cccc

1 *Here we are given w and y as vectors, so we can consider the vector dot product or multiplication whenever required.*

$$L(w) = ||w - w_0|\,|_2^2$$

$$L(w) = \sum_{i=1}^{d} ||w - w_0|\,|_2^2$$

a *Now to find the gradient of L(w) which is just the partial derivative of L(w) w.r.t w_j,we get the following: (i)*

$$\nabla L(w) = \frac{\partial L(w)}{\partial w_j} = \frac{\partial \sum_{i=1}^{d} ||w - w_0|\,|_2^2}{\partial w_j}$$

$$= \frac{\partial \sum_{i=1}^{d} (\sqrt{(w - w_0)^2})^2}{\partial w_j}$$

$$\nabla L(w) = 2(w - w_0)$$

(ii)

$$\nabla L(w) = 0$$

$$= 2(w - w_0) = 0$$

$$w = w_0$$

b

$$L(w) = ||Aw - b|\,|_2^2$$

$$L(w) = \sum_{i=1}^{d} ||Aw - b|\,|_2^2$$

(i)

$$\nabla L(w) = \frac{\partial L(w)}{\partial w_j} = \frac{\partial \sum_{i=1}^{d} ||Aw - b|\,|_2^2}{\partial w_j}$$

$$= \frac{\partial \sum_{i=1}^{d} (\sqrt{(Aw - b)^2})^2}{\partial w_j}$$

$$\nabla L(w) = 2(Aw - b)A^T$$

(ii)

$$\nabla L(w) = 0$$

$$= 2(Aw - b)A^T = 0$$

$$w = bA(AA^T)^{-1}$$

c

$$L(w) = ||Xw - y|\,|_2^2 + \gamma\,||w|\,|_2^2$$

$$L(w) = \sum_{i=1}^{d} ||Xw - y|\,|_2^2 + \sum_{i=1}^{d} ||w|\,|_2^2$$

(i)

$$\nabla L(w) = \frac{\partial L(w)}{\partial w_j} = \frac{\partial \sum_{i=1}^{d} ||Xw - y|\,|_2^2 + \sum_{i=1}^{d} ||w|\,|_2^2}{\partial w_j}$$

$$= \frac{\partial \sum_{i=1}^{d} (\sqrt{(Xw - y)^2})^2 + (\sqrt{(w)^2})^2}{\partial w_j}$$

$$\nabla L(w) = 2(Xw - y)X^T + 2\gamma(w)$$

$$w = yX^T(XX^T + \gamma)^{-1}$$

(ii)

$$\nabla L(w) = 0$$

$$= 2(Xw - y)X^T + 2\gamma(w) = 0$$

$$= 2(XX^T w\gamma(w) - yX^t) = 0$$

$$w = yX^T(XX^T + \gamma)^{-1}$$

2 *a. Pseudo code for multi variate linear regression*

(1).Set the initial point w_0 to an arbitrary value in vector space R^n.
(2).Until a stopping criterion(for now we can take it up to iteration of (K= 0,1,2,.....T)) is met, update by:

$$w_{k+1} = w_k + \frac{\alpha_k}{S_k} \sum_{i=1}^{S} (y_i - < x_i, w_i >)$$

here $\frac{\alpha_k}{S_k}$ can be considered as an average of functions ,where S_k, is the number of mini batches.The selection of S_k is uniformly random. For a special case we consider $S_k = 1$ which is 1 mini-batch we get the updated weights as :

$$w_{k+1} = w_k + \alpha_k(y_1 - < x_1, w_1 >)$$

b *Here we often consider that diminishing learning rate such that $_k = \frac{1}{k}$. Now if we consider the that learning rate is constant of cyclic update then we have $\alpha_k = k$,so the updated weights for m rows will be:*

$$w_{k+m} = w_k - k \sum_{i=1}^{m} \nabla L(w_{k+i-1})$$

Meanwhile, full gradient with step size mk would give:

$$w_{k+1} = w_k - k \sum_{i=1}^{m} \nabla L(w_k)$$

The difference here:

$$k\sum_{i=1} m\nabla L(w_{k+i-1}) - \nabla L(w_k)$$

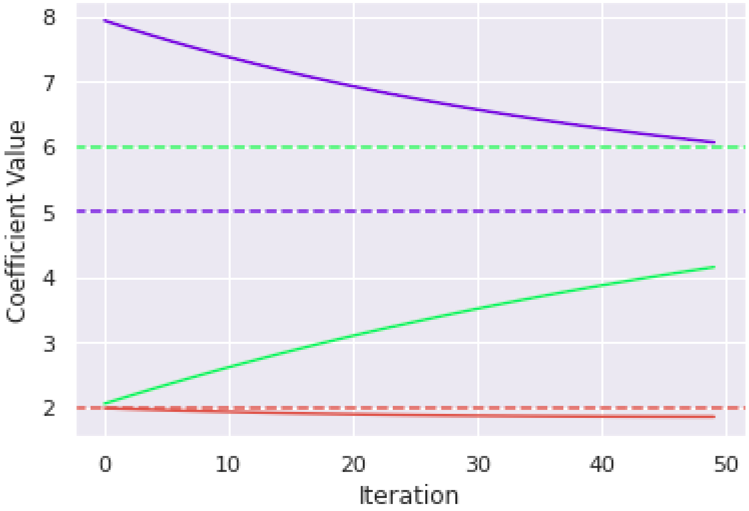
and if we hold t constant, this difference will not generally be going to zero.So we can say that we cannot optimize the convex curve and hence the algorithm will not converge.

c *It has been observed in practice that when using a larger batch there is a degradation in the quality of the model, as measured by its ability to generalize large-batch methods tend to converge to sharp minimizers of the training and testing functions—and as is well known, sharp minima lead to poorer generalization. n. In contrast, small-batch methods consistently converge to flat minimizers, and the experiments support a commonly held view that this is due to the inherent noise in the gradient estimation.So it is fair to say that we can use larger batch size if out training data set is large so that we can reach optimum point even with a large batch size and our convex curve is optimised. Using a smaller batch size ensures that there is a smoother transition and we reach the optimum point within the epoch size and get a regular curve, although there is a compromise with noise intrusion.*

Solution 3 :

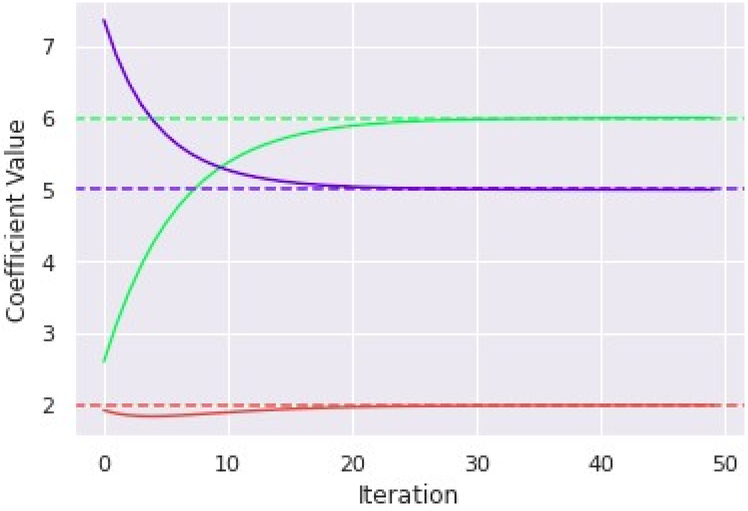
1. **For learning rate = 0.0002**

- **the plot of coefficient value vs. iteration :**



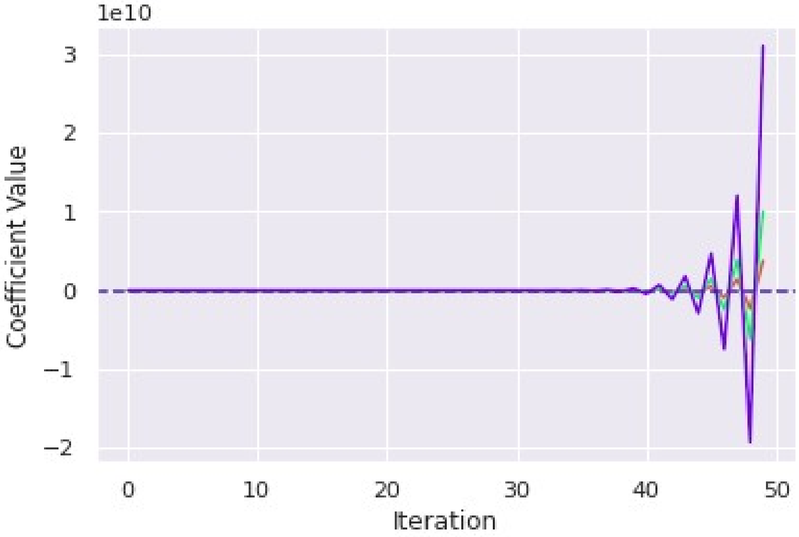
For learning rate = 0.002 :

- **the plot of coefficient value vs. iteration :**

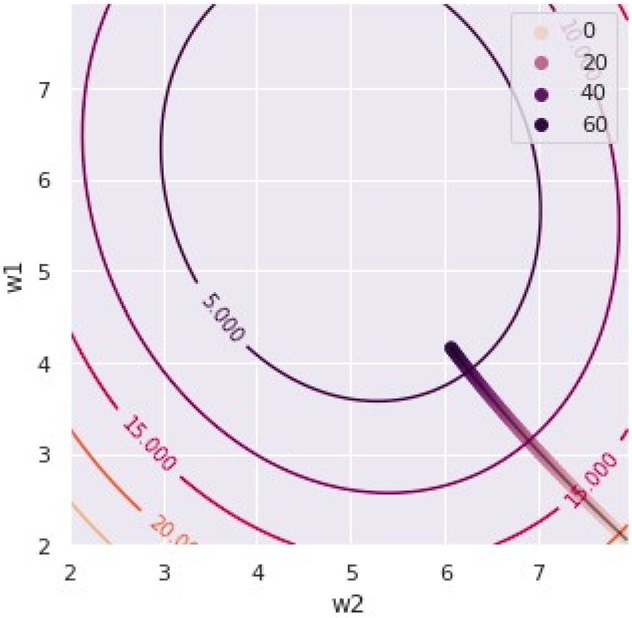


For learning rate = 0.02 :

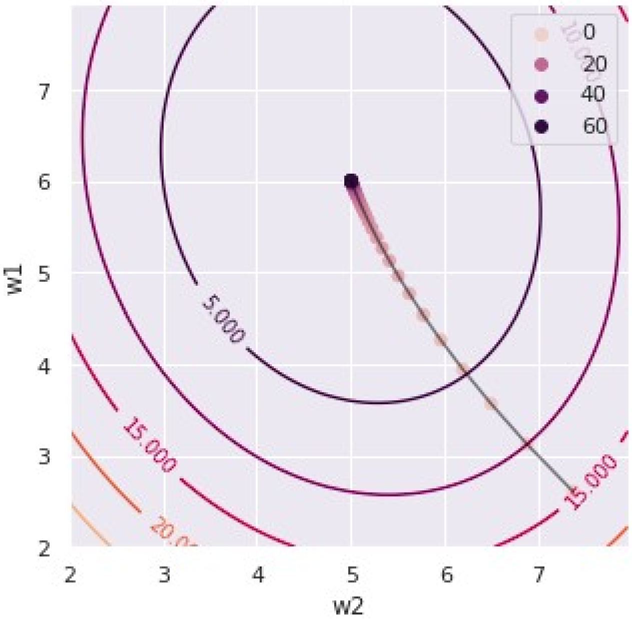
- **the plot of coefficient value vs. iteration :**



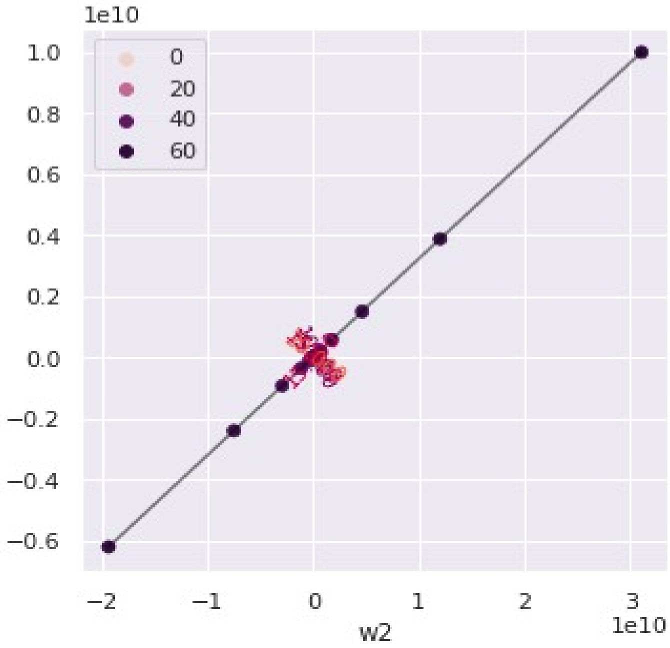
- **plot of the descent path on the MSE contour for learning rate = 0.0002 :**



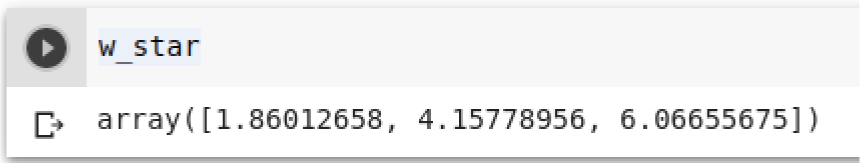
- **plot of the descent path on the MSE contour for learning rate = 0.002:**



- **plot of the descent path on the MSE contour for learning rate = 0.02:**



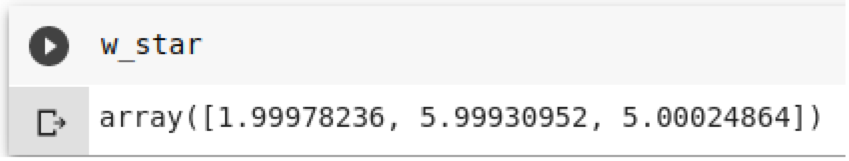
- **the estimate of w after 50 iterations for learning rate = 0.0002:**



```
w_star
array([1.86012658, 4.15778956, 6.06655675])
```

For the learning rate set to 0.0002 , the model is predicting the values correctly but since the learning rate is low it is getting to the optimal point rather slowly. The model seems to predict the w estimate value as decently as the values were supposed to be [2 , 6 , 5]. This can be improved by increasing the learning rate or by increasing the number of iterations.

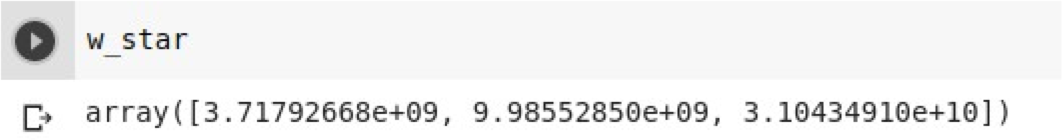
- **the estimate of w after 50 iterations for lr = 0.002:**



```
w_star
array([1.99978236, 5.99930952, 5.00024864])
```

For a learning rate of 0.002 , the estimate of w is very close to the original value of 2 , 6 , 5 .

- **the estimate of w after 50 iterations for lr = 0.02 :**



```
w_star
array([3.71792668e+09, 9.98552850e+09, 3.10434910e+10])
```

For a learning rate of 0.02 , the estimate of w is way off the original value of [2 , 6 , 5].

- **Behavior of convergence for learning rate = 0.0002 :**

The model is performing decently as per the general trend but since the learning rate is small , the convergence to the optimal point will require more iterations greater than 50. If we increase the step size then the optimal point can be reached but within 50 iterations the model is not at the optimal point..

- **Behavior of convergence for lr= 0.002 :**

This model performs well for this particular learning rate and the step size. The prediction converges and also show the optimum value within the 50 iterations.

- **Behavior of convergence lr = 0.02:**

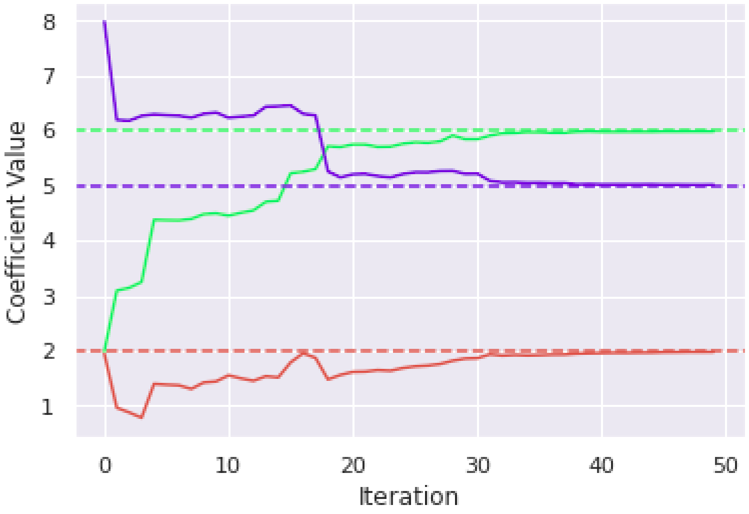
This model performs significantly bad for this particular learning rate and the step size. The prediction diverges from the optimal point and the optimal point cannot be attained within 50 iterations .

3.

(b). **Stochastic gradient descent:**

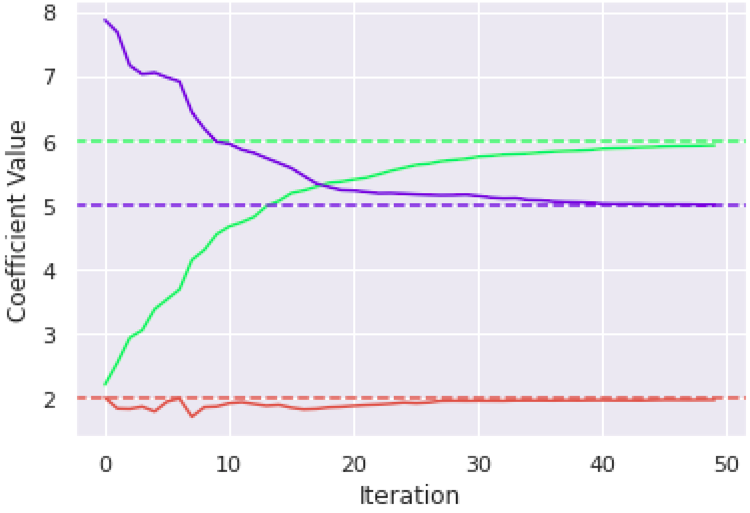
1. **Learning rate = 0.1 and n=1 :**

- **plot of coefficient value vs. iteration:**



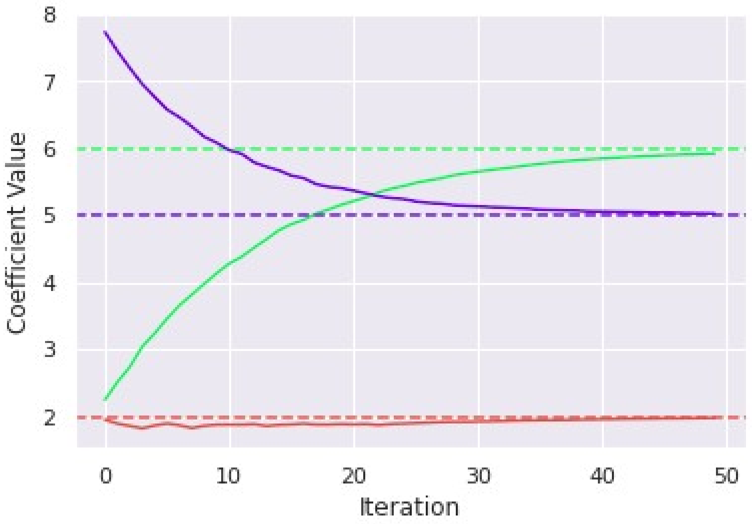
Learning rate = 0.01 and n=10 :

- **plot of coefficient value vs. iteration:**

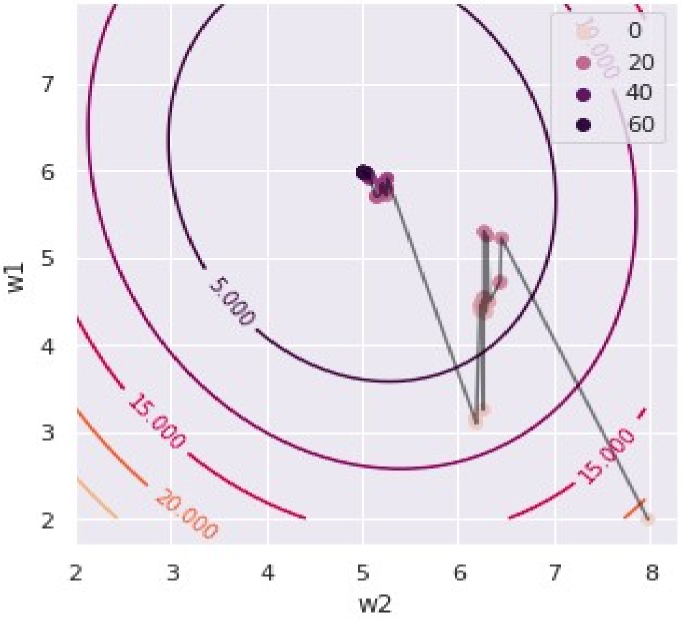


Learning rate = 0.001 and n=100 :

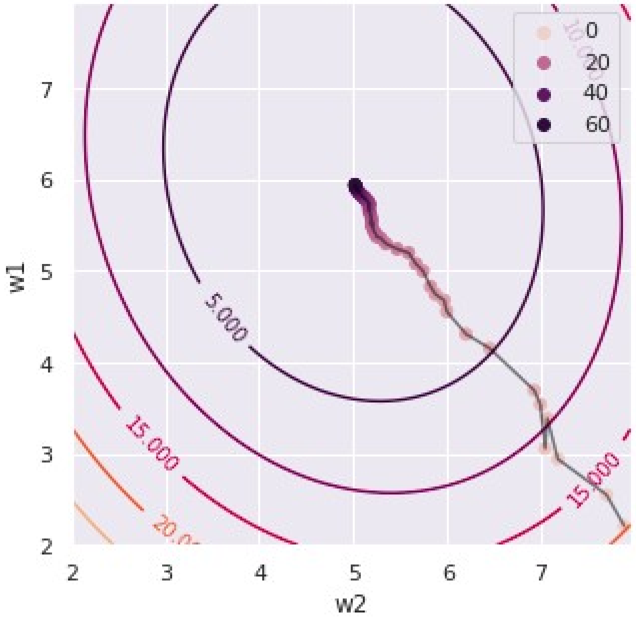
- **plot of coefficient value vs. iteration:**



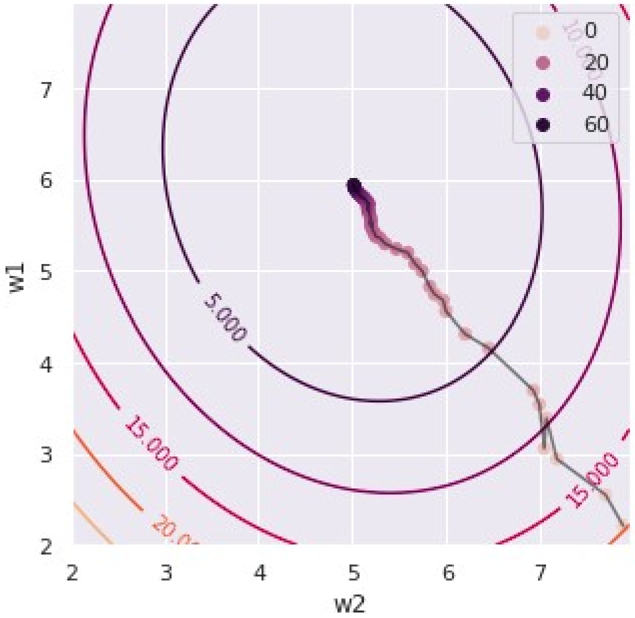
- **plot of the descent path on the MSE contour for Learning rate = 0.1 and n=1:**



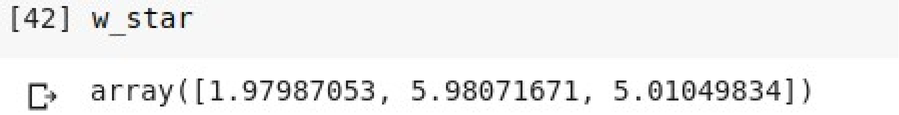
- **plot of the descent path on the MSE contour Learning rate = 0.01 and n=10:**



- **plot of the descent path on the MSE contour Learning rate = 0.001 and n=100:**



- **Comment for Learning rate = 0.1 and n=1 :**

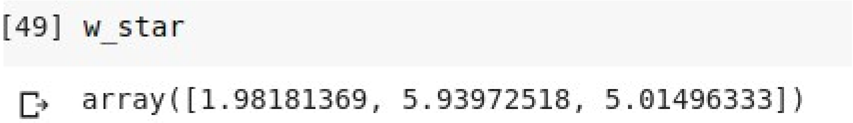


```
[42] w_star

array([1.97987053, 5.98071671, 5.01049834])
```

As we can see that these parameter for learning rate and sample gives us accurate results for w estimation but the convergence to the optimal math is not very smooth , this shows that still the model can be improved. The random behavior of the convergence is because of the weights might no be the best selected or the appropriate learning rate and samples are not selected.

- **Comment Learning rate = 0.01 and n=10:**

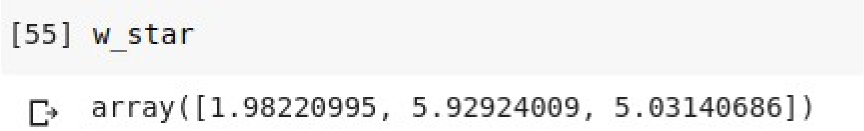


```
[49] w_star

array([1.98181369, 5.93972518, 5.01496333])
```

As we can see that these parameter for learning rate and sample gives us accurate results for w estimation and the convergence to the optimal path is smoother compared to the previous model parameters, this shows that the model can be might be closer to its optimal point but still there is a room for improvement .

- **Comment Learning rate = 0.001 and n=100:**



```
[55] w_star

array([1.98220995, 5.92924009, 5.03140686])
```

As we can see that these parameter for learning rate and sample gives us accurate results for w estimation and the convergence to the optimal path is smoothest than the other two, this shows that the model can be might be closest to its optimal point from the given parameters.