Machine Learning HW

Team Member:

0516017 李柏毅

0516059 劉嘉豪

0516306 尤健羽

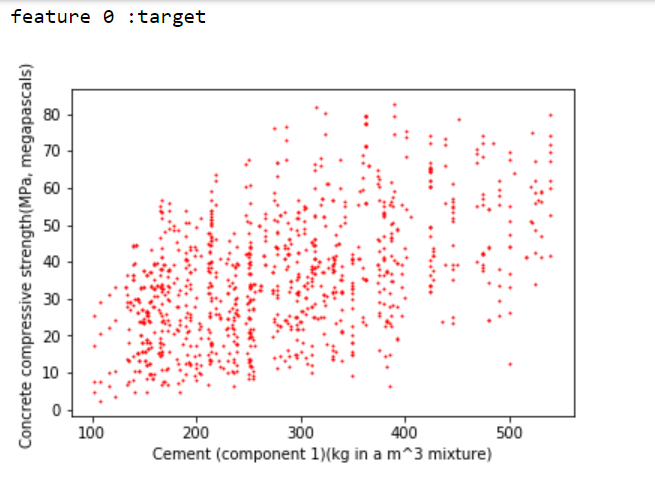
0516319 傅信瑀

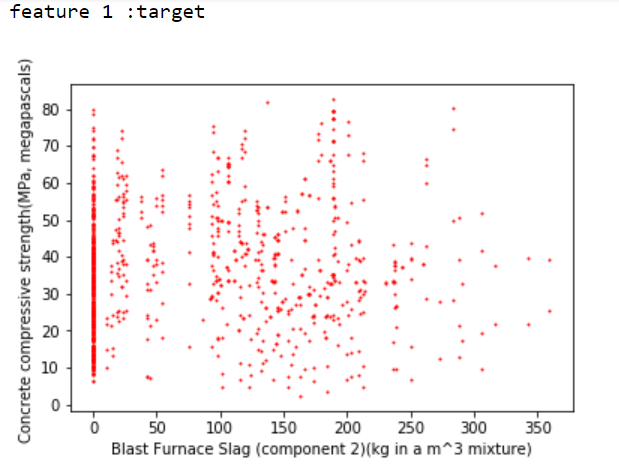
0516322 朱蝶

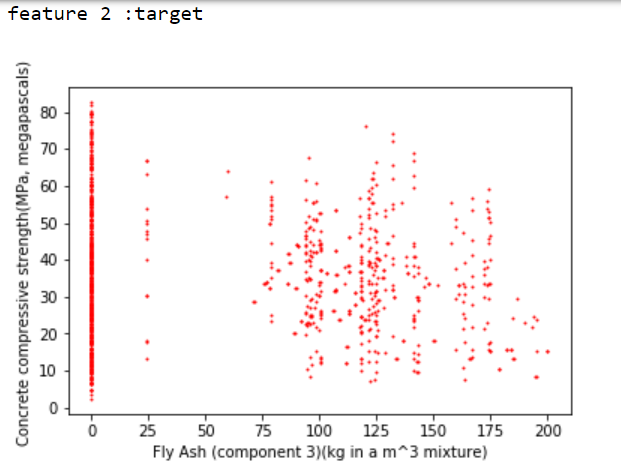
1. What environments the members are using:

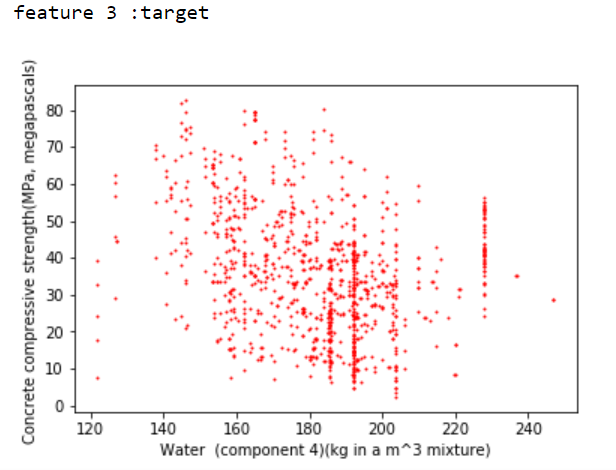
Use Jupyter as the environment

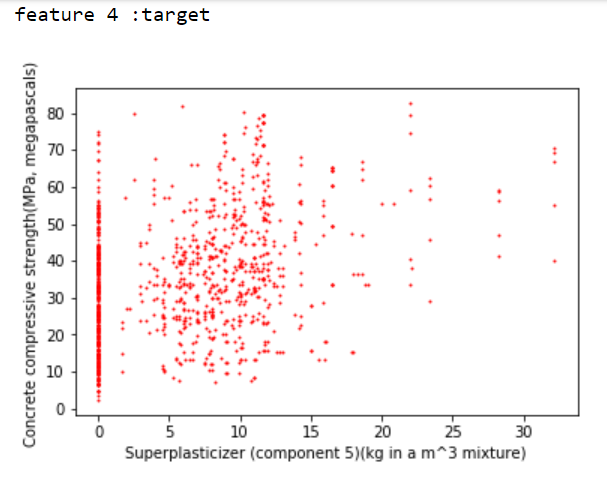
1. Visualization of all the features with the target

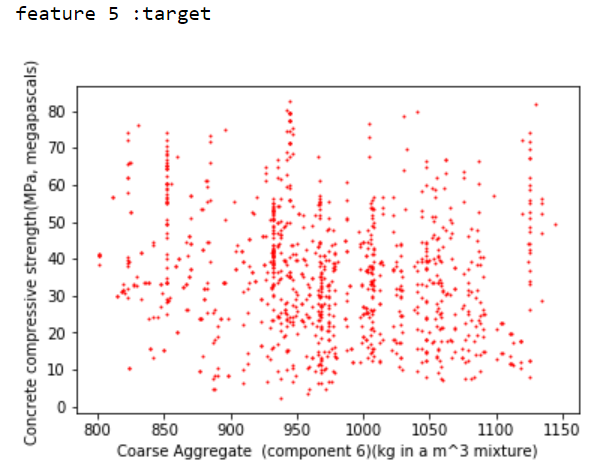


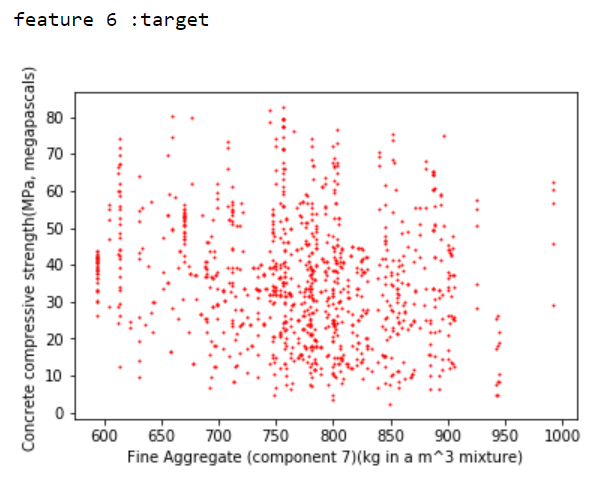


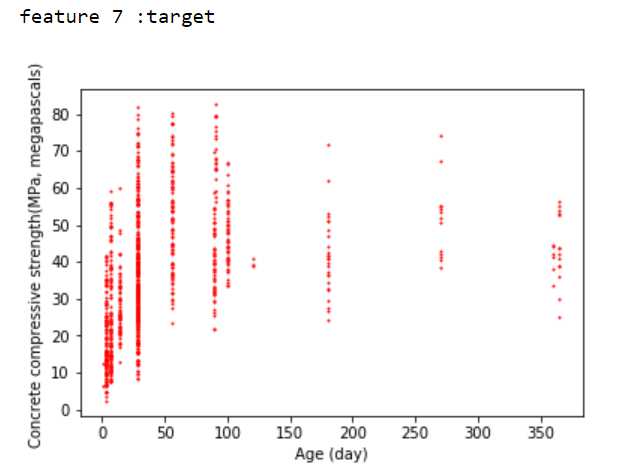






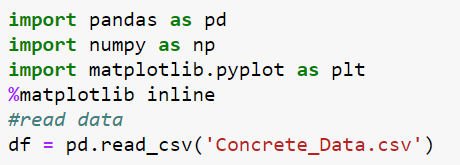


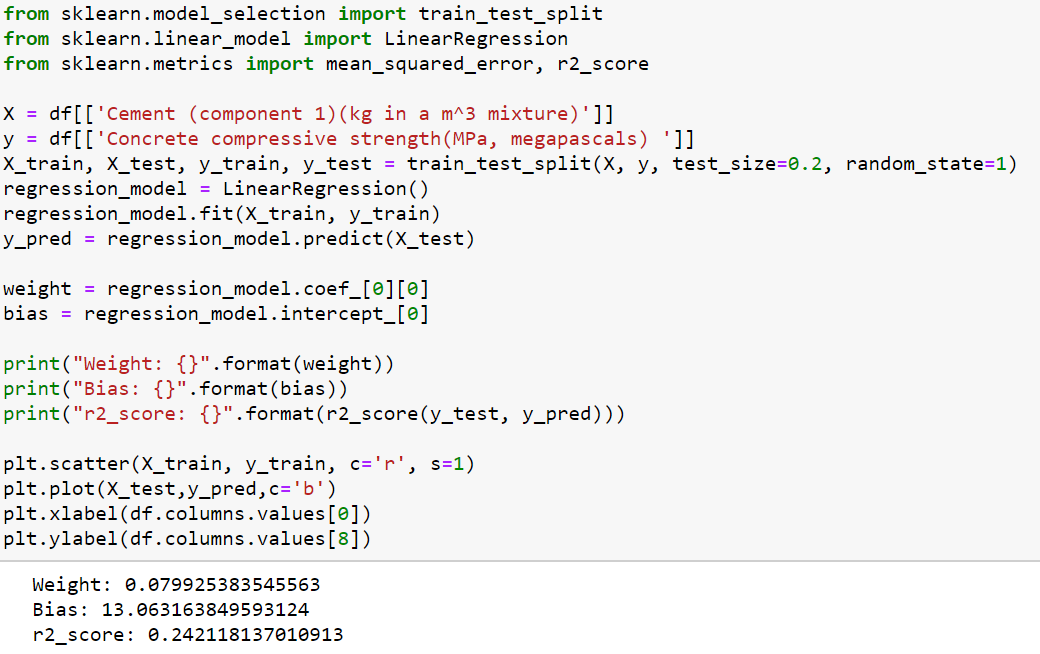


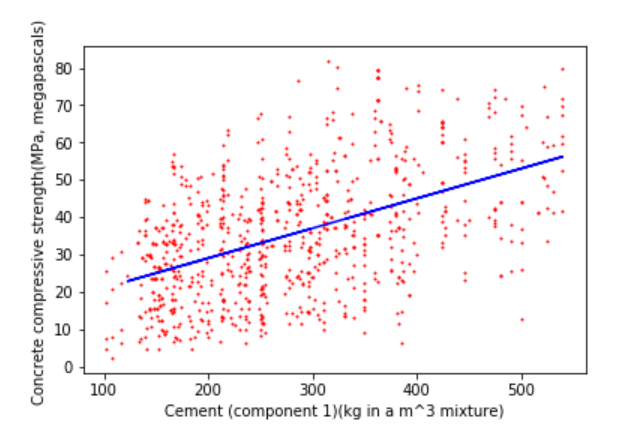


1. Code, graph and attributes for problem 1

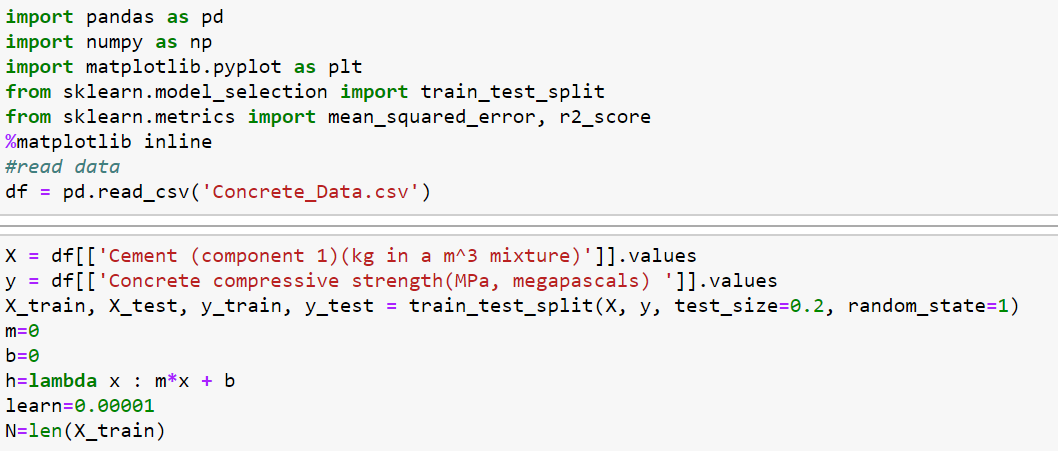
Wight 和 bias初始值設為0的是因為由第一題結果的圖看出最適直線是一條斜率為正，並且bias>0的線。這樣從0開始，每次增加一些，即可較快找到答案。

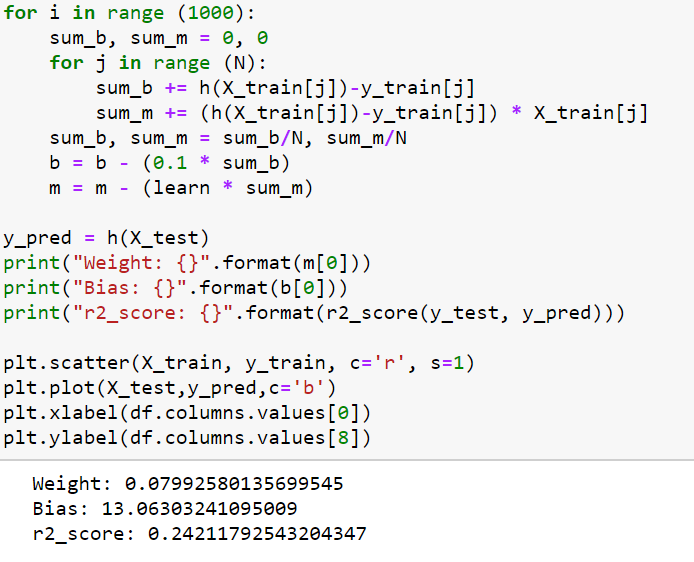


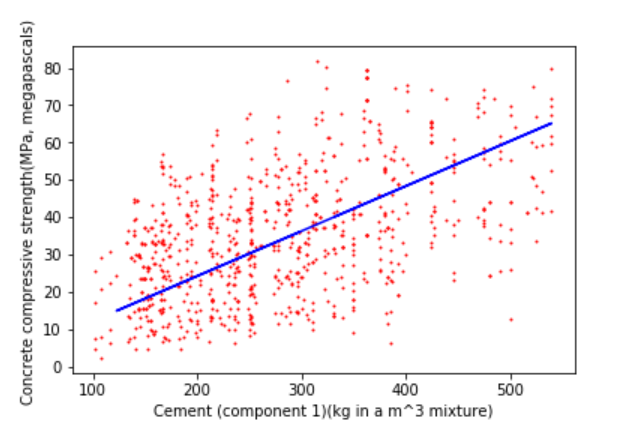




1. Code, graph and attributes for problem 2



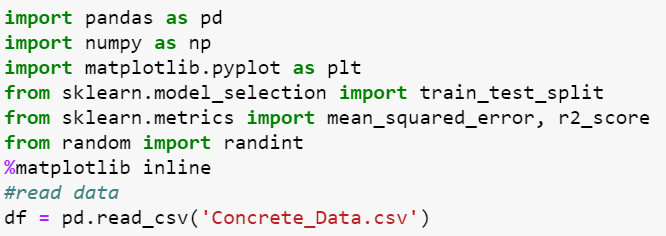


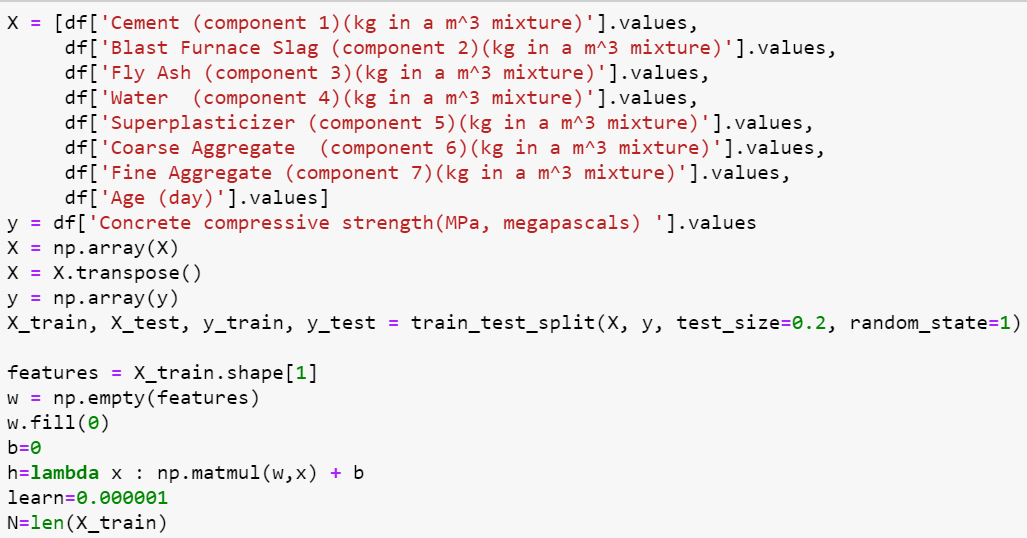


1. Compare problem 1 and problem 2

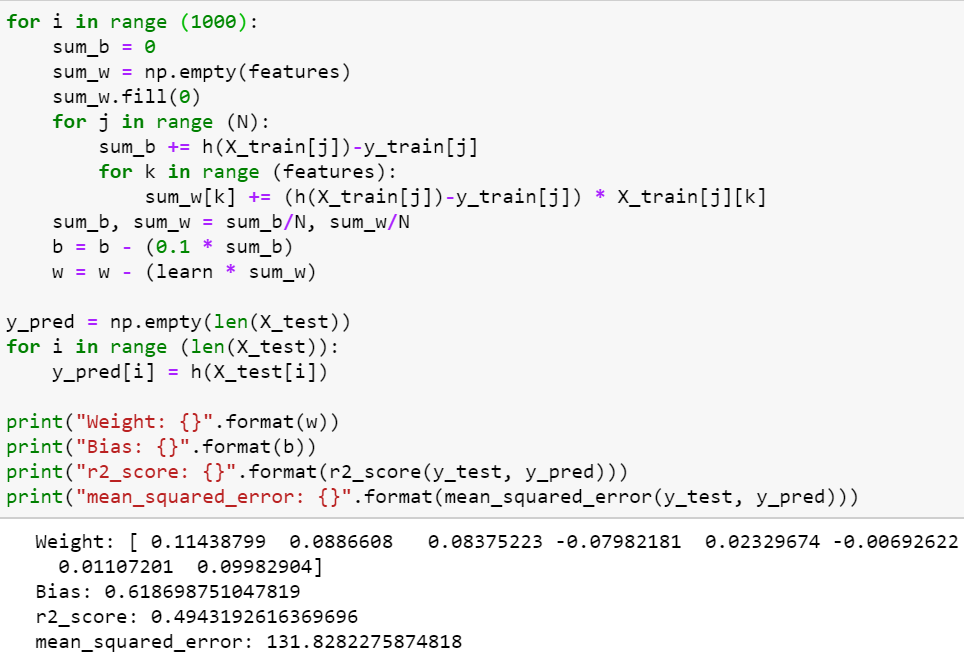
第二題一開始把weight和bias的learning rate設0.00001，原因是因為weight的sum太大，會無法收斂，可是r2\_score只能達到0.14，低於第一題0.1。後來發現這樣bias的每次的變動會很小，造成bias變動速度太慢，不滿足預期。於是把bias的learning rate改0.1後，發現結果則會跟第一題差不多r2\_score達到0.24，這個發現讓我了解到針對不同數值大小給予不同的learning rate能幫助其增加正確率及收斂。

6. The code,MSE,and the r2\_score for problem3

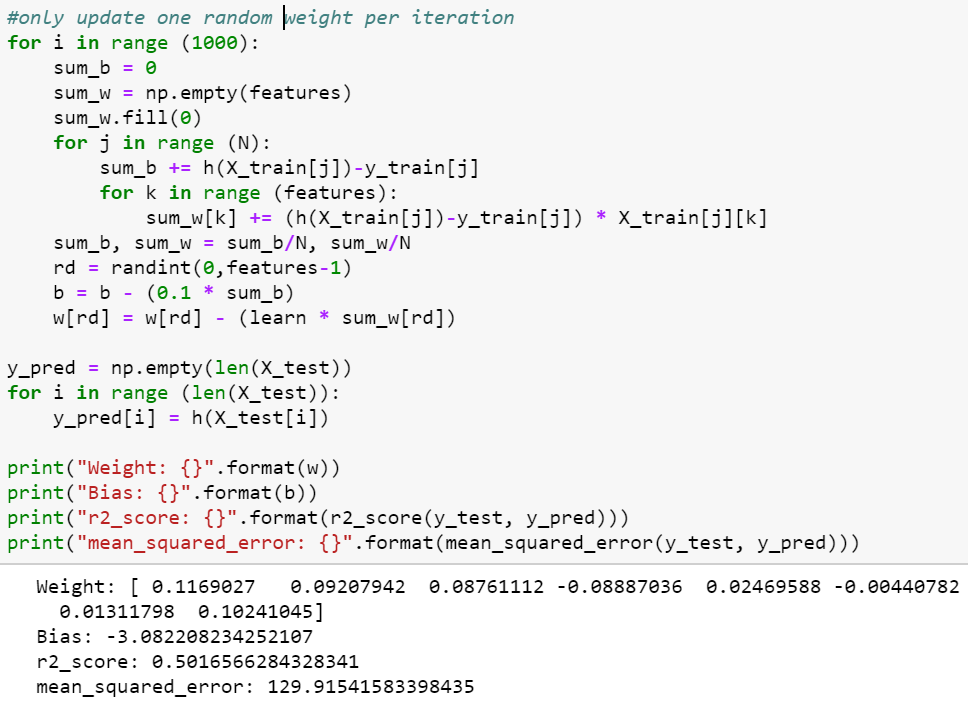




Each iteration updates w:



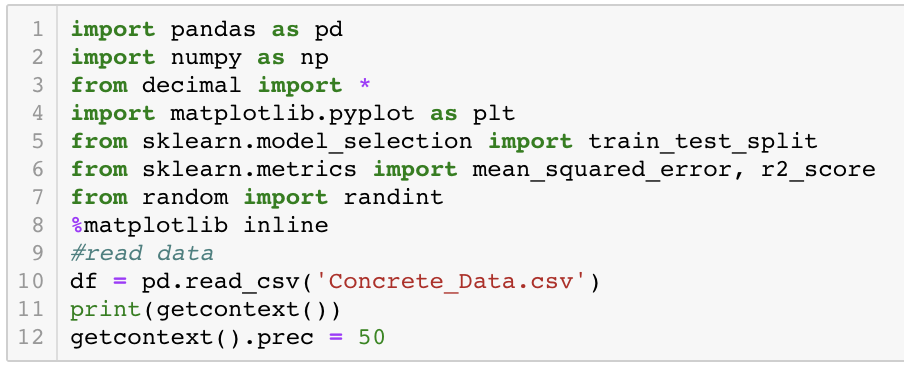
Each iteration only updates w*j*:



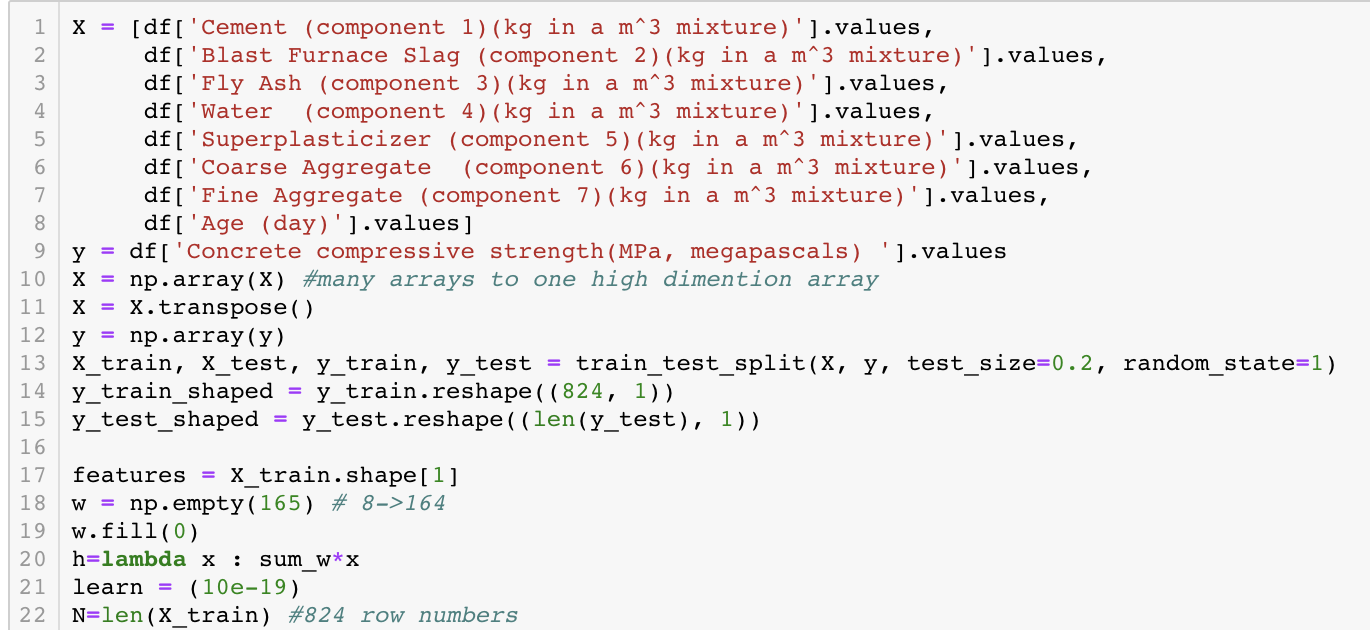
7. Compare the performance between two different update method

嘗試多次之後，發現兩者的performance其實差不多。一開始認為隨機更新參數的performance可以跟上全部參數更新的performance是因為iteration夠多，以至於已經達到需更新的量，因此我們試著調低iteration的次數，卻發現performance仍然接近，有時甚至更高，讓人不禁佩服想出這個作法的人，在weight很多的情況下，隨機更新參數可以降低運算量，同時又可以有不錯的performance，是個很不錯的參數更新方法。

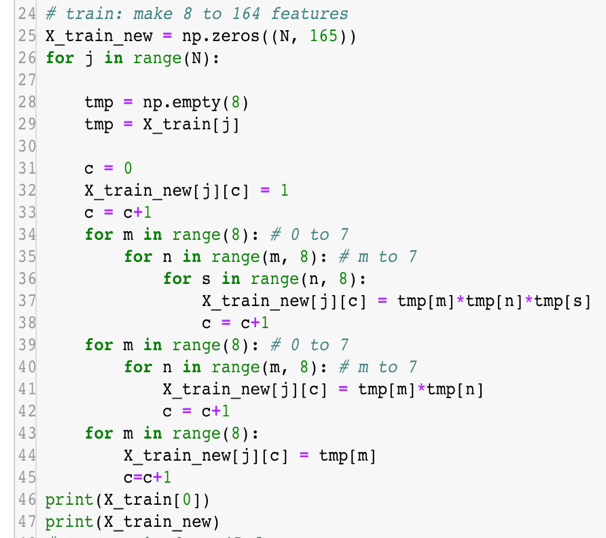
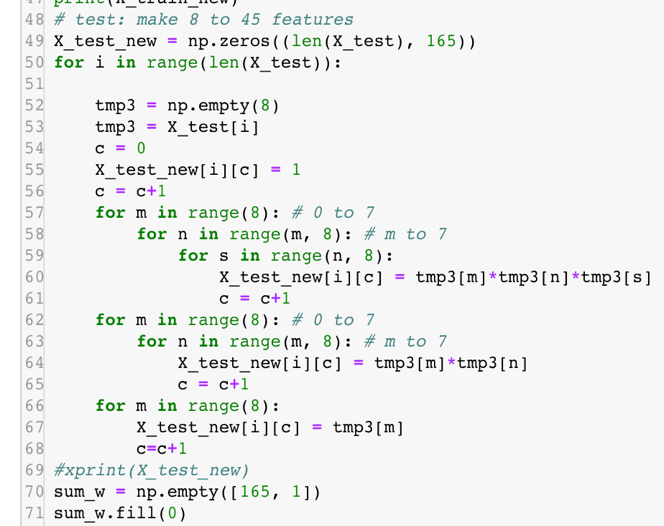
8. The code, MSE, and the r2\_score for problem 4



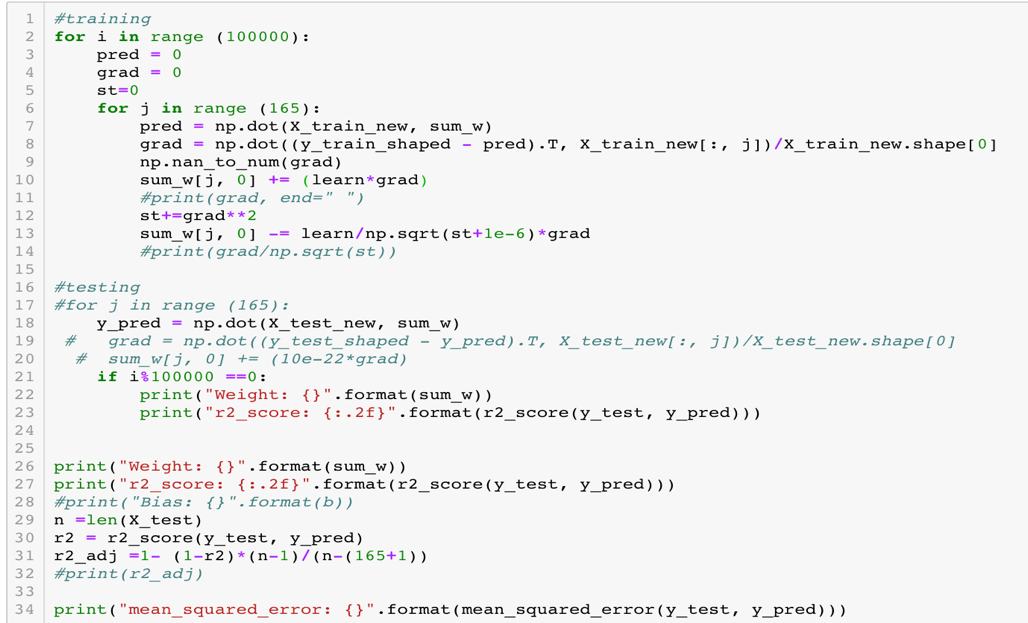
因為y\_train, y\_test都是(N, )所以都樣reshape變成(N, 1)



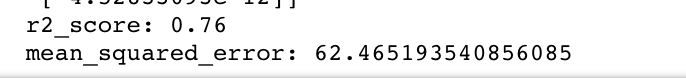
把train, test從原本的8 features 變成165 features(三次方)

把sum\_w\*X得到prediction, 再用此prediction算出gradient, 最後update sum\_w, 因為是165個feature, 所以總共做那個多次，然後跑十萬次iteration，後來用了adagrad optimizer加強，發現結果有比較好一點。



最後的MSE跟r2\_score



9. Answer the question

1. What is overfitting?

The hypothesis learned by machine learning is extremely close to the training data. That may let error gets bigger.

1. Stochastic gradient descent is also a kind of gradient descent, what is the benefit of using SGD?
2. When the sample is big, it can compute the data more easy.
3. It can be close to the optimal solution in a relatively short time.

3.Why the different initial value to GD model may cause different result?

GD model is that find local minima based on the initial value. So different initial value may have different local minima, the cause different result.

4.What is the bad learning rate? What problem will happen if we use it?

The learning rate is not in the proper range of speed.

If the learning rate is too big, the result may exceed the best solution.

If the learning rate is too small, the descending rate may be small, and cost tons of time.

5. After finishing this homework, what have you learned, what problems you encountered, and how the problems were solved?

We have learned that adjusting the learning rate is really important. The lower the value is, the slower we will travel along the slope, although it might be a good idea since we don’t want to miss the local minima, it could really take a lot of time to update and converge. So the solution is to make learning rates bigger, but sometimes we may encounter overflow problems, so we need to be really careful to adjust learning rates.

10. Bonus

Hope we can finish this!!!