1. **Sklearn.LinearRegression**

Linear Regression of sklearn after 0.039s, Score : 0.6718788122915272

1. **GD**

W\_init = np.zeros chạy nhanh hơn np.ones vì w có cả thành phần âm, dương.

Fix\_step\_size <1.02 thì chạy ok. Nhỏ quá thì chạy chậm. Từ 1.03 trở ra chạy ra gradient = NaN

Hội tụ:

* W\_init = np.zeros
* Fix\_step\_size = 1
* Tol = 1e-2
* Max\_iter = 100.000

=> hội tụ sau 9.101564884185791 s, after 40650 steps; Score : 0.6449954864989078 => chậm gấp 233 lần sklearn, score thấp hơn. Xem lại vì sao ko hội tụ sau 1/epsilon bước như lý thuyết

Nếu max\_iter = 10.000, chưa hội tụ nhưng Score : 0.6635298053413301

Max\_iter = 1.000, chưa hội tụ, Score : 0.6211606493374768

1. **BGD**

Alpha = beta = 0.5

Nếu khởi tạo step\_size =1 thì chạy như GD, không có cập nhật stepsize

Nếu khở tạo step\_size = 2,3,… thì chạy chậm hơn GD, có cập nhật stepsize

Max\_iter = 100.000 thì thuật toán đều hội tụ dưới mức max này nên đều có R2 = 67% (tốt):

Stepsize = 1 => BGD complete in 28.480236530303955 s, after 40700 steps, Score : 0.6734595287018551

Stepsize = 2 => BGD complete in 36.29433989524841 s, after 33900 steps, Score : 0.6734595283681817

Stepsize = 3 => BGD complete in 54.44734334945679 s, after 32800 steps, Score : 0.6734595279759363

Stepsize = 4 => BGD complete in 17.371493577957153 s, after 14900 steps, Score : 0.6734595281156686 => chậm gấp 445 lần sklearn, score tốt hơn

Stepsize = 5 => BGD complete in 45.47954821586609 s, after 20900 steps, Score : 0.6734595272478054

Stepsize = 6 => BGD complete in 60.74782371520996 s, after 32800 steps, Score : 0.6734595279759363

>> BGD không nhanh hơn GD nhưng hội tụ sau ít bước cập nhật w và cho R2 tốt hơn

1. **Newton**

* W\_init = np.zeros
* Fix\_step\_size = > ko phụ thuộc
* Tol = 1e-4
* Max\_iter = 100.000
* Checkafter = 1 => vì số bước ít

Newton complete in 0.022215843200683594 s, after 10 steps => nhanh hơn sklearn, time = 0.57 lần sklearn, score tốt hơn

>> pure Newton có stepsize fix = 1, nếu với 1 tol thì chỉ có 1 kết quả time, steps

1. **Newton backtracking**

* W\_init = np.zeros
* Fix\_step\_size = …
* Tol = 1e-4
* Max\_iter = 100.000
* Checkafter = 1 => vì số bước ít

Stepsize = 5 => Backtracking Newton complete in 0.07260465621948242 s, after 20 steps, Score : 0.6734595252441942

Stepsize = 4 => Backtracking Newton complete in 0.05302143096923828 s, after 15 steps, Score : 0.6734595252362119

Stepsize = 3 => Backtracking Newton complete in 0.19379544258117676 s, after 26 steps, Score : 0.6734595252351516

Stepsize = 2 => Backtracking Newton complete in 0.05106830596923828 s, after 15 steps, Score : 0.6734595252362119

Stepsize = 1 => Backtracking Newton complete in 0.04331612586975098 s, after 15 steps, Score : 0.673459525237138 => chậm hơn sklearn 1.1 lần, score tốt hơn

Stepsize = 0.5 => Backtracking Newton complete in 0.0681610107421875 s, after 32 steps, Score : 0.6734595252348237

Stepsize = 0.1 => Backtracking Newton complete in 0.41141319274902344 s, after 215 steps, Score : 0.673459525235789