資料科學導論 exetcise3 大氣4A 黃展皇 106601015

程式碼如附件code\_106601015.py

Requestment：sys、sklearn、numpy、pandas

學生作業實作的部分是gradient\_descent方法的code內容，不過也另外做了Close form的實作，以便一起比較。

1. R^2值結果討論：

Gradient\_descent：

Linear train R^2: 0.922953

Linear test R^2: 0.902184

Close form：

Linear train R^2: 0.925422

Linear test R^2: 0.905320

可以看到R^2值無論訓練/測試在此案例中都有約0.92/0.90，兩種方法在最後結果的部分相去不遠。

1. (長度為8的矩陣)值討論

Gradient \_descent的：

[[ -3.39659641][-45.01500853][ 33.06177329][ 33.03745135]

[ 17.98336829][ -0.32619675][ 7.50926399][ 0.84975238]]

Close form的：

[[ -20.509636 ][-188.53919107][ 106.81649471][ 123.10270717]

[ 13.66996781][ -0.33374323][ 7.47571038][ 0.82014191]]

可以看到兩種方法算出來的值在第0、1、2、3項差距很大，但在第4、5、6、7項差距不大，代表在後四項兩方法得出的結論一致，但在前四項則有不同的詮釋，再看回R^2值可以發現兩者的詮釋都是合理的，沒有哪種方法準度較高的討論空間。

1. 運算複雜度討論

Close form公式：，只需要X以及y即可推導出，程式實現設計上也非常簡單且快速，但相對空間複雜度也高(如ppt所演示，若資料集很大則空間需求會快速膨脹)，只能針對小型資料集做處理；反之Gradient \_descent是unconstrained method適用在所有情況，但相對的時間上因為J始終降不到預設的eps=1e-4，又另外的終止條件即iters = 100000使得運算時間相當長。但根據對於J的觀察，其實J在訓練前期即達到穩定的局部低點，因此可以考慮加入第三個終止條件：若下次訓練的進步低於一預設低值則可終止，以防已經在局部低點但J又高於eps而一直訓練的狀況，減少不必要的訓練時間。