Deep Learning Homework1

StudentID: 107062650

Name: 曾昱榮

Part1 (a):

我將 train.csv 檔所有資料存到 df 這個 dataframe，再使用 DataFrame.sample()的函式將 df 內 80%的資料存到 train_df，而剩下的 20%存到 test_df。

```
def data_preprocess():
    select_columns = ['school','sex','age','famsize','s
    df = pd.read_csv('train.csv')
    df = df[select_columns]
    df = pd.get_dummies(df)

    #divide into train and test dataframe
    train_df = df.sample(frac=0.8)
    test_df = df.drop(train_df.index)
    train_df = train_df.reset_index()
    test_df = test_df.reset_index()
    train_G3 = train_df[['G3']]
    test_G3 = test_df[['G3']]
    train_df = train_df.drop(columns=['index', 'G3'])
    test_df = test_df.drop(columns=['index', 'G3'])
```

Part1 (c):

Part1 (f):

-

Linear Regression with Pseudo-inverse

RMSE: 11.621016643974272

-

Linear Regression with regularization λ = 1.0

RMSE: 11.621107867670135

-

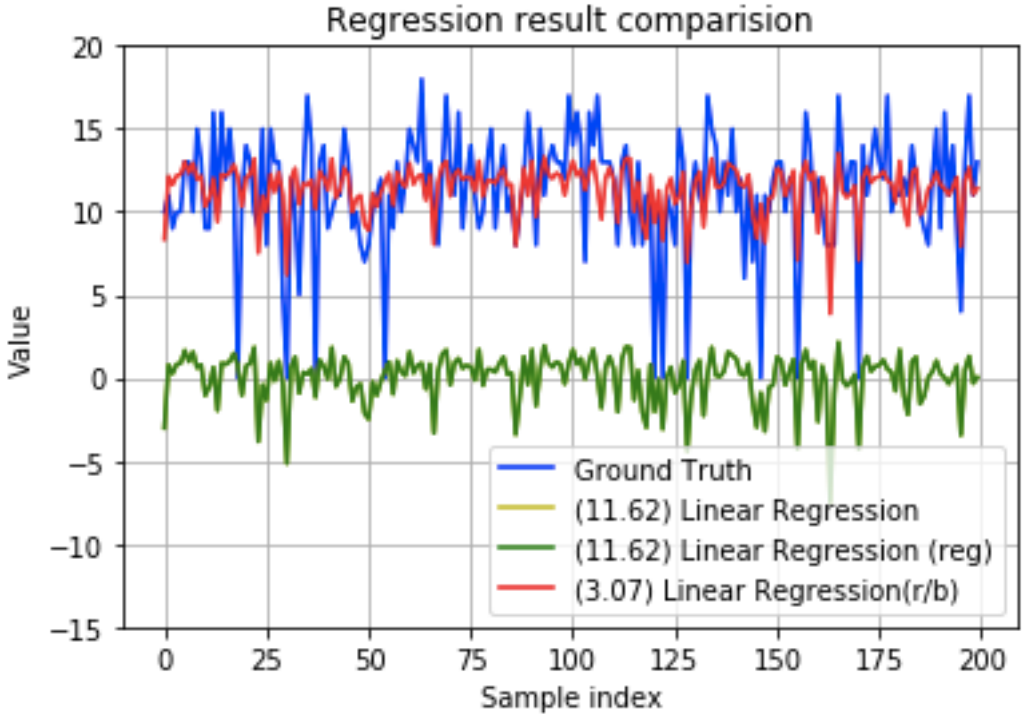
Linear Regression with regularization λ = 1.0 and bias term

RMSE: 3.0677115773854338

-

有加 bias term 的 model 預測的 G3 值會較接近 Ground Truth，從下圖可以看

出，bias term 可以上下調整 Value 的誤差值，所以有 bias 項的(d)(e) model 會有較接近 Ground Truth 的 G3 預測值。



Part2 (b):

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h_\theta(x^{(i)})) + (1-y^{(i)})log(1-h_\theta(x^{(i)})] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

上圖為 regularized logistic regression 的 cost function，下圖為經過上圖公式針對 θ_j偏微分更新 feature 的權重 theta 值的公式。α值是 learning rate，值越高訓練越快，但太高也有可能導致 theta 更新幅度過大而越過 minimum 的情形。而 gradient descent 還需設定好 iteration 的次數，次數只需達到 cost 的值收斂即可。

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} \qquad (\text{for } j = 0)$$

$$\theta_j := \theta_j - \alpha [\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j] \qquad (\text{for } j = 1, 2, \ldots, n)$$

}

Part2 (e):

	threshold = 0.1	threshold = 0.5	threshold = 0.9
Linear regression	Acc: 0.805 Precision: 0.804	Acc: 0.835 Precision: 0.843	Acc: 0.315 Precision: 0.848
Logistic regression	Acc: 0.815 Precision: 0.814	Acc: 0.84 Precision: 0.857	Acc: 0.28 Precision: 0.95

threshold 從 0.5 變 0.9，Accuracy 大幅下降的原因為，在原始 train.csv 的資料裡 G3 大於 10 分的為大多數，所以大多數的 G3 都被分類至 label 1，只有少部分被分到 label 0，所以預測當分類器在分類類別時，預測 1 會較容易預測成功而 Accuracy 也會較高。而當 threshold 變為 0.9 時，因預測為 1 的門檻變太高，而導致預測 1 的數量大幅下降，然而 G3 真實類別又大部分為 1，所以針對真實類別為 1 的數據預測準確度也大幅下降，所以 accuracy 變得相當低。