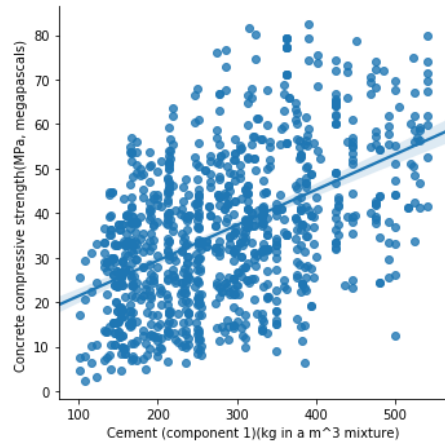
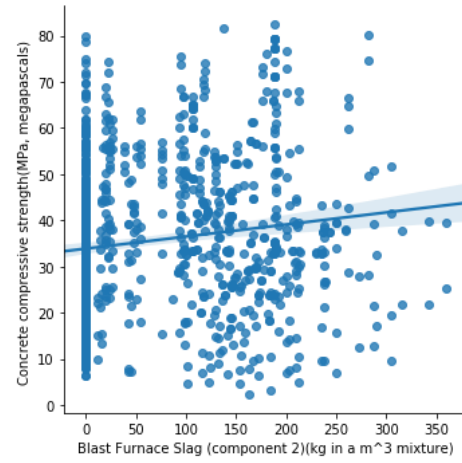


# Machine Learning Homework 3 Report

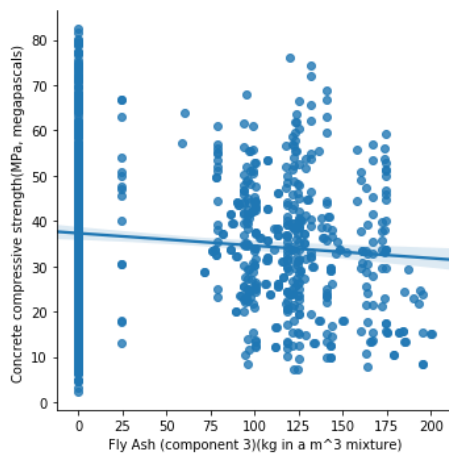
## 一、Visualization



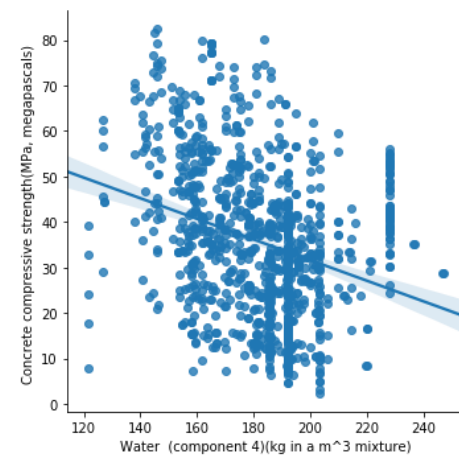
(a)



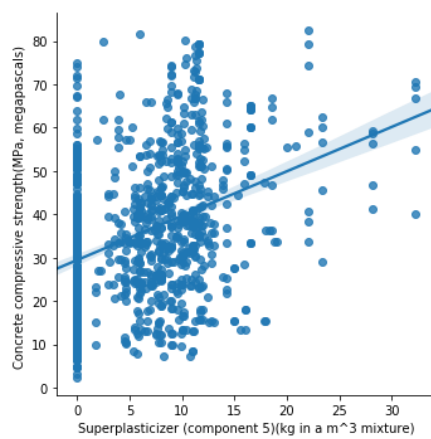
(b)



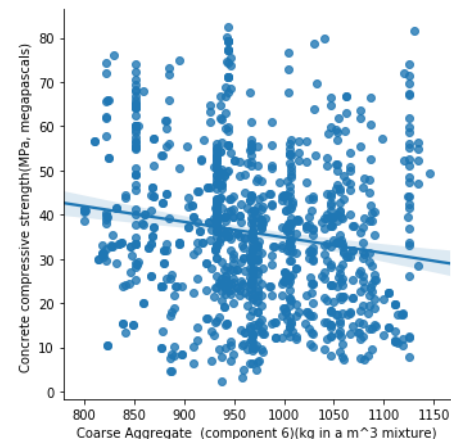
(c)



(d)



(e)



(f)

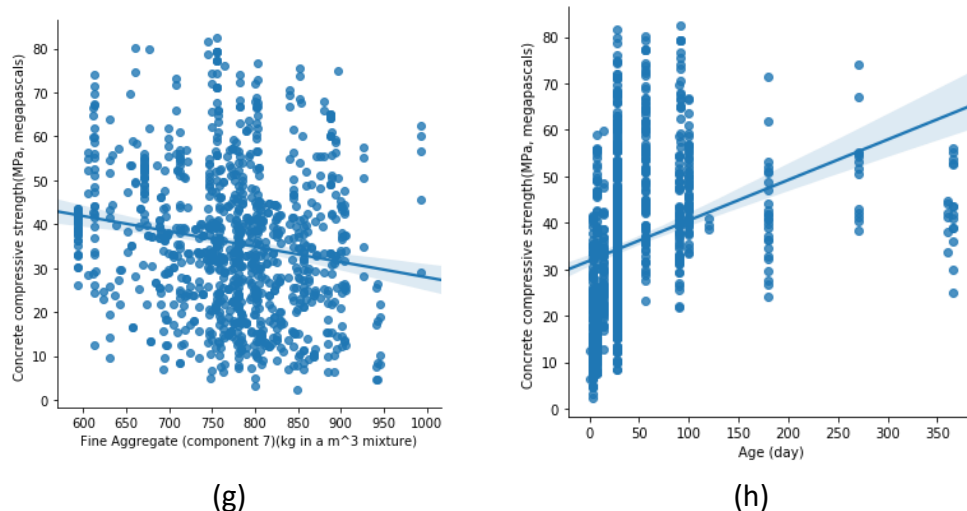


Fig.1 The scatter plot of concrete compressive strength and other features

根據 Fig.1，Cement 此特徵與 Concrete compressive strength 有較明顯的正相關，所以在接下來的單變數線性回歸中，我們使用 Cement 此特徵作為變數預測結果。

## 二、Preprocessing

環境設置方面，我們這組使用 python 進行實作、使用 seaborn 進行視覺化。處理資料方面，對整組資料進行了 normalization，以方便在選取參數出值時有一個較好的結果，式子如下。

$$\bullet \quad (x_i - \mu_x) / \max(x) - \min(x)$$

$x_i$  為第  $i$  個實例  $x$  特徵的值， $\mu_x$  為該特徵的平均。

除此之外，我們還加上了一個叫做 Bias 的特徵，並將值全部設成 1，以方便我們後面進行矩陣運算和偏微分的運算。

## 三、Model

這邊採用了三種 gradient descent 的算法，分別為 Adam、SGD、和每次僅對一個參數進行調整的 Naïve。然而，經過測試 Adam 的學習效果有點慢所以調整  $\alpha = 0.05$ ，其餘的優化方法皆設定 learning rate = 0.05。

## 四、Evaluation

所有模型採用兩種衡量方法 Mean Square Error 和  $R^2$ 。而 Loss function 則定為 MSE。

$$\bullet \quad MSE(X, y, \theta) = \frac{1}{m} \sum_{i=1}^m (\theta^T \cdot x^i - y^i)^2$$

$$\bullet \quad \nabla_{\theta} MSE(X, y, \theta) = \frac{2}{m} X^T \cdot (X \cdot \theta - y)$$

$X$  為特徵所組成的矩陣， $y$  為目標所形成的向量，而 $\theta$ 則是由權重和偏差值所組成的向量。

## 五、Result

Table 1. The results of different regression model and GD method

Model	Train MSE	Train R <sup>2</sup>	Test MSE	Test R <sup>2</sup>
SKlearn Linear Regression	0.321	NaN	0.034	NaN
Self Linear Regression	0.321	NaN	0.034	NaN
Multivariable(Adam)	0.016	0.616	0.016	0.60
Multivariable(SGD)	0.016	0.623	0.02	0.52
Multivariable(Naive)	0.026	0.383	0.025	0.41
Polynomial(Degree=4)	0.003	0.929	0.006	0.845

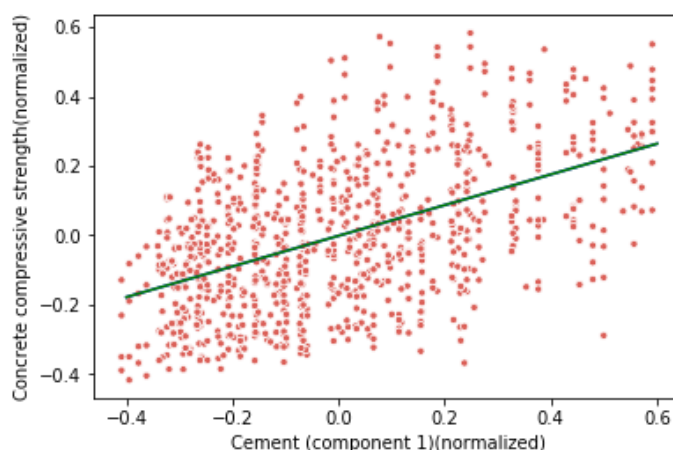
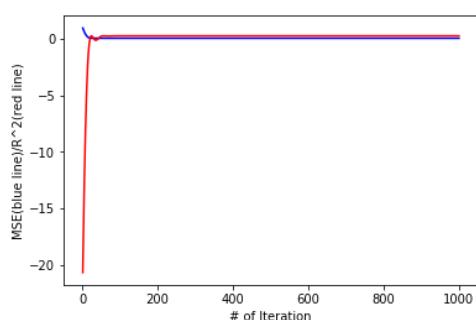
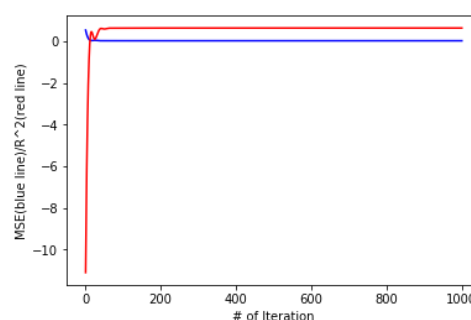


Fig.2 The Regression Line of SKlearn model(blue) and self linear regression model(green)

在 Fig.2 中可以發現，經由 SKlearn 所得到的迴歸直線  $y = -0.178291952 + 0.43220778x$  和經由我們使用 GD 所得到的直線是完全相同的。



(a)Self Linear Regression



(b)Multivariable(Adam)

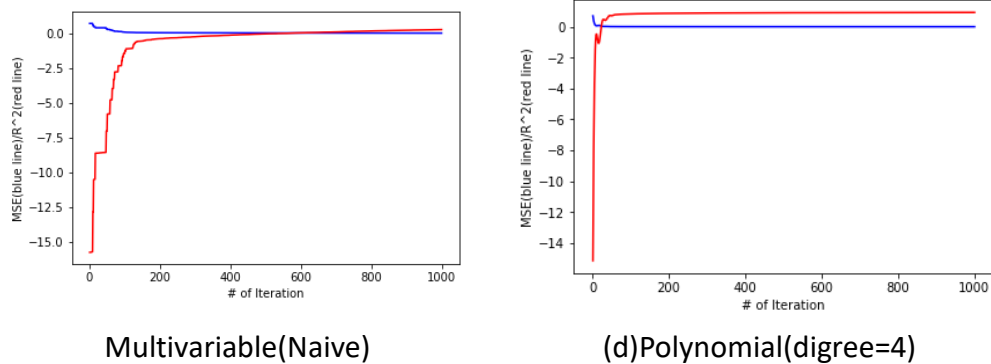


Fig.3 The changing loss after iteration in training process

根據 Table1.同樣是使用全部特徵的 Multivariable linear regression model，Adam 是表現最好的其次是 SGD 最末尾則是 Naïve。個人認為會造成這樣的表現差異是在於考慮的資料量不同 Adam 考慮了所有的實例以及所有的梯度，SGD 則是只考慮了單一筆資料和所有的梯度，Naïve 則是考慮了所有的實例以及單一的偏微分。考慮數量較少的資訊可以加快運算速度，但是可能收斂的效果就變差了。

## 六、Question Answering

1. Overfitting 是指 model 去過度擬合訓練資料集導致喪失了 generalize 的能力進而使得在測試資料集中有很差的表現。
2. 較快的運算，因為只考慮單一筆資料，所以適合使用在大型的資料集上也可達到不錯的成果。
3. 在 MSE 為 loss function 的情況下，Error surface 是一個 convex 的表面，所以不會有 local minima 或者是 saddle point 的問題，然而若使用不同的 loss function 例如：cross entropy，初值就有可能影響收斂快慢或者是否卡在 local minima 或 saddle point 等地方了。
4. Learning Rate 過大會造成無法收斂或者是跳過最小值，Learning Rate 過小則會造成收斂過慢，太花時間。
5. 理解 SGD 的用法並且學會用 numpy 進行矩陣運算。

## 七、Bonus

其實我們在 Polynomial regression model degree 為 4 時，有幾次的  $R^2$  有大於 0.87。(可以看附錄)

## 附錄

```
-----IN SKLEARN LINEAR REGRESSION-----
Coefficient of Regression line: [[0.41780096]]
Interception of Regression line: [-0.00037839]
train_MSE: 0.03212726019226542
test_R^2: 0.2892469291159464
test_MSE: 0.03430513669795984

-----IN SELF LINEAR REGRESSION-----
Train MSE: 0.03212726019226542
Train R^2: 0.23539696472914595
Coefficient of Regression line: [-0.00037839]
Interception of Regression line: [0.41780096]
test_MSE: 0.03430513669795984

-----IN MULTIVARIABLE LINEAR REGRESSION-----
Gradient Descent Method : Adam
Train MSE: 0.01662815595078646
Train R^2: 0.6169272115933898
Coefficient of Regression hyperplane: [ 4.40208875e-04  6.63392596e-01  4.81771954e-01  2.33467305e-01
-2.22473649e-01  1.16504617e-01  8.71240311e-02  1.11678335e-01]
Interception of Regression hyperplane: [0.53672318]
test_R^2: 0.6071642965725064
test_MSE: 0.016747673069627014

-----IN MULTIVARIABLE LINEAR REGRESSION-----
Gradient Descent Method : Naive
Coefficient of Regression hyperplane: [0.00278182 0.44849857 0.30940353 0.24366727 0.42757362 0.71815595
0.12369372 0.27423456]
Interception of Regression hyperplane: [0.70947756]
test_R^2: 0.2788897642985575
test_MSE: 0.030742924762994443

-----IN POLYNOMIAL LINEAR REGRESSION-----
Gradient Descent Method : Adam
Train MSE: 0.0031378175523431792
Train R^2: 0.9285212081603595
test_R^2: 0.845332585650281
test_MSE: 0.006295170538662913
```

```
Anaconda Prompt
(base) C:\Users\呂明哲\Desktop
(base) C:\Users\呂明哲\Desktop>cd code_and_stuff
(base) C:\Users\呂明哲\Desktop\code_and_stuff>cd MLHW3
(base) C:\Users\呂明哲\Desktop\code_and_stuff\MLHW3>python Concrete.py
-----IN SKLEARN LINEAR REGRESSION-----
Coefficient of Regression line: [[0.44201043]]
Interception of Regression line: [-0.00148778]
train_MSE: 0.032539594834593545
test_R^2: 0.21172548942278047
test_MSE: 0.032606515165880735

-----IN SELF LINEAR REGRESSION-----
Train MSE: 0.032539594834593545
Train R^2: 0.2555532775822209
Coefficient of Regression line: [-0.00148778]
Interception of Regression line: [0.44201043]
test_MSE: 0.032606515165880735

-----IN MULTIVARIABLE LINEAR REGRESSION-----
Gradient Descent Method : Adam
Train MSE: 0.01689194335743054
Train R^2: 0.6158093659506206
Coefficient of Regression hyperplane: [-0.00215197 0.68269346 0.50573651 0.23615179 -0.16741175 0.16438163
0.1172664 0.13974888]
Interception of Regression hyperplane: [0.49984553]
test_R^2: 0.6077326045349931
test_MSE: 0.01592189915511655

-----IN MULTIVARIABLE LINEAR REGRESSION-----
Gradient Descent Method : Naive
Coefficient of Regression hyperplane: [-0.00324023 0.96091611 0.61182628 0.48313882 0.76791127 0.46602238
0.43420152 0.672714 ]
Interception of Regression hyperplane: [0.7119543]
test_R^2: 0.40684961134385655
test_MSE: 0.02407561979706719

-----IN POLYNOMIAL LINEAR REGRESSION-----
Gradient Descent Method : Adam
Train MSE: 0.0031339241373762898
Train R^2: 0.9254572305949191
test_R^2: 0.8727177997549085
test_MSE: 0.006091891469170832
```