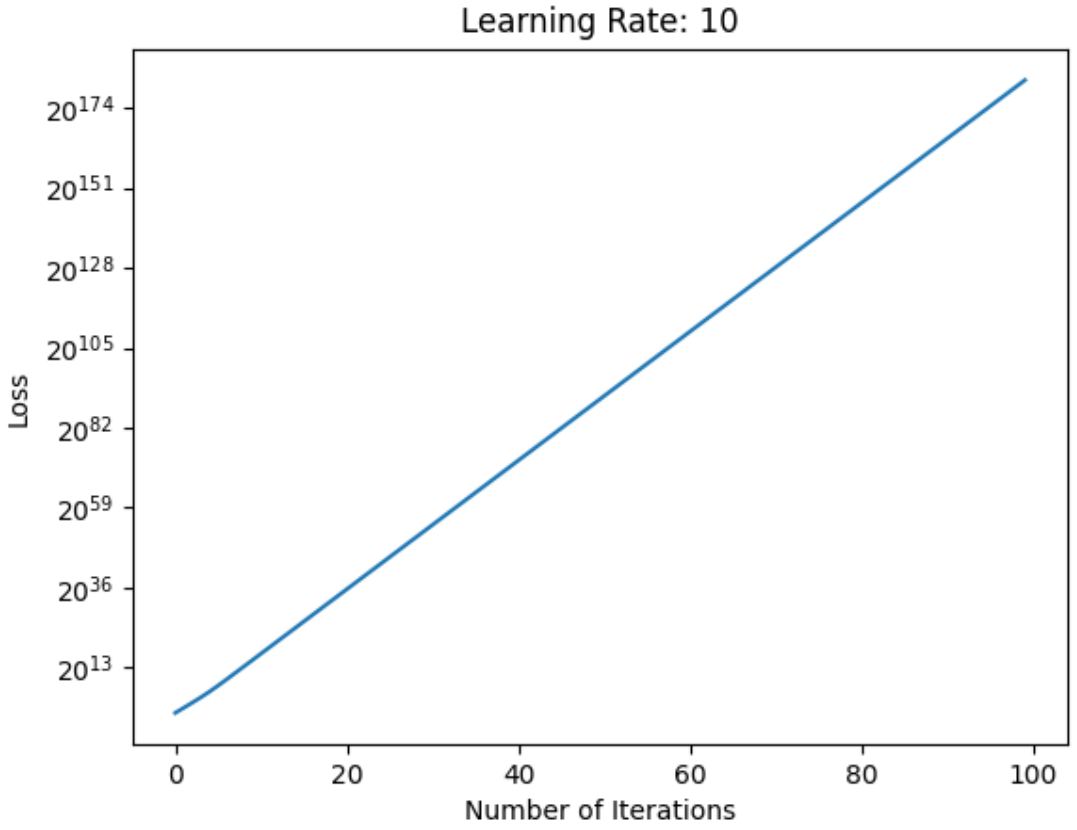

Learning Rate: 0.05
Loss
20^{-1}
20^{-2}
0
20
40
60
80
100
Number of Iterations

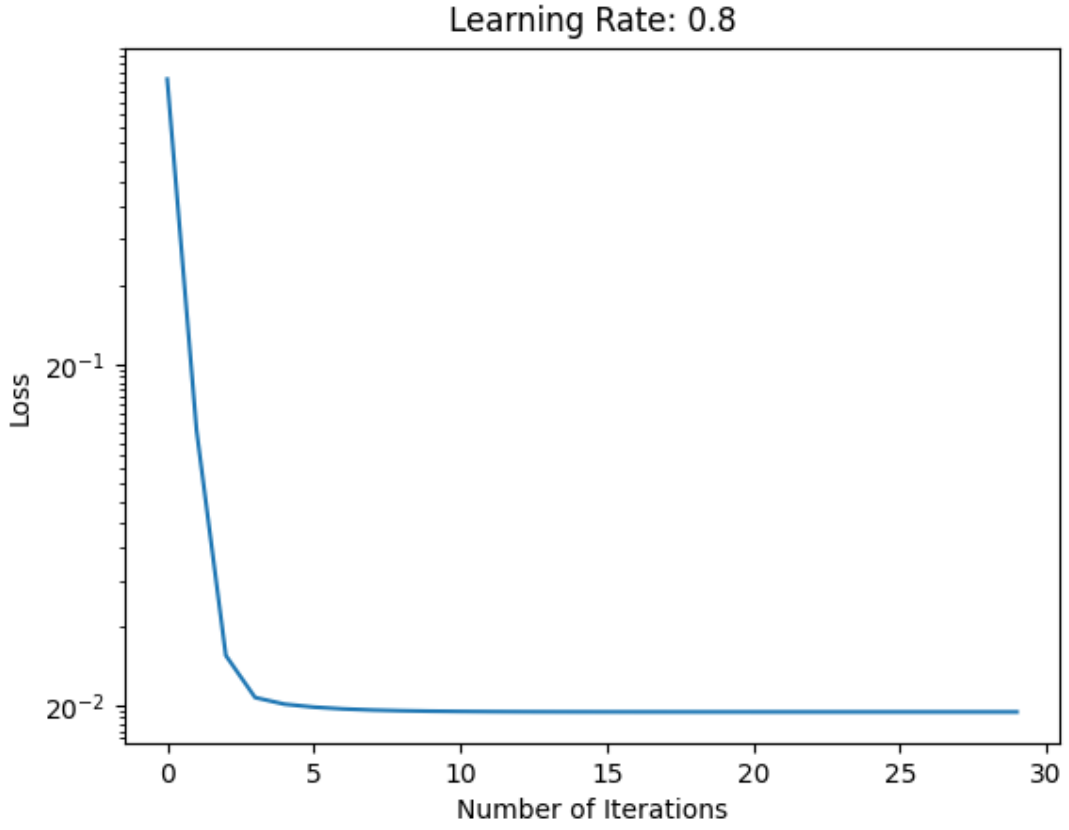

Learning Rate: 0.1
Loss
20^{-1}
20^{-2}
0
20
40
60
80
100
Number of Iterations


Learning Rate: 0.5
Loss
20^{-1}
20^{-2}
0
20
40
60
80
100
Number of Iterations


Learning Rate: 1
Loss
20^{-1}
20^{-2}
0
20
40
60
80
100
Number of Iterations

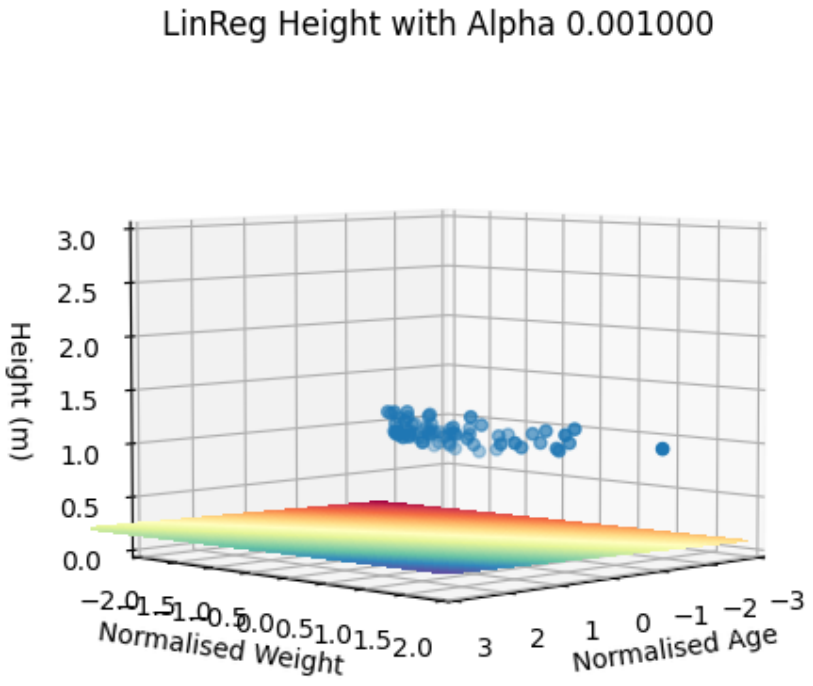

Learning Rate: 5
Loss
20^{129}
20^{112}
20^{95}
20^{78}
20^{61}
20^{44}
20^{27}
20^{10}
0
20
40
60
80
100
Number of Iterations


Learning Rate: 10
Loss
20^{174}
20^{151}
20^{128}
20^{105}
20^{82}
20^{59}
20^{36}
20^{13}
0
20
40
60
80
100
Number of Iterations

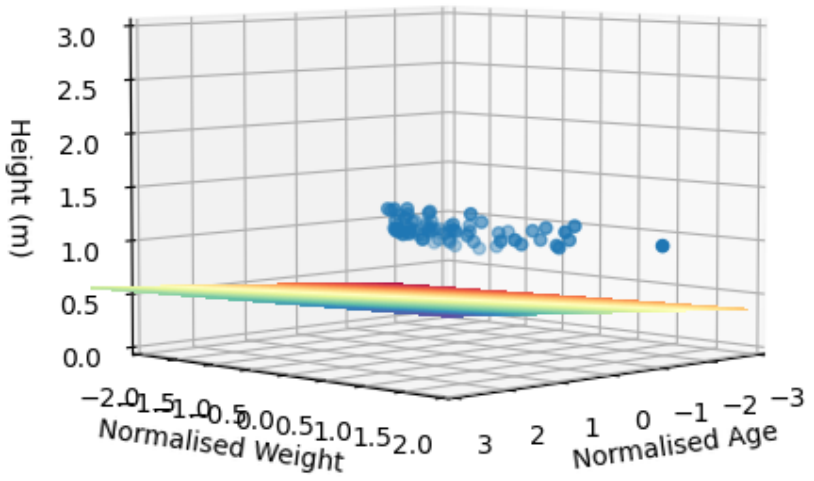

Learning Rate: 0.8
Loss
20^{-1}
20^{-2}
0
5
10
15
20
25
30
Number of Iterations

Actually, in the field of machine learning, we often see that a small learning rate (alpha) can be more accurate than a large learning rate, but it suffers from getting converged. Also, it may get stuck in the local minimum since it is hard to jump out. On the other hand, a large learning rate may converge quickly and may escape from local minimums, but it may also mislead the algorithm and let it go in the entirely opposite direction, which is the case for 5 and 10 in this problem. We now make some surface plots to visualise our results and verify that with alpha = 0.8 and iteration = 30, we get good performance (see next page).

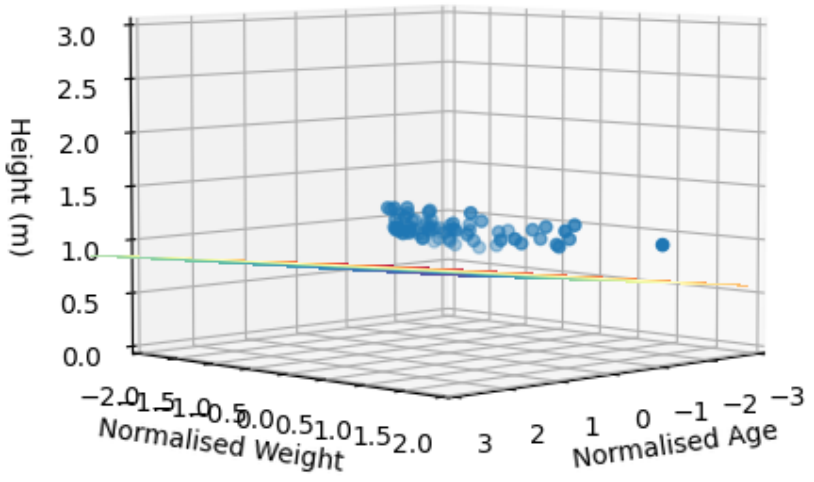
We may find that with small alpha, the surface is far away from the dataset. With a very large alpha, the surface even disappears. With appropriate alpha, most of the data points are just on the surface and our choice is good. Please note that we use normalised weight and age here.



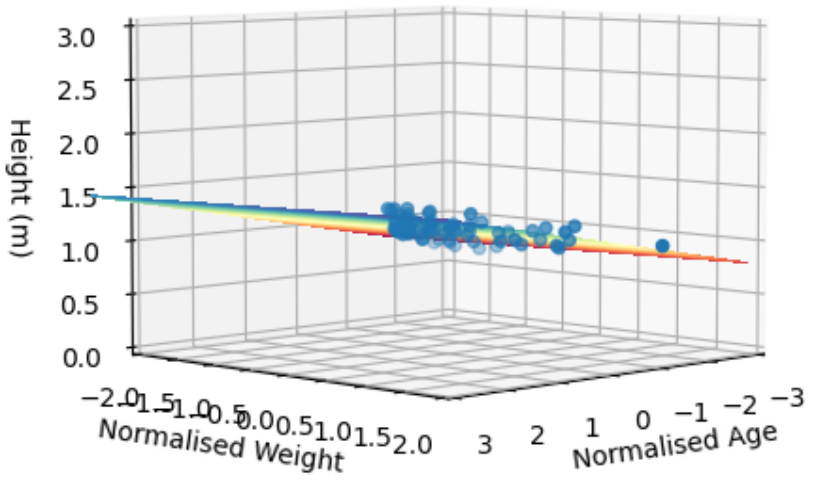
LinReg Height with Alpha 0.005000



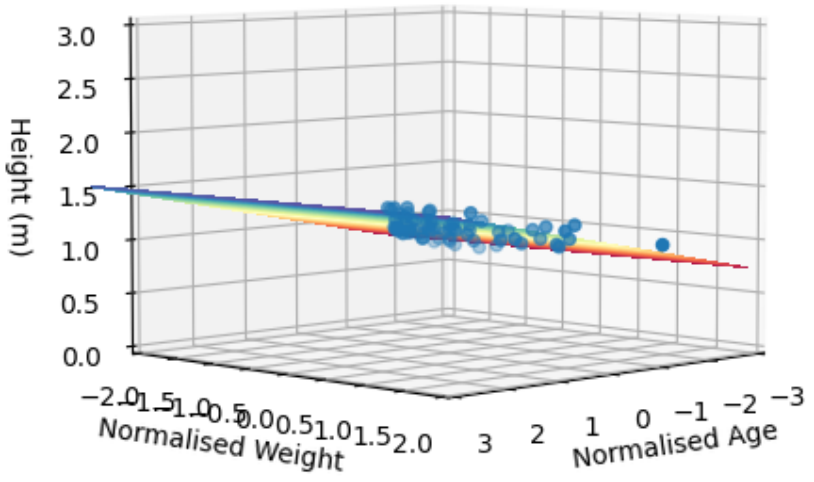
LinReg Height with Alpha 0.010000



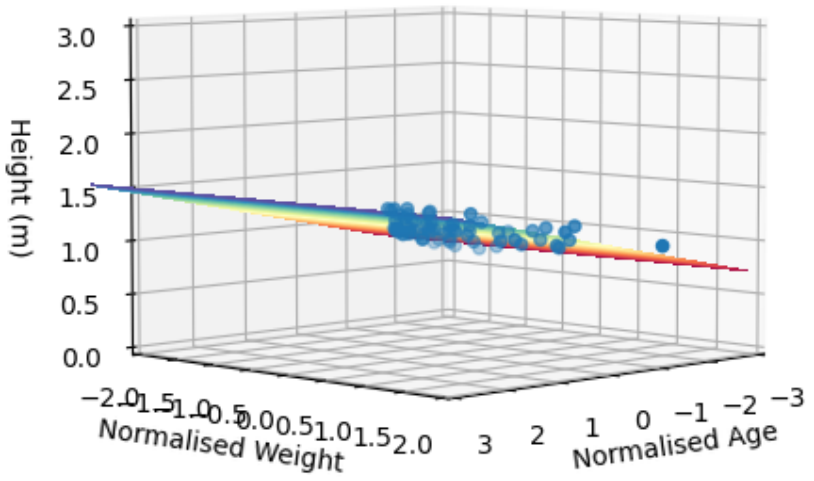
LinReg Height with Alpha 0.050000



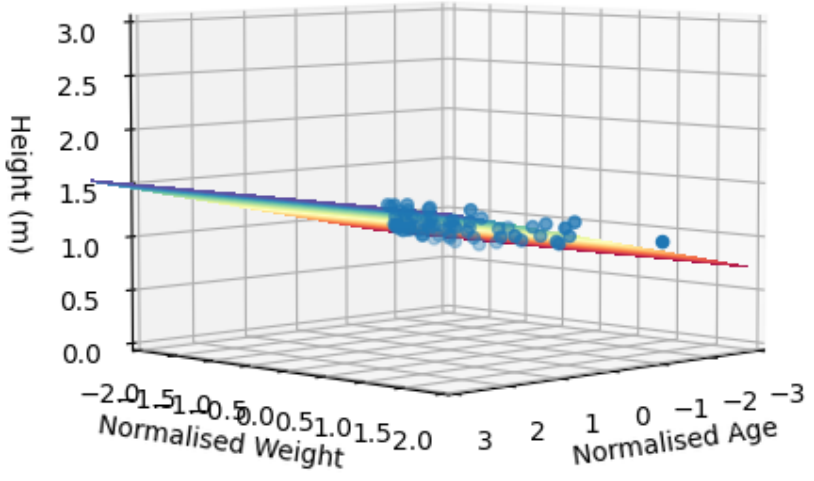
LinReg Height with Alpha 0.100000



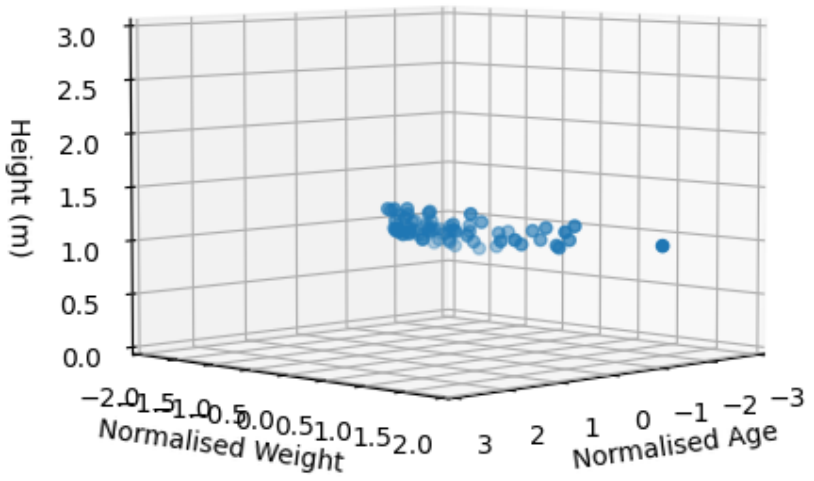
LinReg Height with Alpha 0.500000



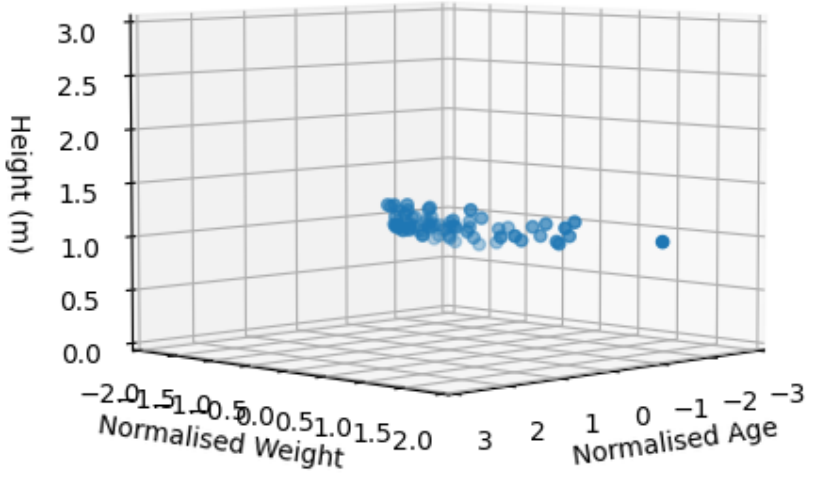
LinReg Height with Alpha 1.000000



LinReg Height with Alpha 5.000000



LinReg Height with Alpha 10.000000



This one used an alpha of 0.8 and 30 iterations. We see that it has good performance, with the dataset on the surface.