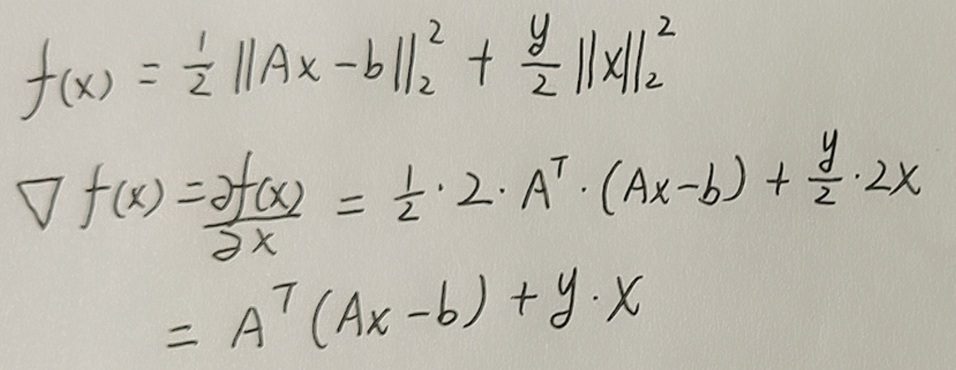
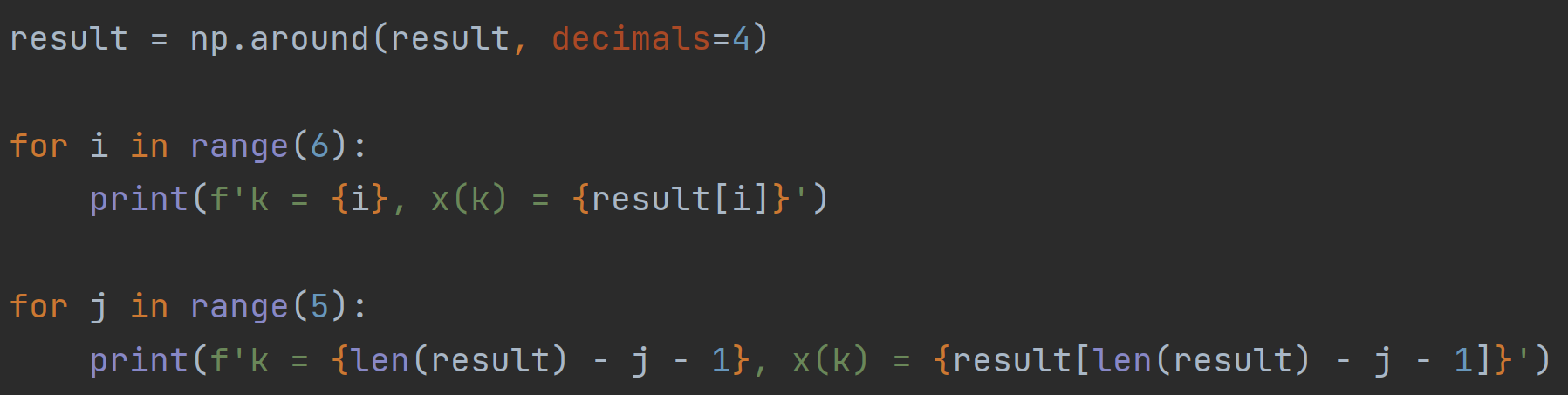
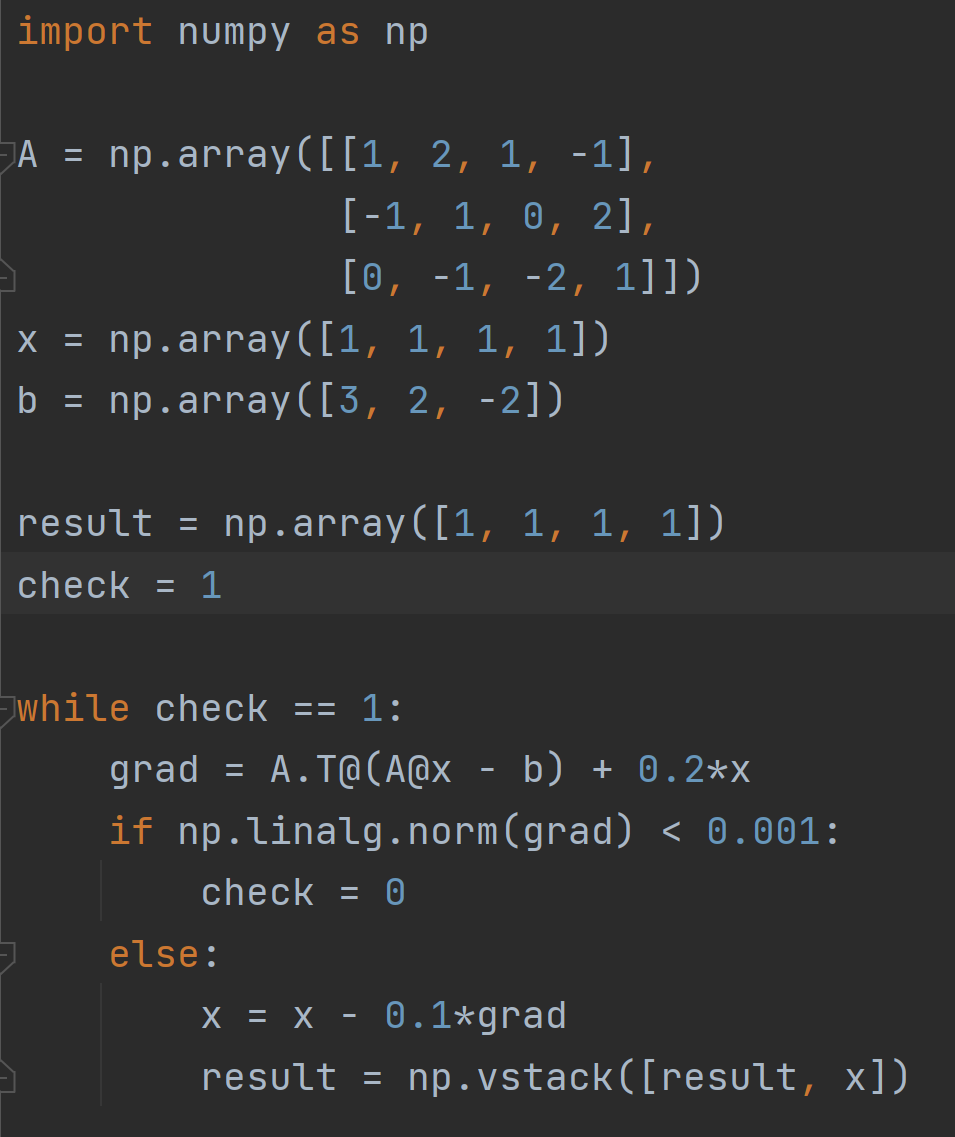
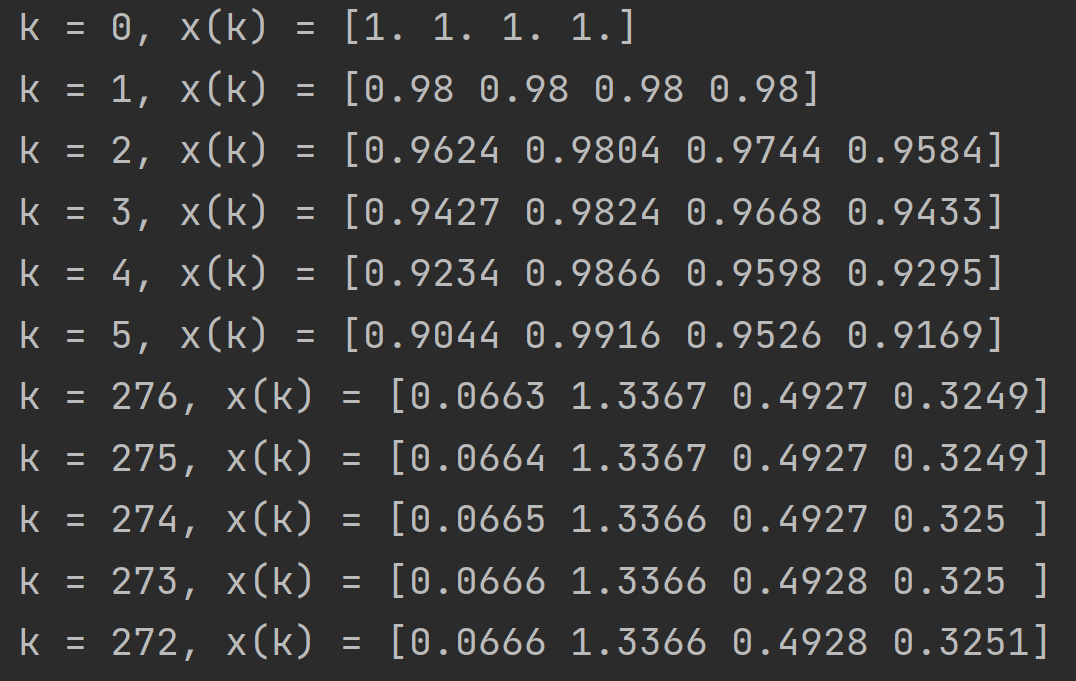
Question1(a)



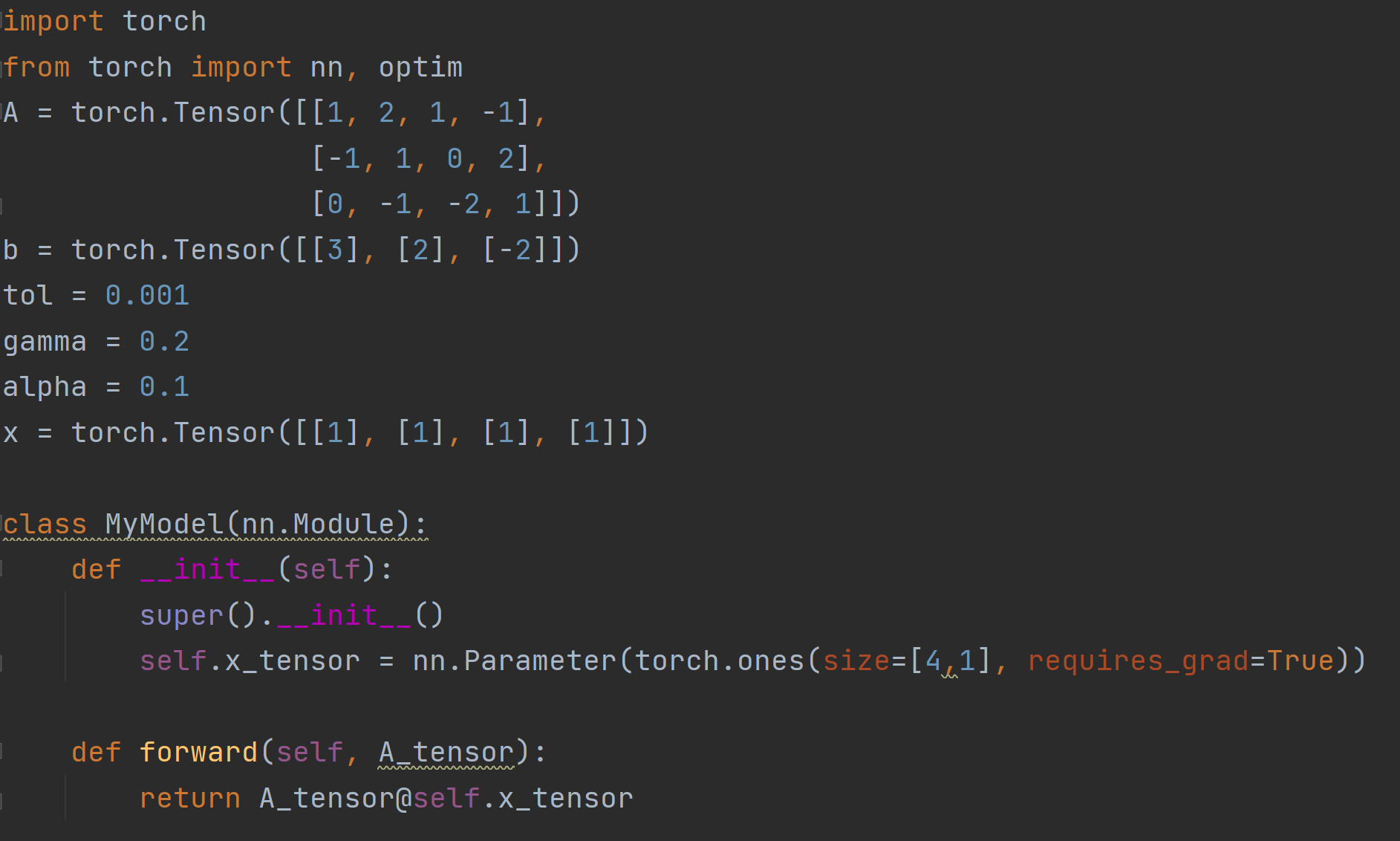
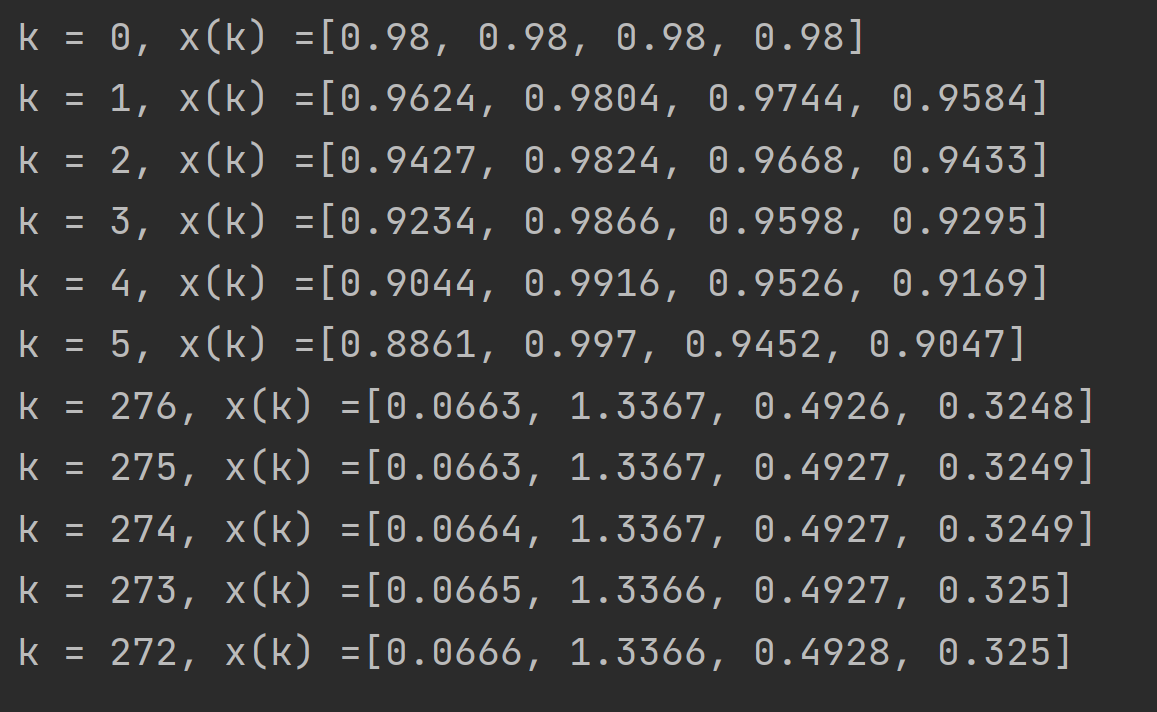




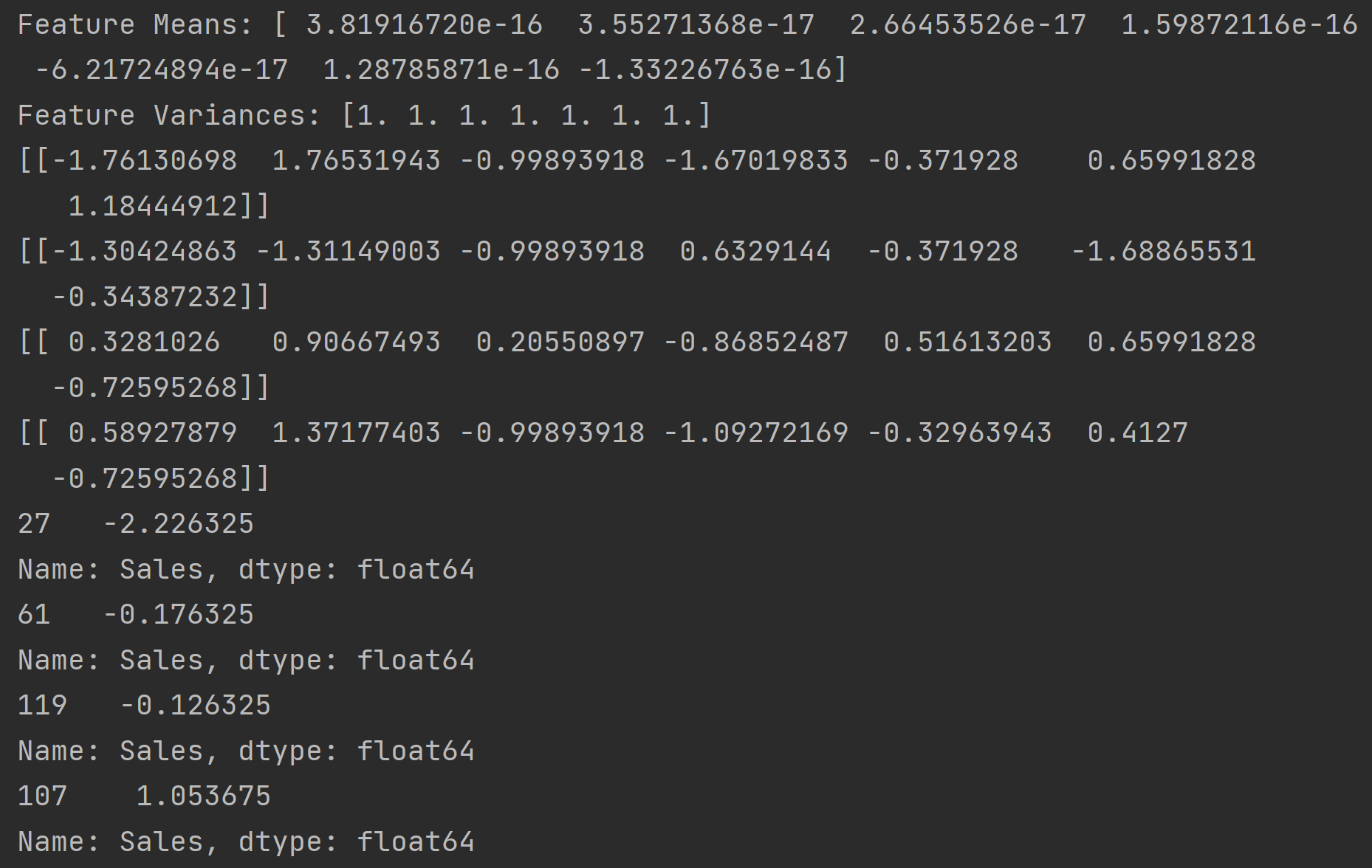
Question1(b)

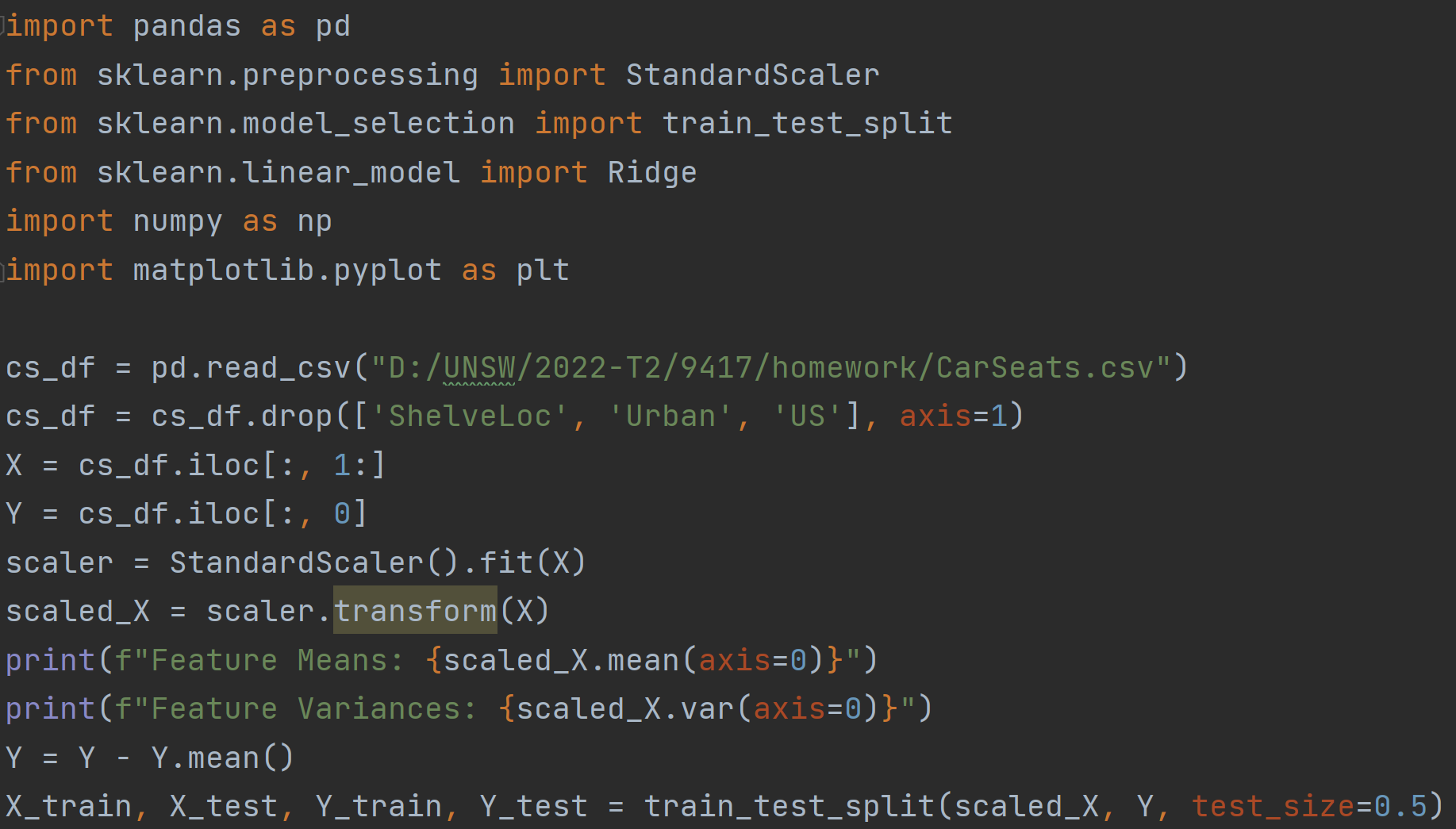
This means that the minimum algorithm converges to three decimal places. If the righthand side smaller (say 0.0001), the output precision will be higher, the first five rows of data will not be affected, the value of K in the last five rows of data will increase, and x(K) will decrease accordingly.

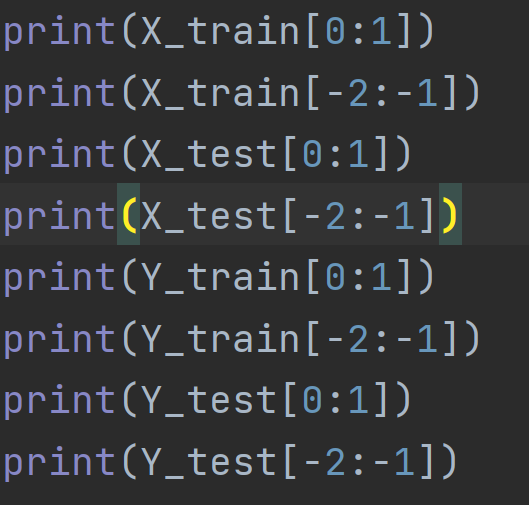
Question1(c)



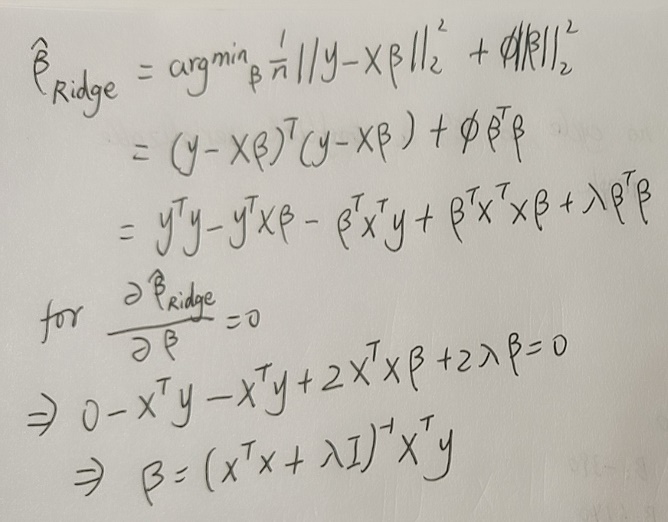
Question1(d)

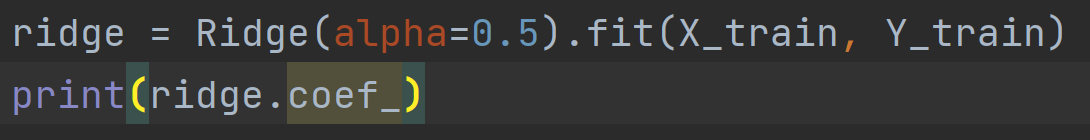
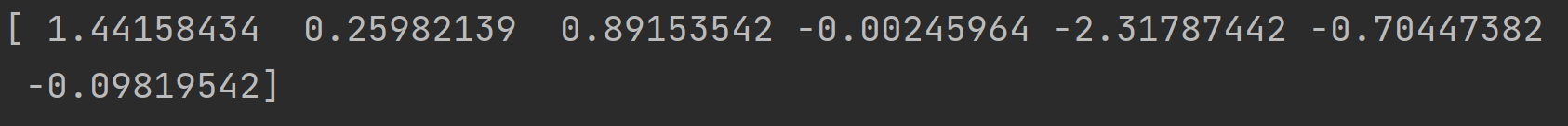




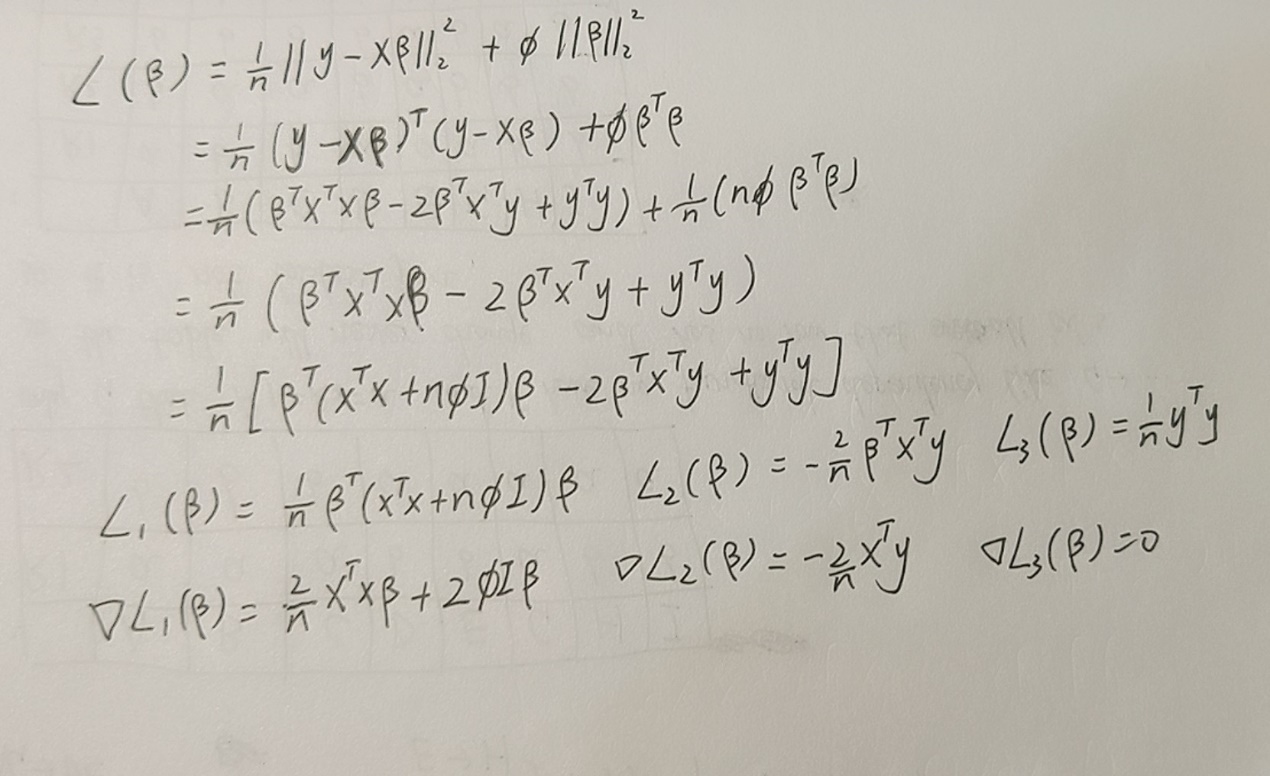


Question1(e)

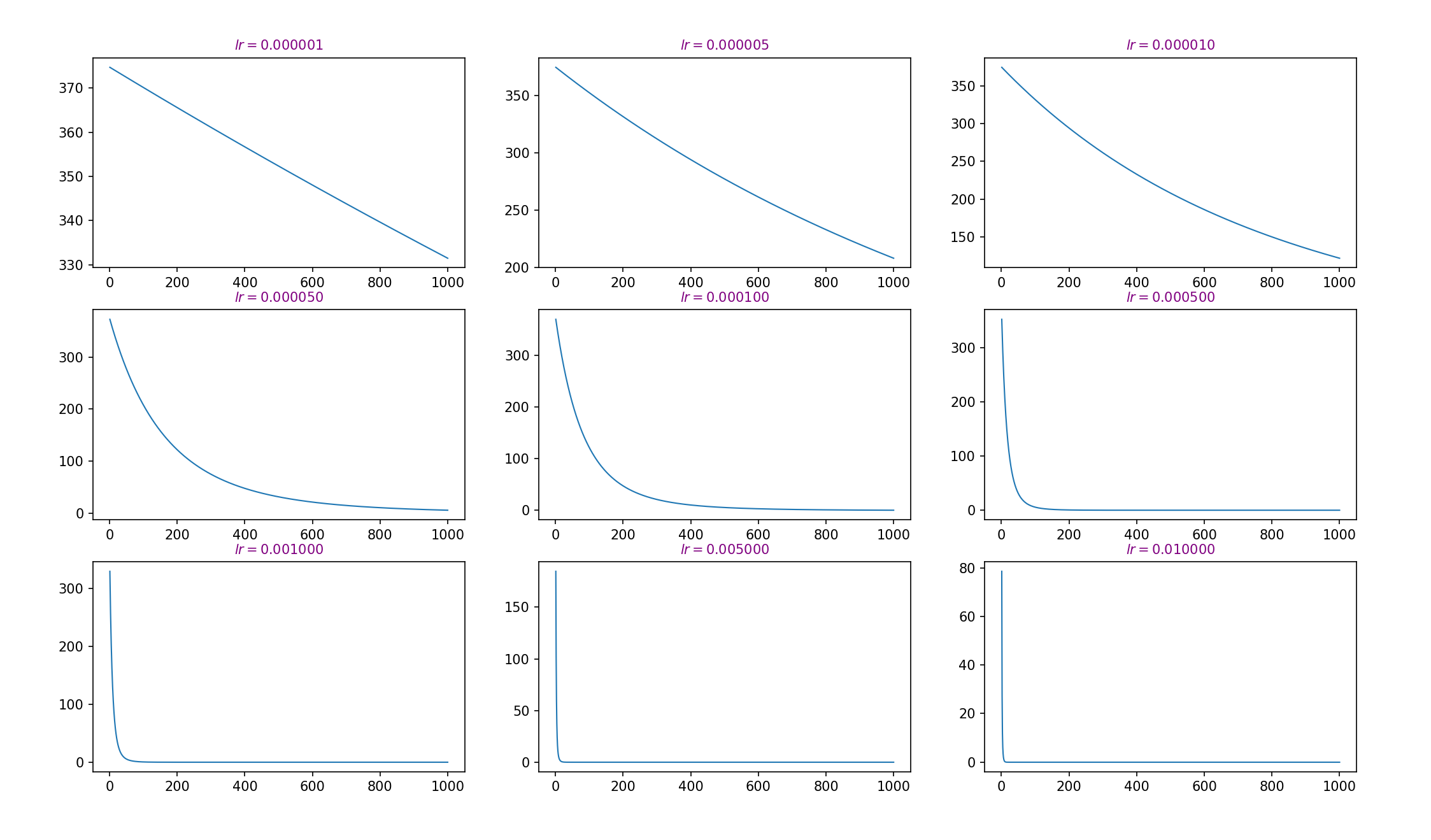


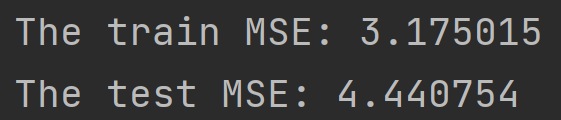
Question1(f)

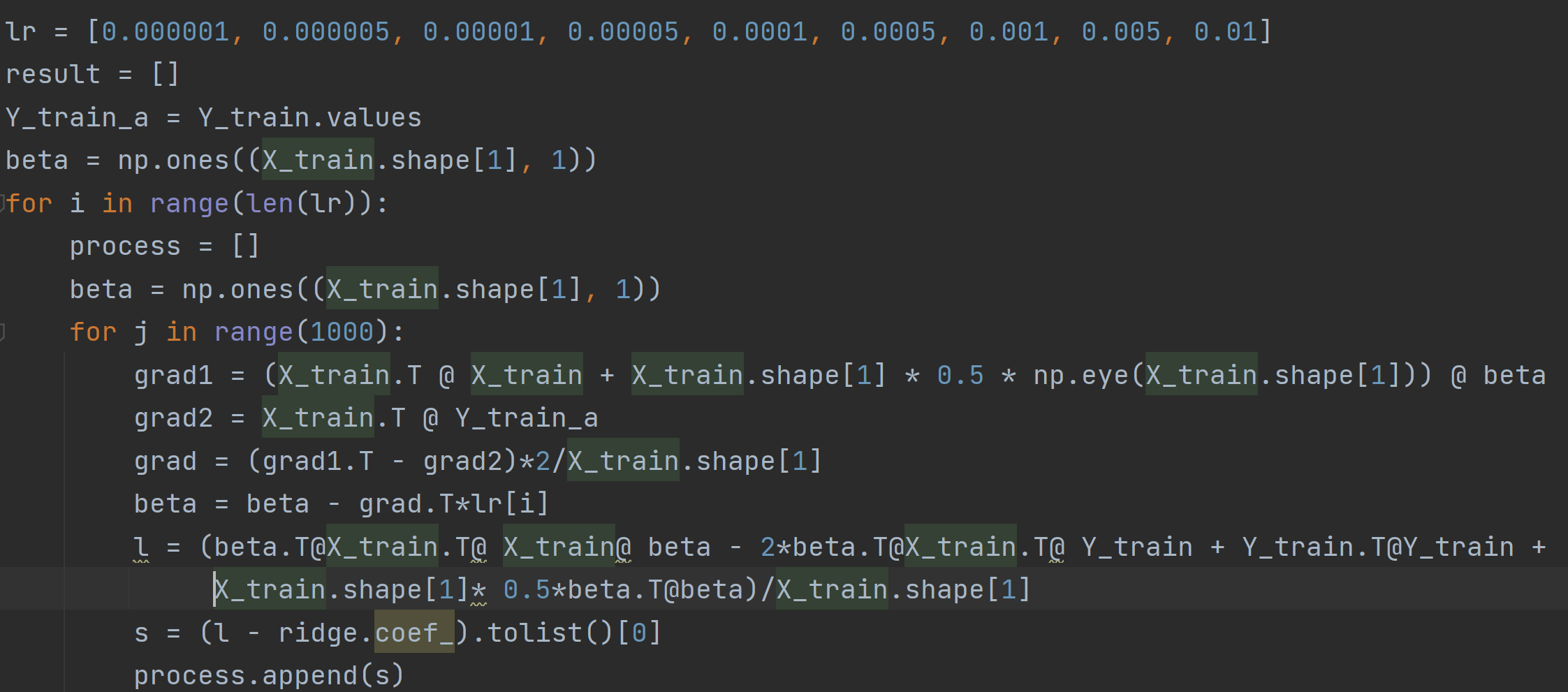
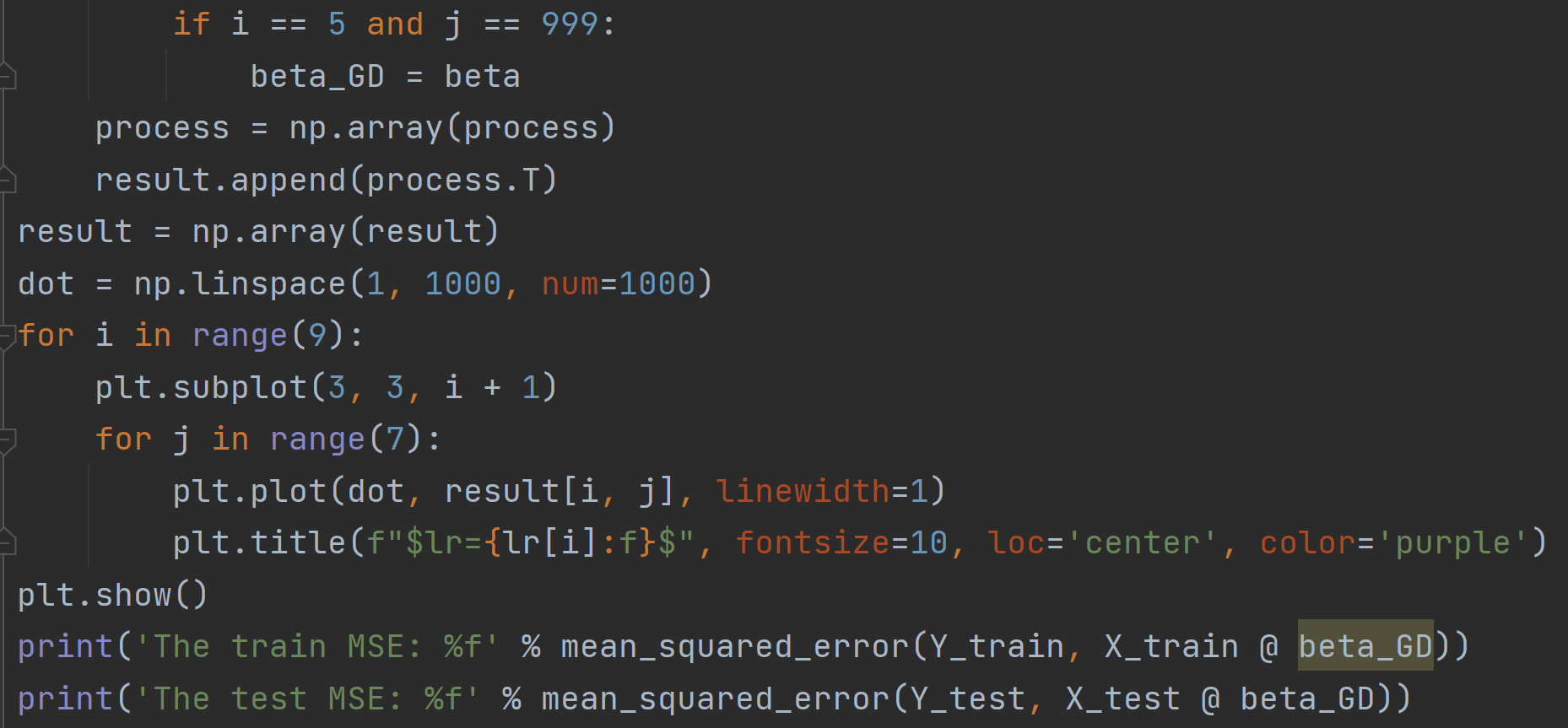


Question1(g)

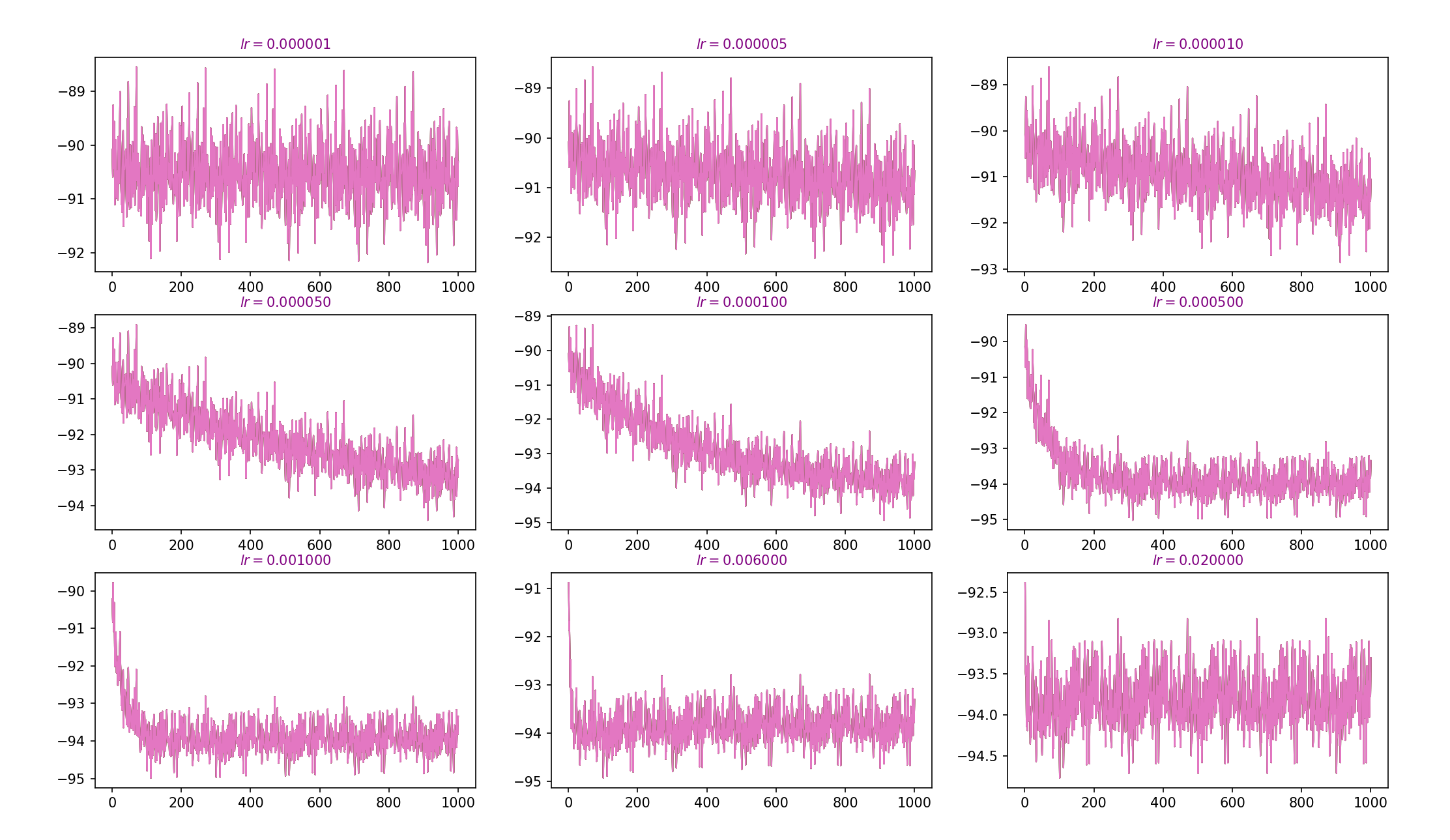


In my opinion, the progress is best when the step size is 0.005. As can be seen from the image, when the step size is 0.005, the value gradually converges and finally becomes stable, and the accuracy obtained will be higher than that of other step sizes.

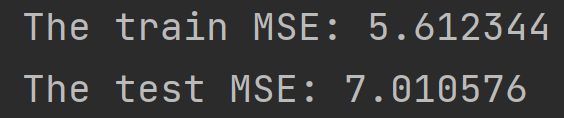


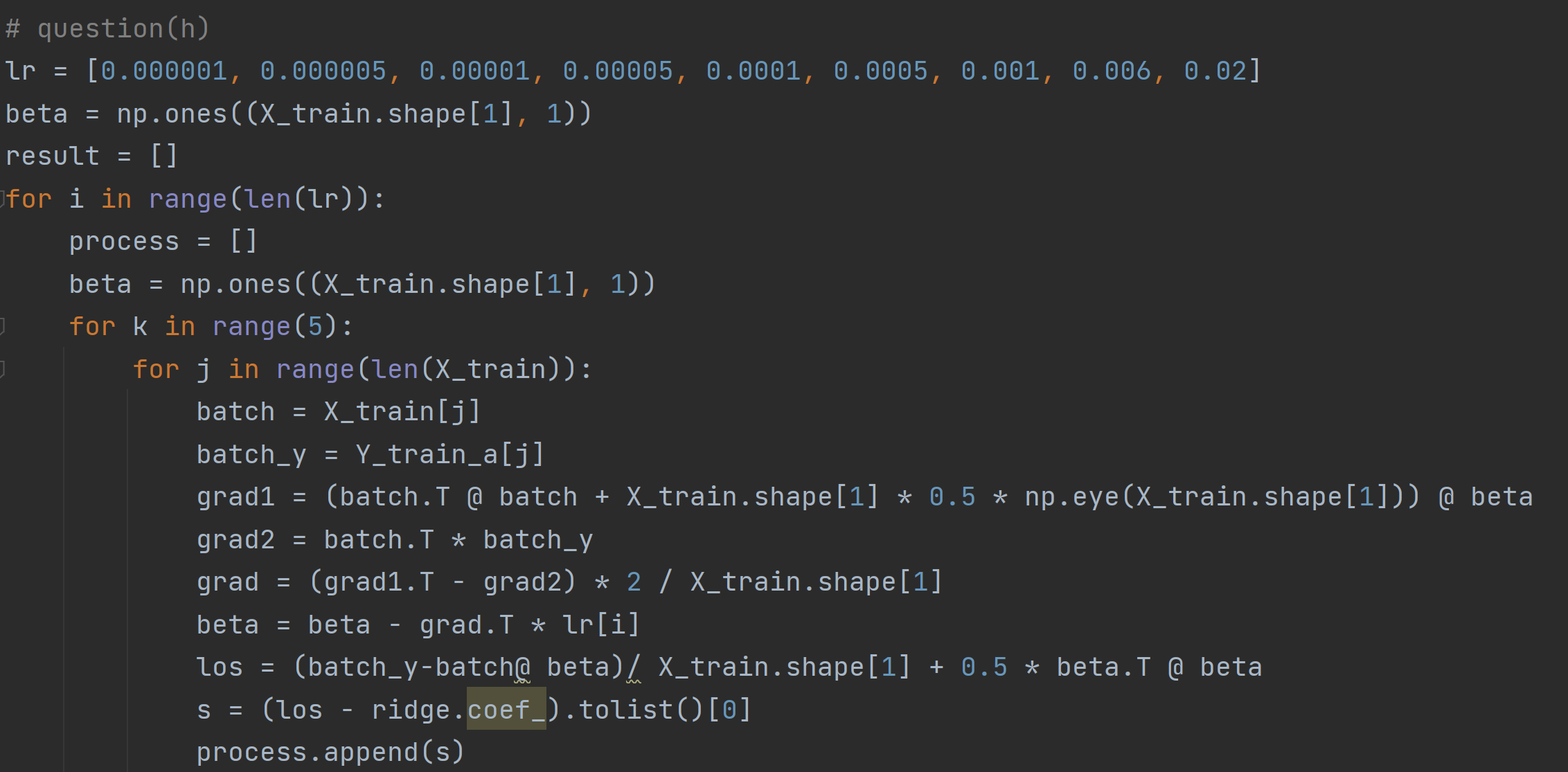
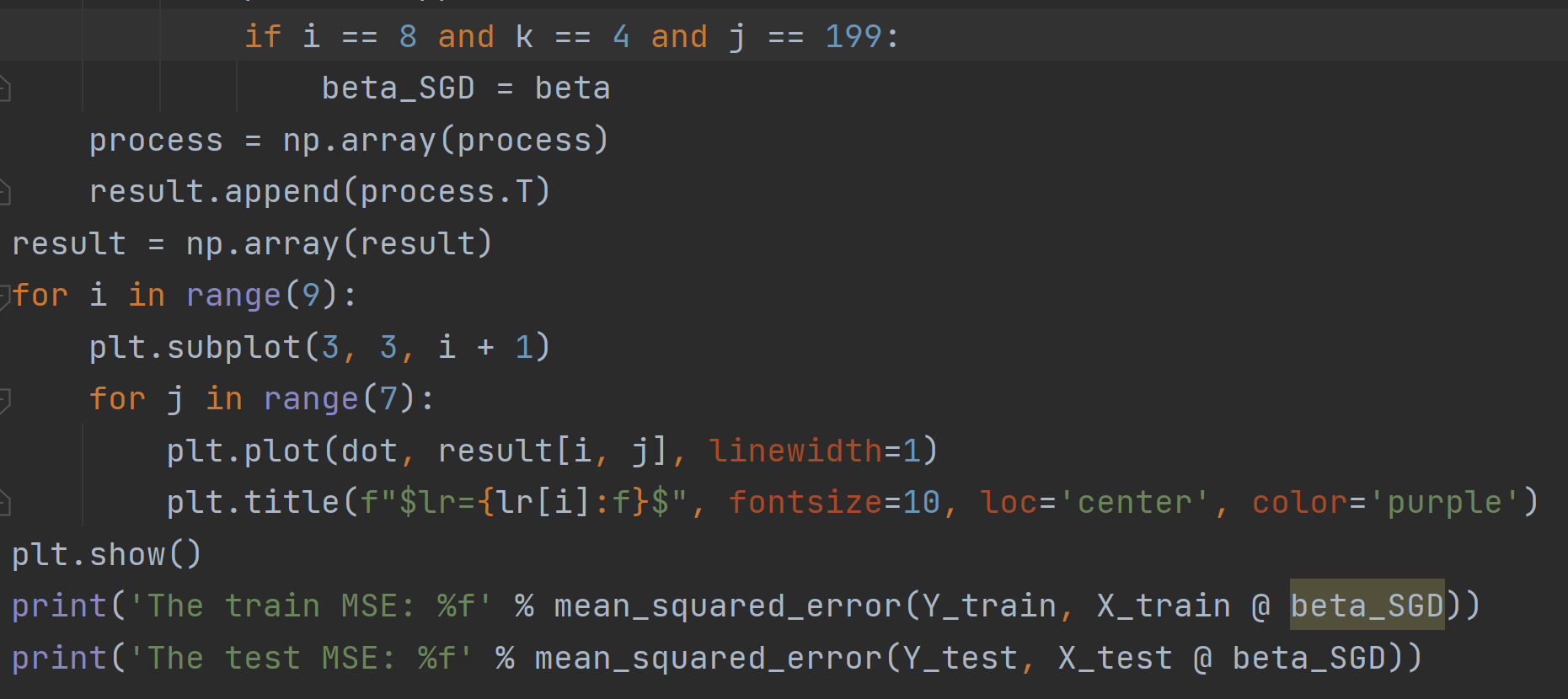
 

Question1(h)



the best step-size choice:0.02



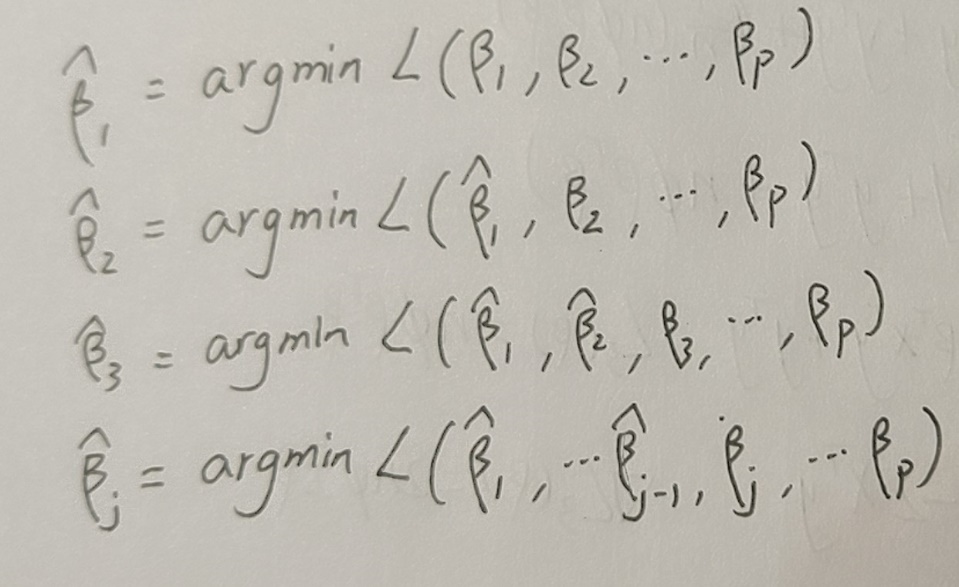
Stochastic gradient descent may not go in the right direction each time it is updated, thus causing optimization fluctuations.

Question1(i)

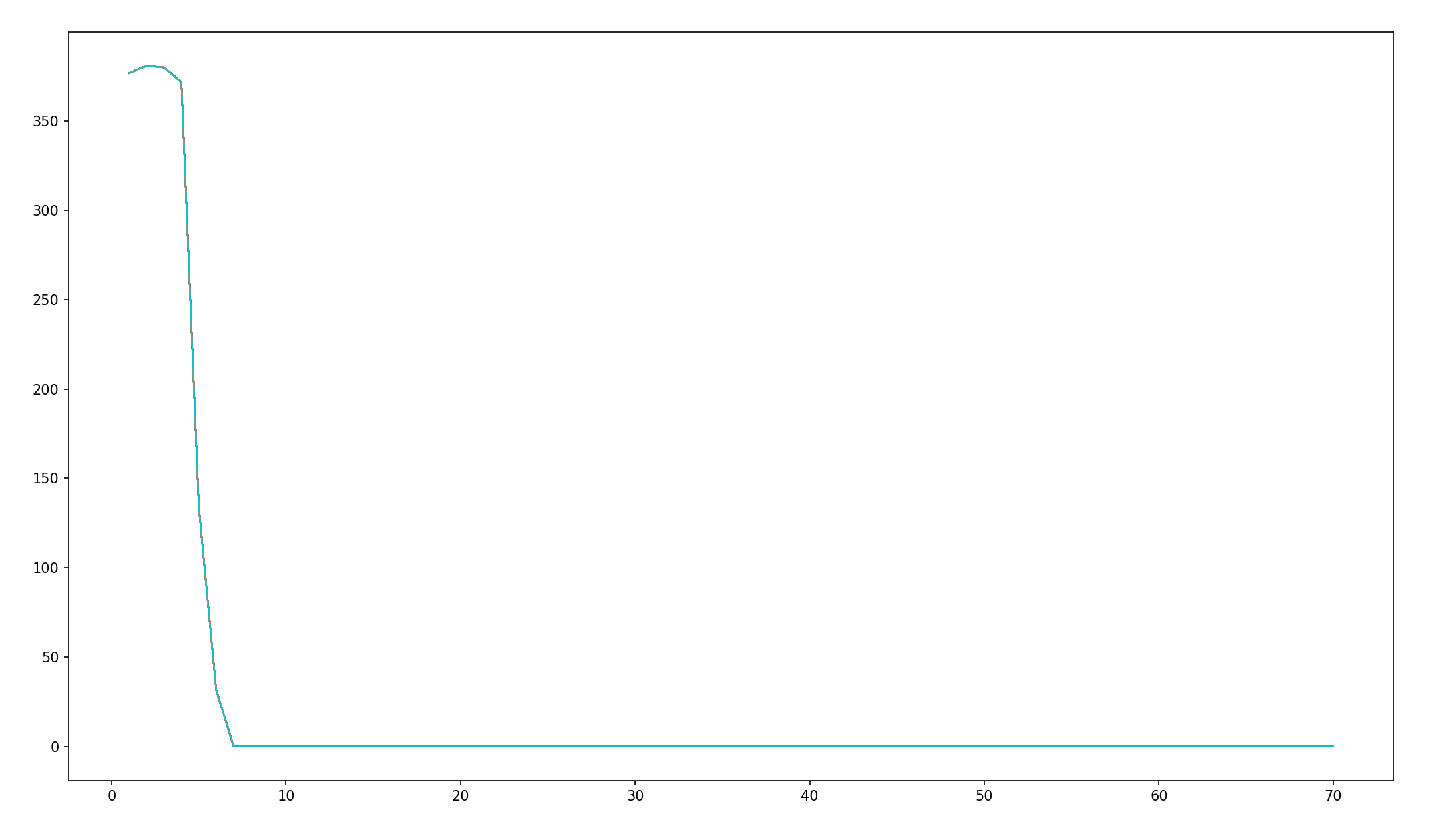
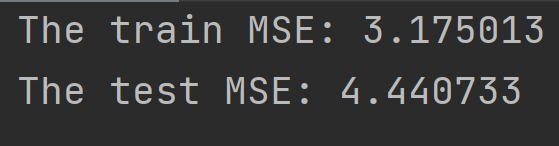
I prefer GD because the MSE for GD is smaller and more accurate in this data set.

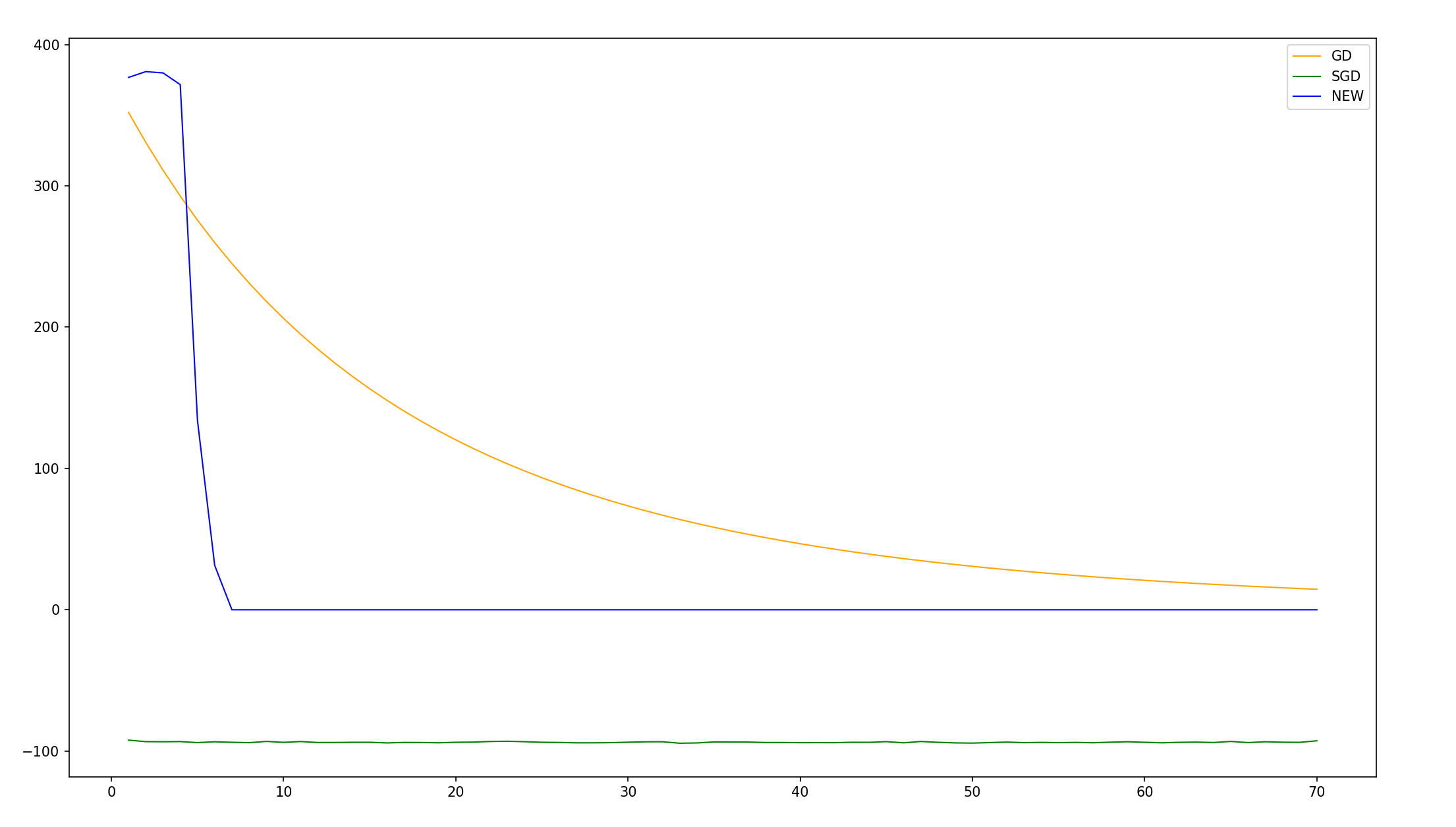
An advantage of the fluctuation caused by stochastic gradient descent is that for non-convex functions, the optimization direction may jump from the current local minimum point to another better local minimum point, and eventually converge to a better local extremum point, or even the global extremum point. That is, use GD for convex functions and SGD for non-convex functions.

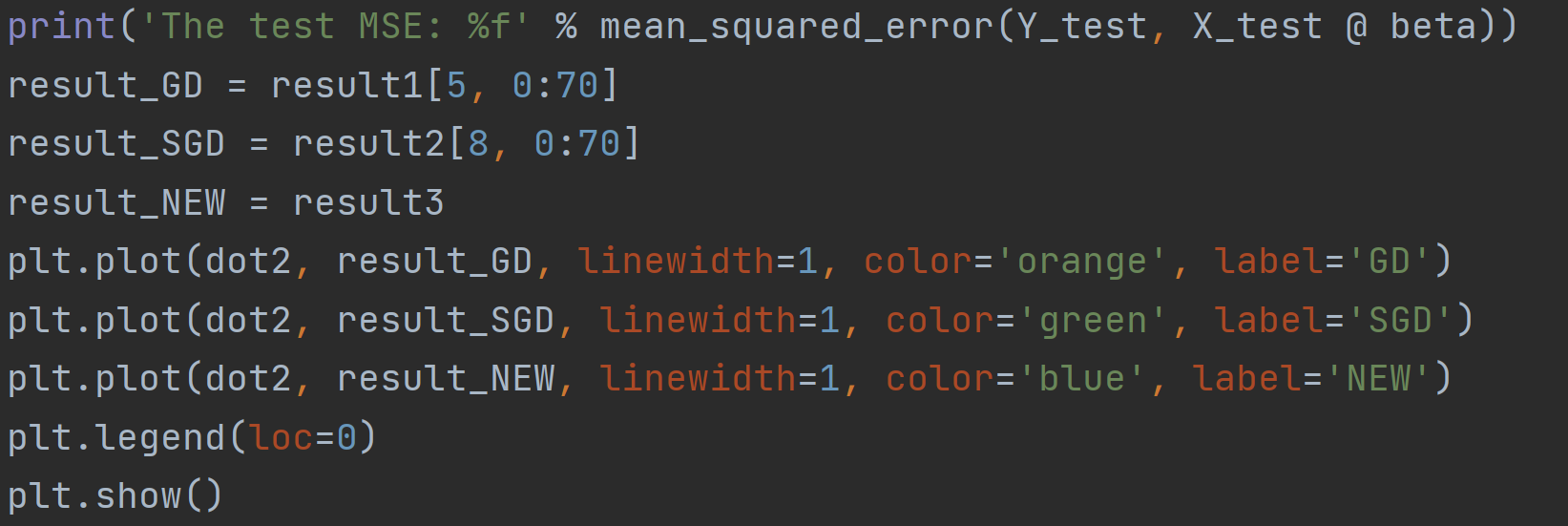
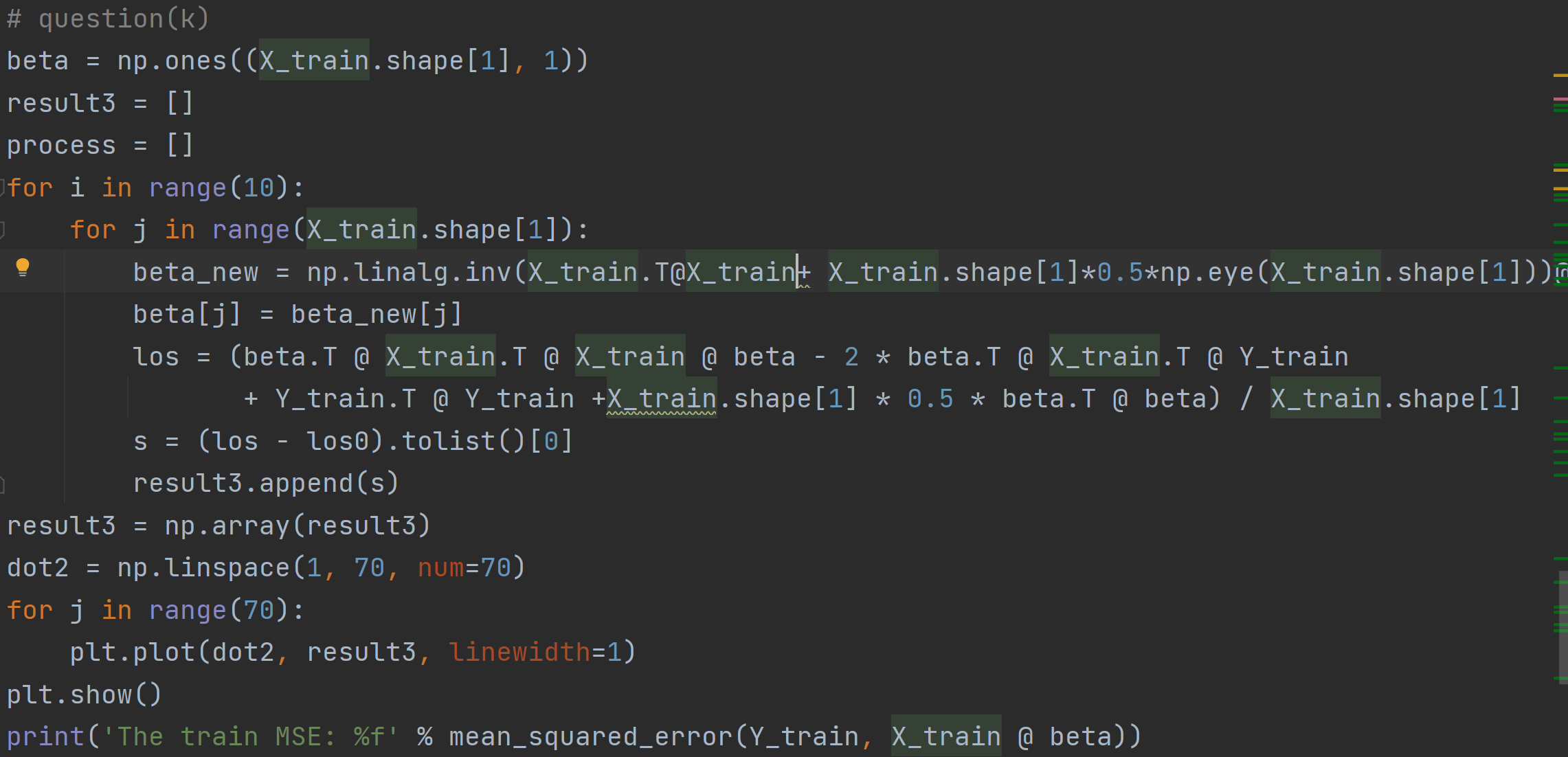
Question1(j)



Question1(k)





Question1(l)

My results in parts (e)-(k) are more reliable, because the features in this dataset basically follow gaussian normal distribution.