1.

(a) False. If weight $\hat{w}$ is sparse, then it means that some of the feature are not useful in the classification, there is no guarantee that the final model is sparse . A common way for seeking a sparse result is use the 1-norm, here, we used the L2-norm, so the result will not be sparse.

(b) False. If the data is linearly separable, then there exist a hyperplane to divide the data to two-part with different label. Even if we don't we the regularization term, the $logLoss_{train}(\hat{w}, \hat{b})$ joggling around the global optimum point and weights will not blow up but joggling too.

(c) True. We use the $R(w, b)$ to prevent the model overfit, overfit problem can simply describe as the train data error is going down(loss going dow) and the test data error is going up. Therefore, when the λ increase the $logLoss_{train}(\hat{w}, \hat{b})$ will increase..

(d) False. We use the $R(w, b)$ to prevent the model overfit, overfit problem can simply describe as the train data error is going down and the test data error is going up(loss going down up). Therefore, when the λ increase the $logLoss_{test}(\hat{w}, \hat{b})$ will decrease.

(e)

i. $$\underset{w,b}{argmax} \sum log \frac{1}{1 + exp(-y_n(w[1]x_n[1] + b))}$$

$$\underset{w,b}{argmax} \sum log \frac{1}{1 + exp(-y_n(w'[1]x_n'[1] + w'[2]x_n'[2] + b'))}$$

ii. The final model of the two dataset will be the same since they have the same value for the features, so the relation ship should be: $\hat{w}'[1] = \hat{w}'[2] = \frac{1}{2}\hat{w}[1], b' = b$

iii. No, they will not change. Because that doesn't change the fact the the final model will be the same, and the penalty given by the regularization will also be the same, so at the end they will still be the same.

2.

(a)

Linear regression weight update:

$$w \leftarrow w + \eta(2y_n - 2w \cdot x_n - 2b)x_n$$
$$b \leftarrow b + \eta(2y_n - 2w \cdot x_n - 2b)$$

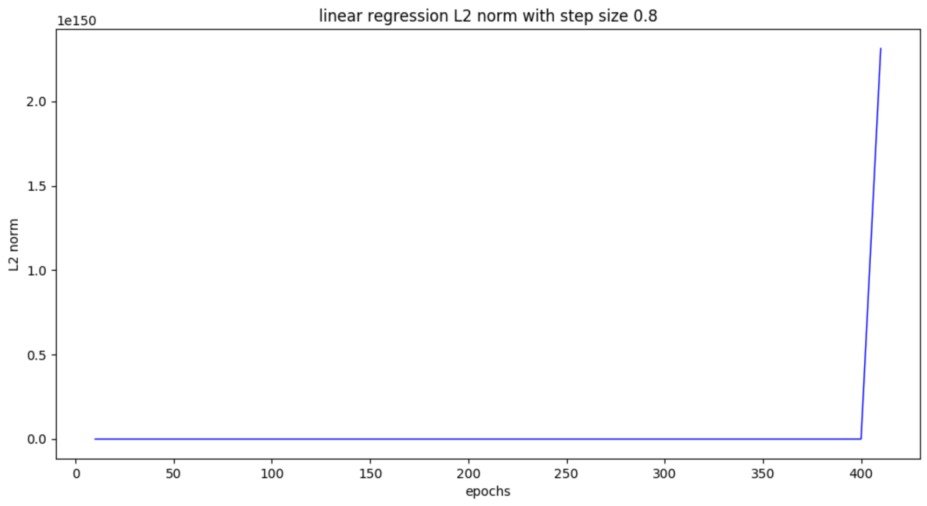
Logistic regression weight update:

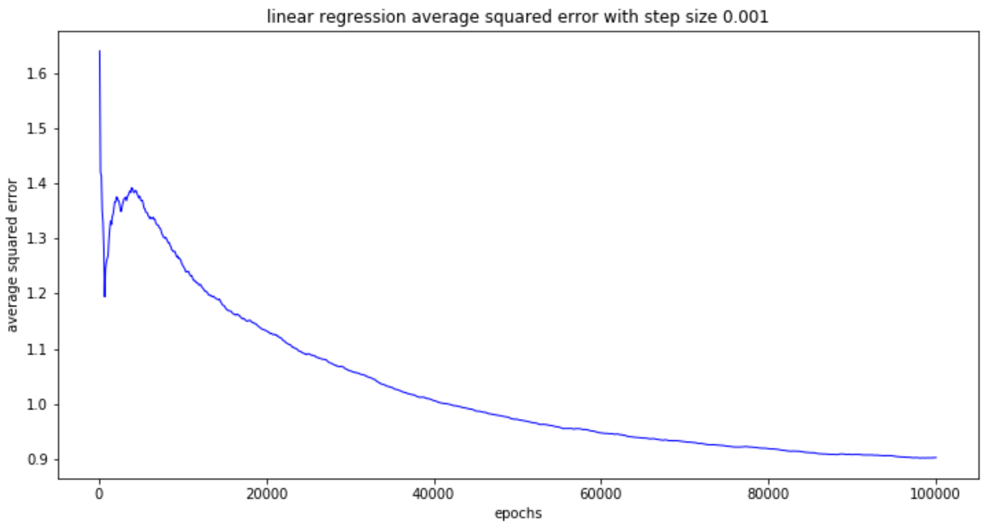
$$w \leftarrow w + \eta(\frac{1}{1 + exp(y_n \cdot (w \cdot x_n + b))}y_n x_n)$$
$$b \leftarrow b + \eta(\frac{1}{1 + exp(y_n \cdot (w \cdot x_n + b))}y_n)$$

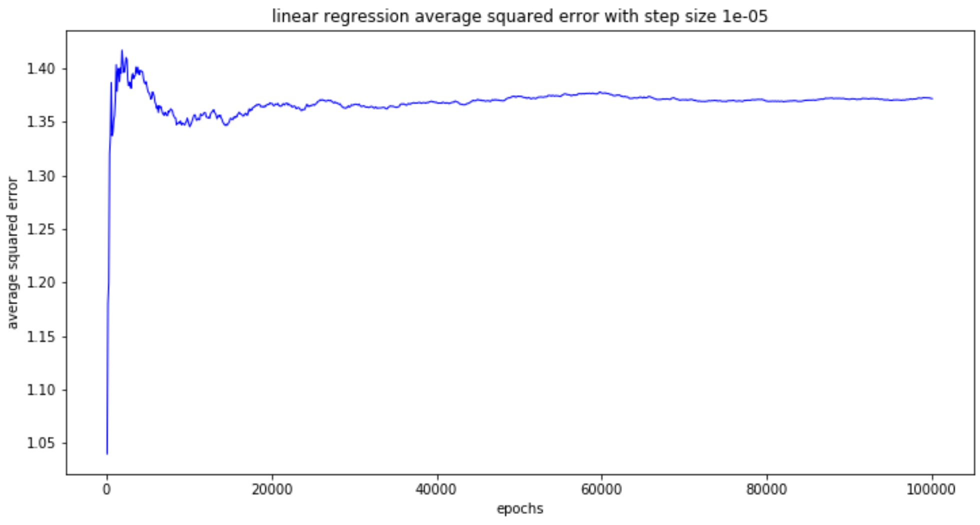
(b)

i. Average squared error

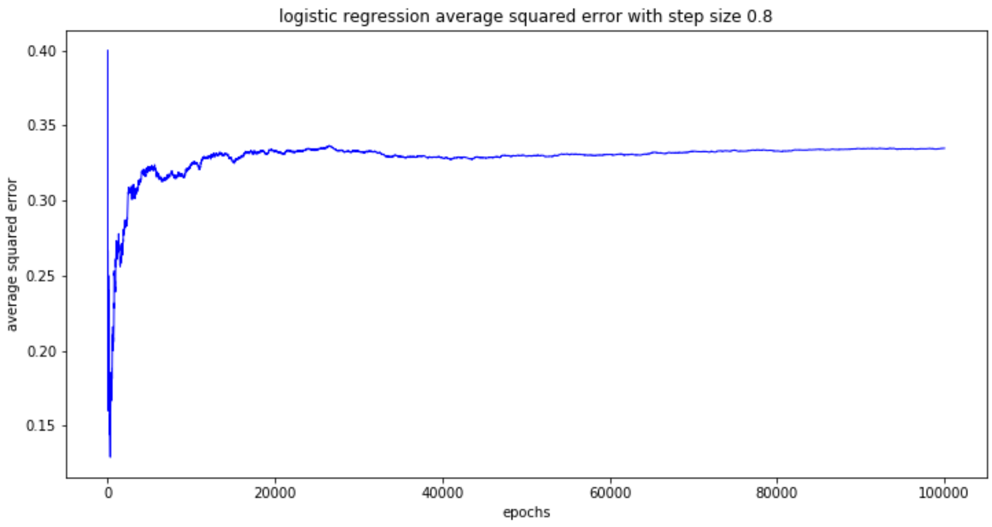
For linear regression

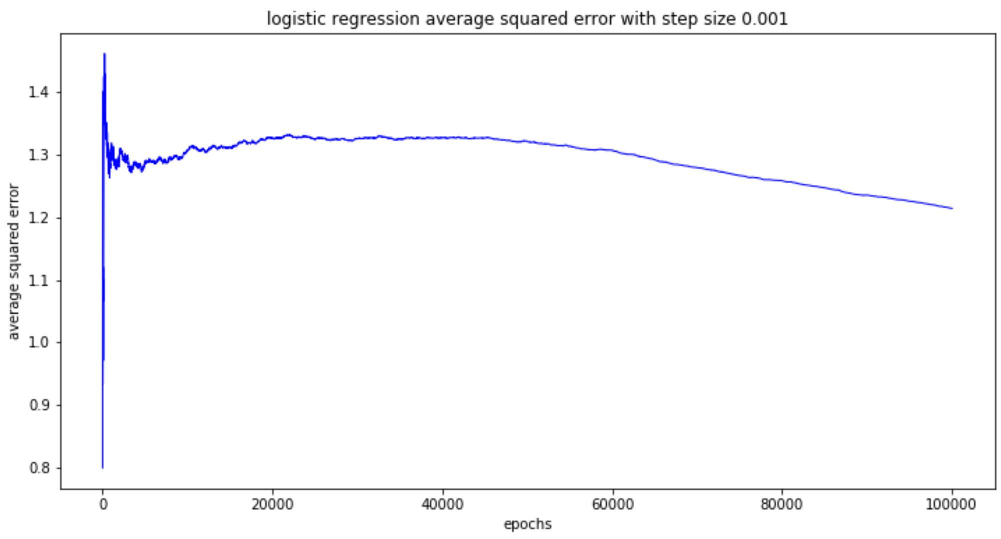


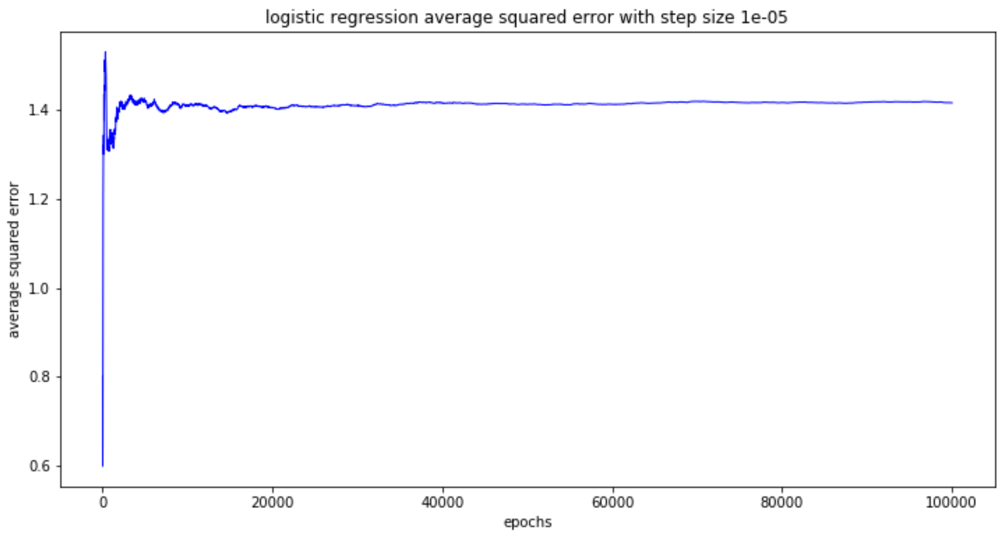




For logistic regression:

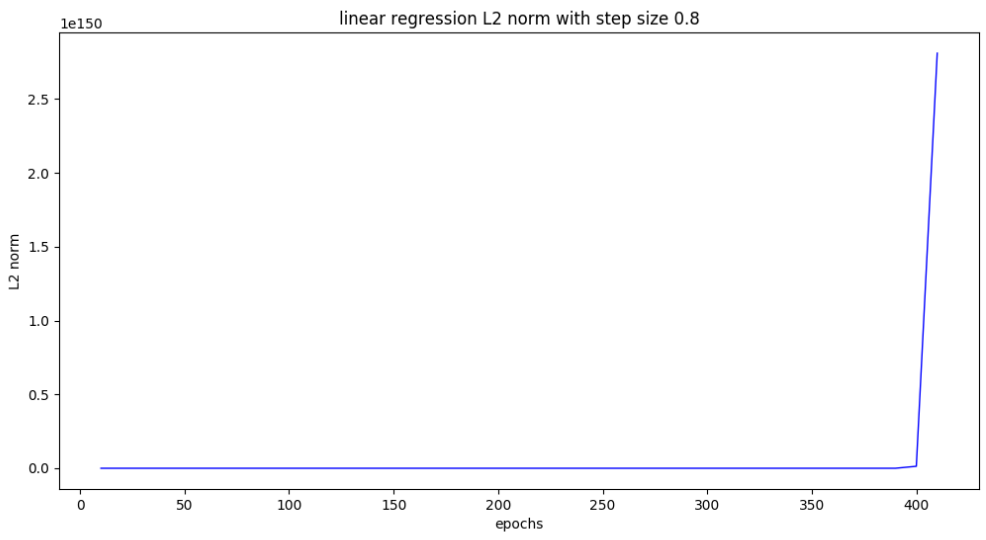


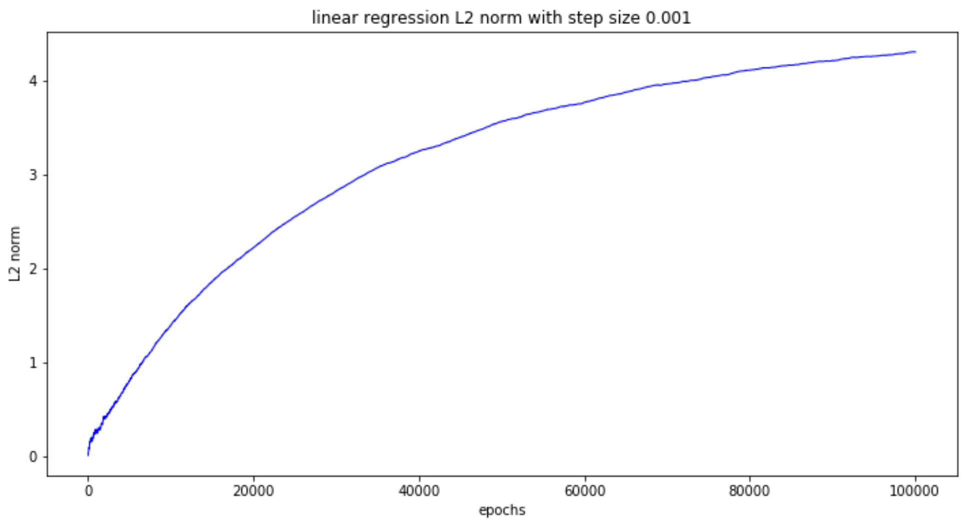


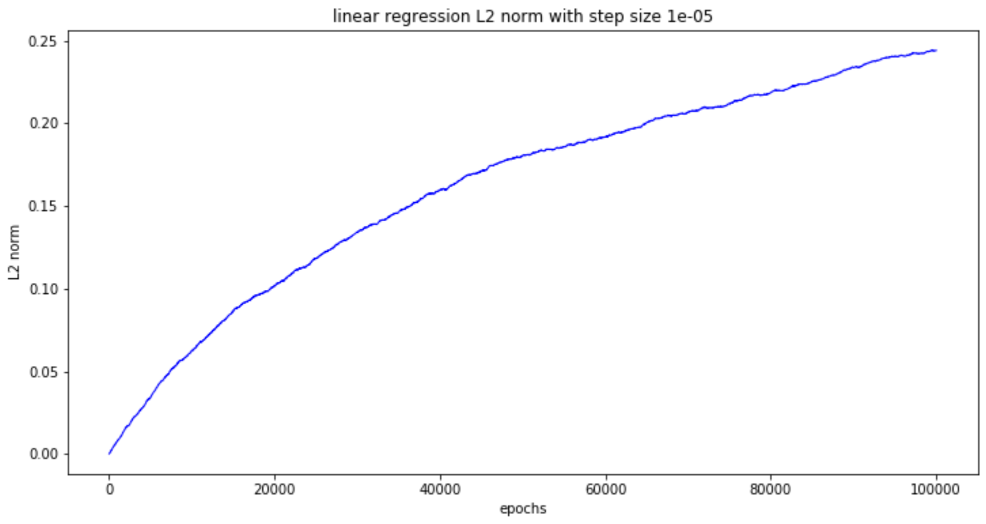


Note that for the linear regression with step size 0.8 there is a overflow issue, so I've stop the updating for weight before when the epoch is 500.

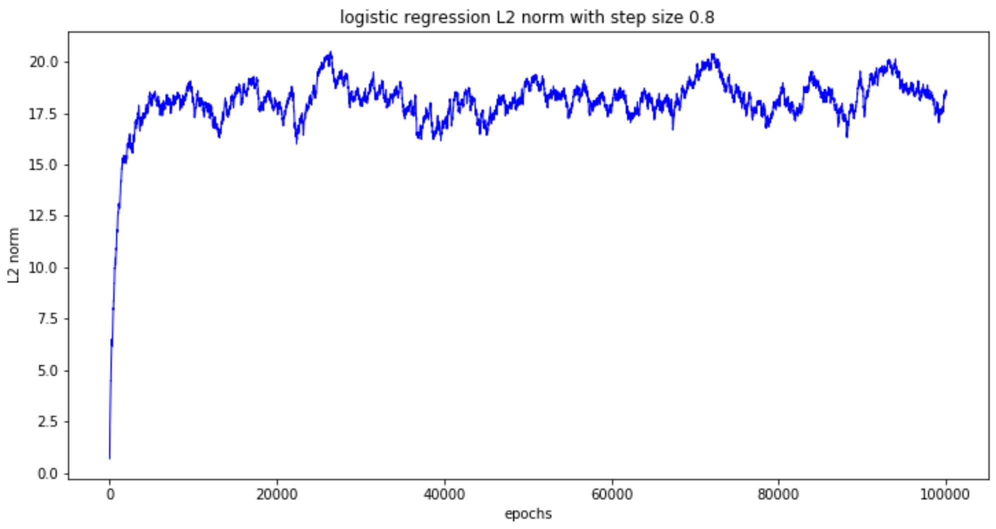
ii. L_2 norm
For linear regression

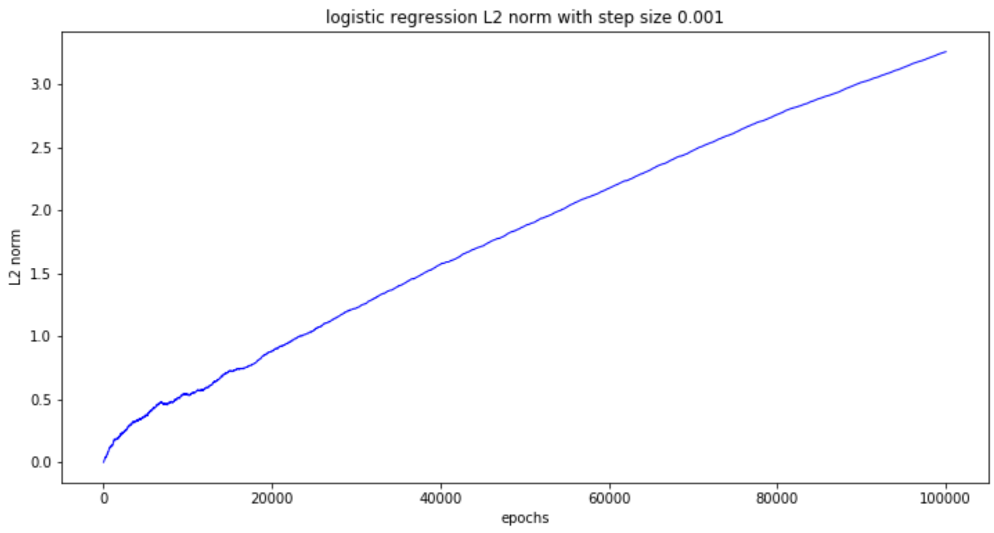


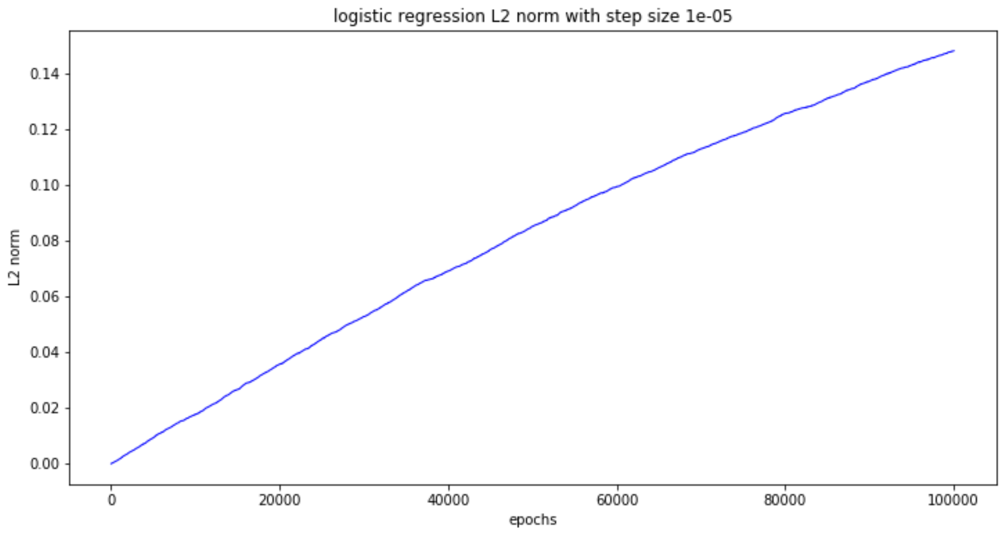




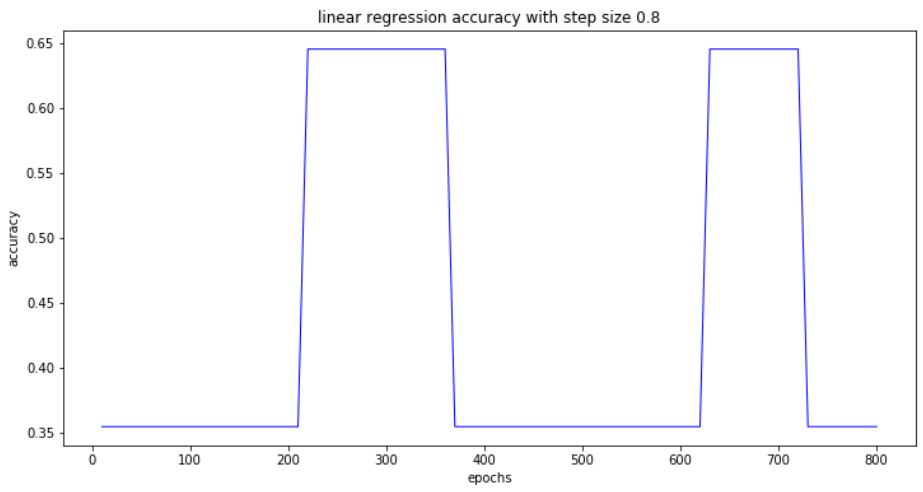
For logistic regression:

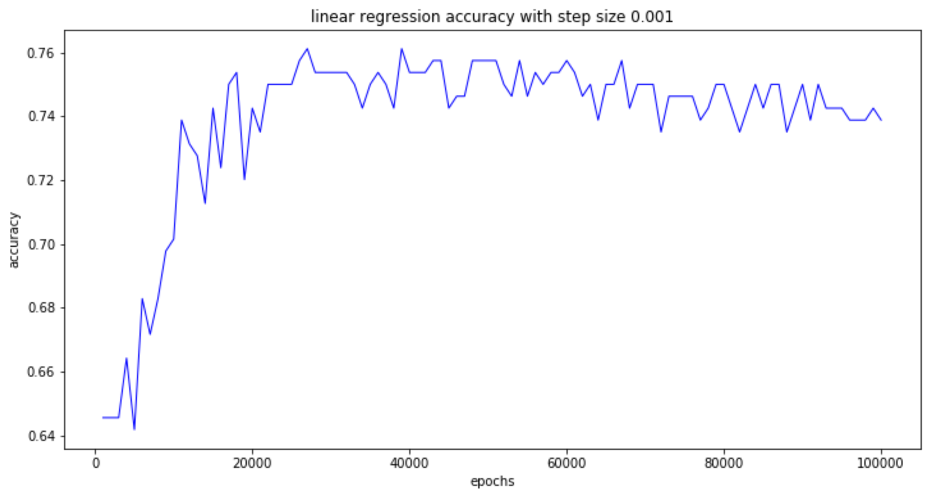


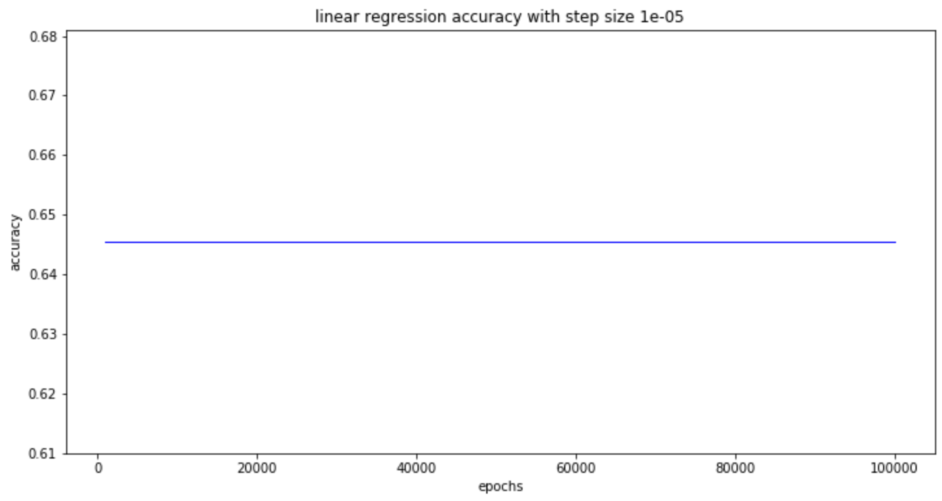




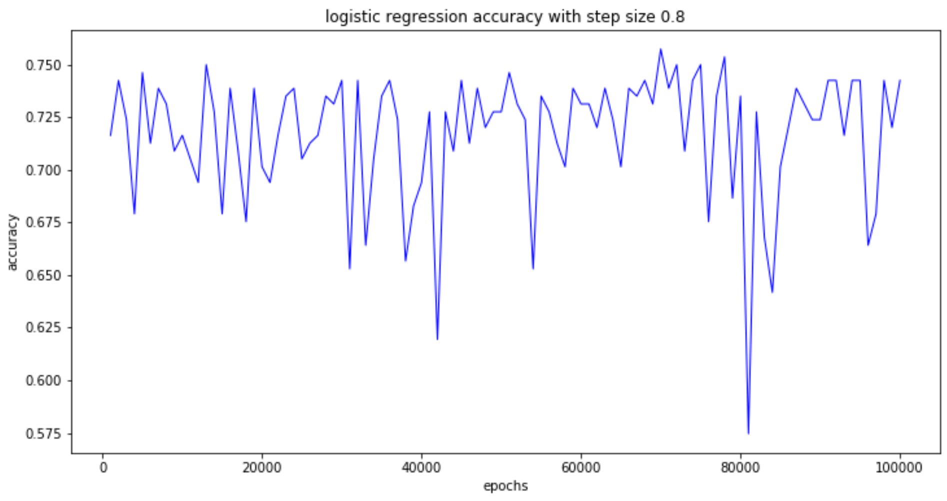
Note that for the linear regression, the L_2 norm is growing so fast that the python can't plot the value for that, so we only plot the first 500 epoch result.
iii. Test-set accuracy
For linear regression:

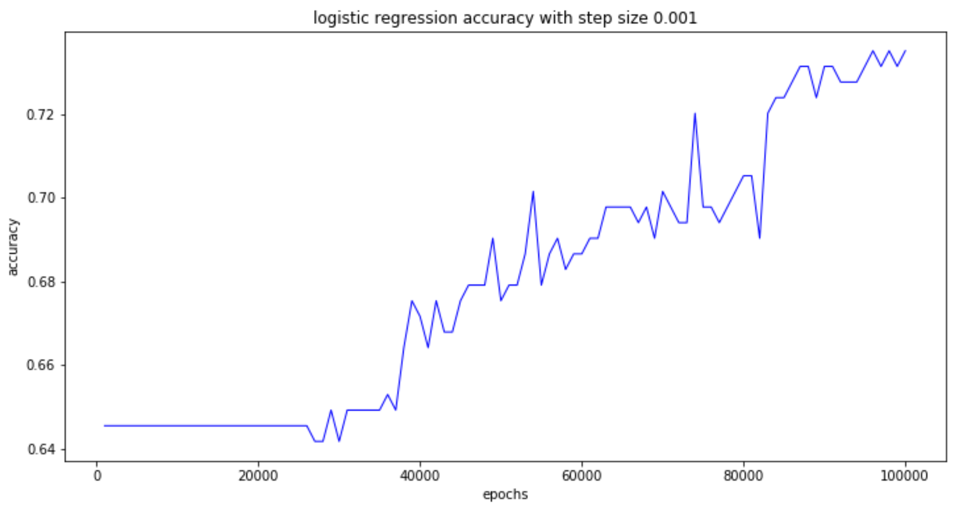


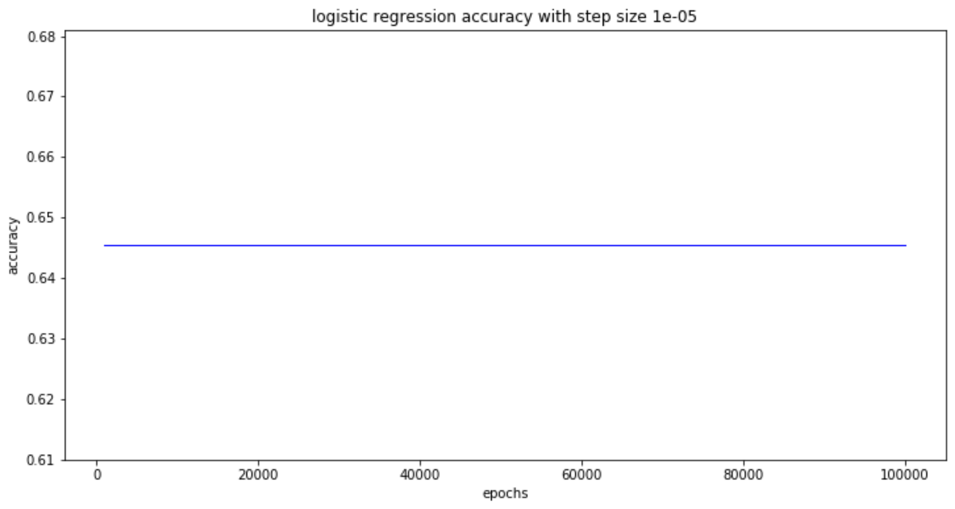




For logistic regression:







(c)
i.
In this problem, I will sample the develop data accuracy result for every 100 epochs with totally 100,000 epochs. The best model will select from the last 100 weights with best accuracy on develop data.
For linear regression, the best model is built with the step size as 0.001. The weights and bias showed as follow:

BMI	**2-hour serum insulin level**	**Plasma glucose concentration**	**Bias**
1.69	0.07	2.55	-2.65

For logistic regression, the best model is built with the step size 0.8. The weights and bias showed as follow:

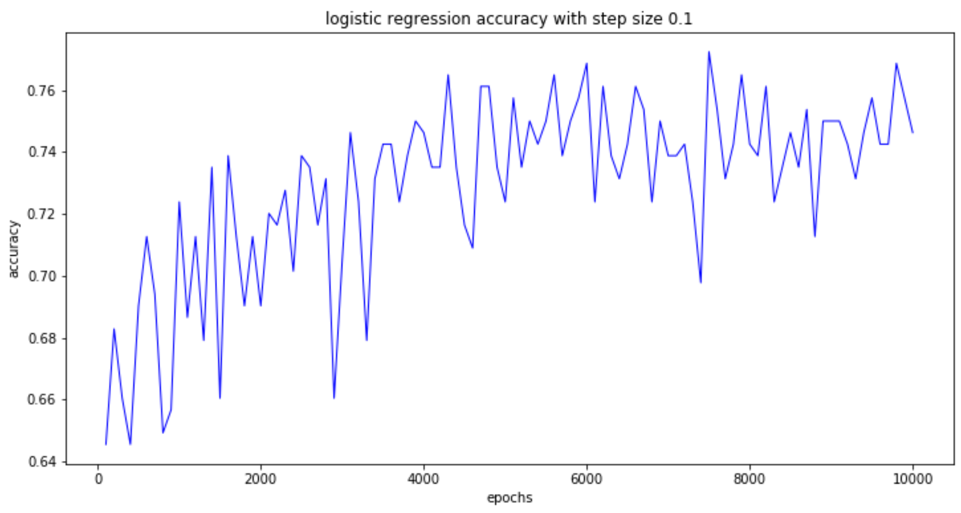
BMI	**2-hour serum insulin level**	**Plasma glucose concentration**	**Bias**
7.76	-2.25	11.17	-10.4

ii.
For linear regression, beside the bias, the weights of PGC has the biggest values, which means that the PGC is the most important feature when deciding the diabetes. The weight of 2-hour serum insulin level has the smallest value, which means it is the least meaningful feature when diagnose the diabetes.

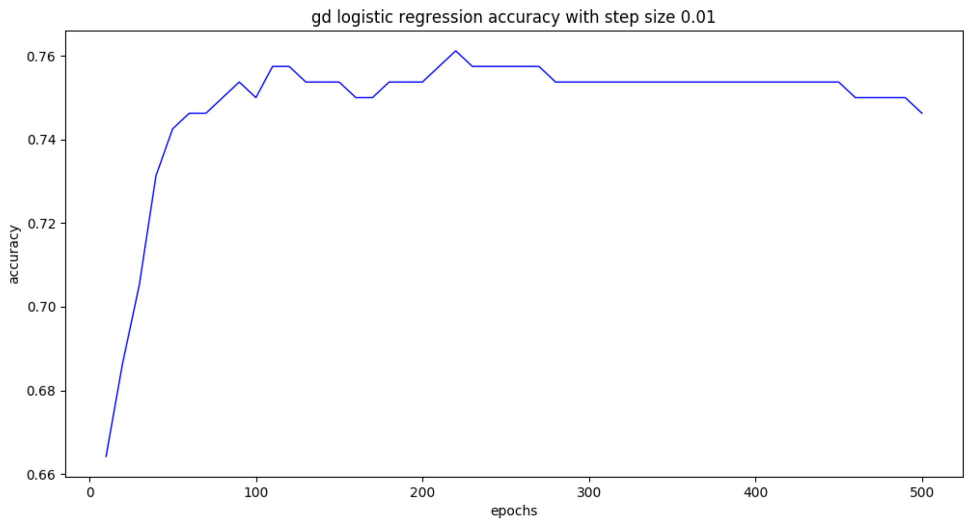
For the logistic regression, we got the same finding as linear one. The PGC has the biggest weight which means it is most important, and the 2-hour serum insulin level got the smallest weight which imply it is least important to decide the disease.

3.

i. I will focus on the logistic model.
ii. The final test data accuracy is 76%.
iii. The step size I used is 0.01. I've tried 0.1, 0.01, and 0.001 and the best result come from 0.01.
iv. It took about 200 epochs to get the result.
v. With the same step size 0.1, I've plot two plot for SGD and GD, the SGD showed as followed:



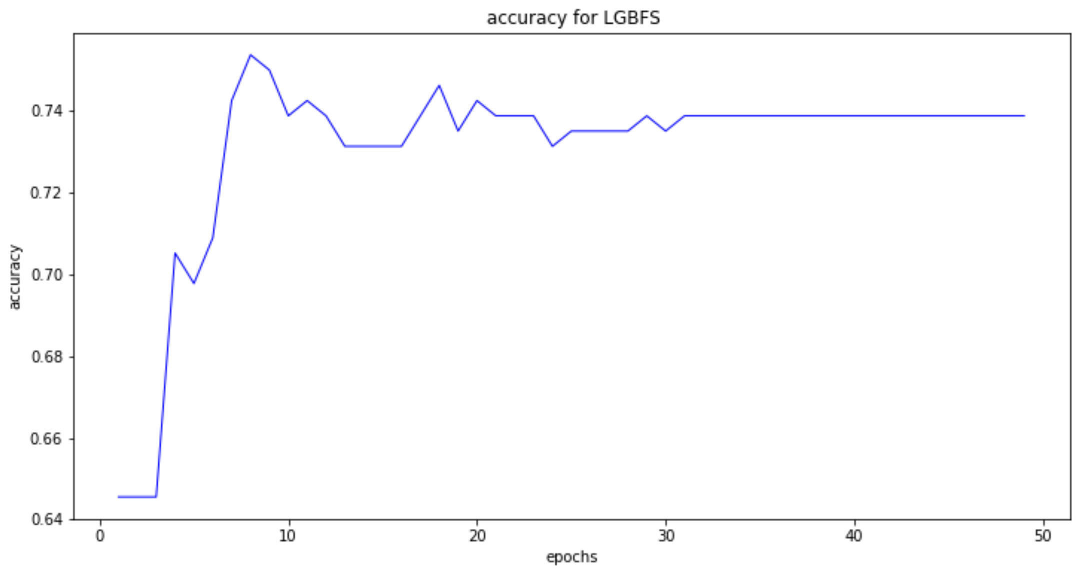
And the GD's showed as:



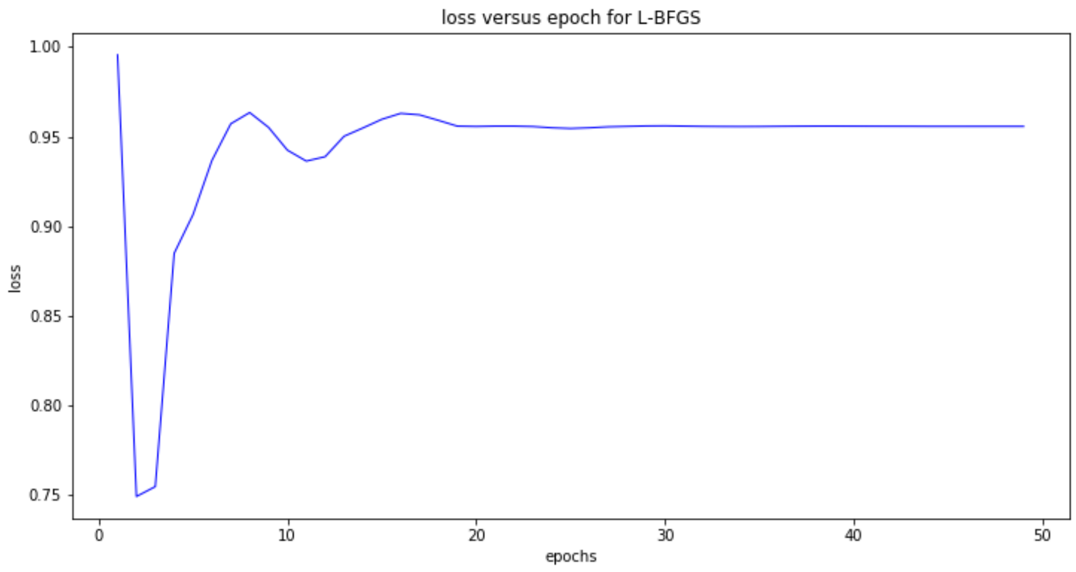
Both method reach around 76% accuracy. As we can see from the plots, there are three finding:
1). The accuracy of SGD joggling with a bigger magnitude and less stable, it reach the 76% at some point, but will immediately change then. The accuracy of the GD improve a lot at the beginning and don't change a lot.
2). The SGD reach the accuracy of 76% around 8000 epochs and GD reach it round 200. This means that the GD did better on minimizing the loss function each epoch, it converge faster.

3). The accuracy of SGD seems like will still growing while there occur overfitting problem with GD. This means GD need less epoch than SGD to build a final model. However, the runtime of theSGD may still less than GD.
4). Overfit. The GD showed the overfit issue at the late 400 and SGD didn't have the issue.

4.
i. I will focus on the linear model with L-BFGS.
ii. The final test data accuracy is 76%.
iii. There is no step size for this algorithm, but there are two other hyper parameters — ρ and σ.
iv. It took about 10 epochs to get the result
v. The L-BFGS' accuracy for test data showed as followed:



The loss value for the train data showed as:



As we can see from the plots, there are several findings can be concluded:
1). The L-BFGS converge super fast when compare with the GD and SGD. It reach the 76% accuracy within 10 epochs, which took SGD 000 and GD for 200.

2). Compare with the GD, the L-BFGS doesn’t joggling a lot, the accuracy and the loss are stable. Also, there is no overfilling issue with L-BFGS
3). Process speed. The L-BFGS compute for a very long time even with only 50 epoch compare with the GD and SGD.