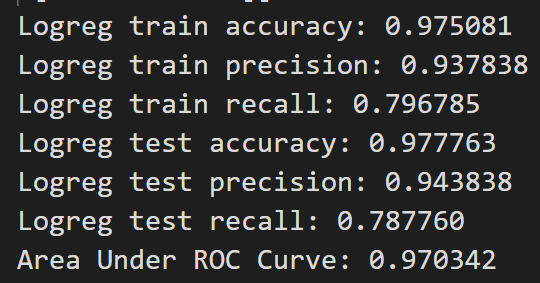
**Log\_Regression\_SGD**

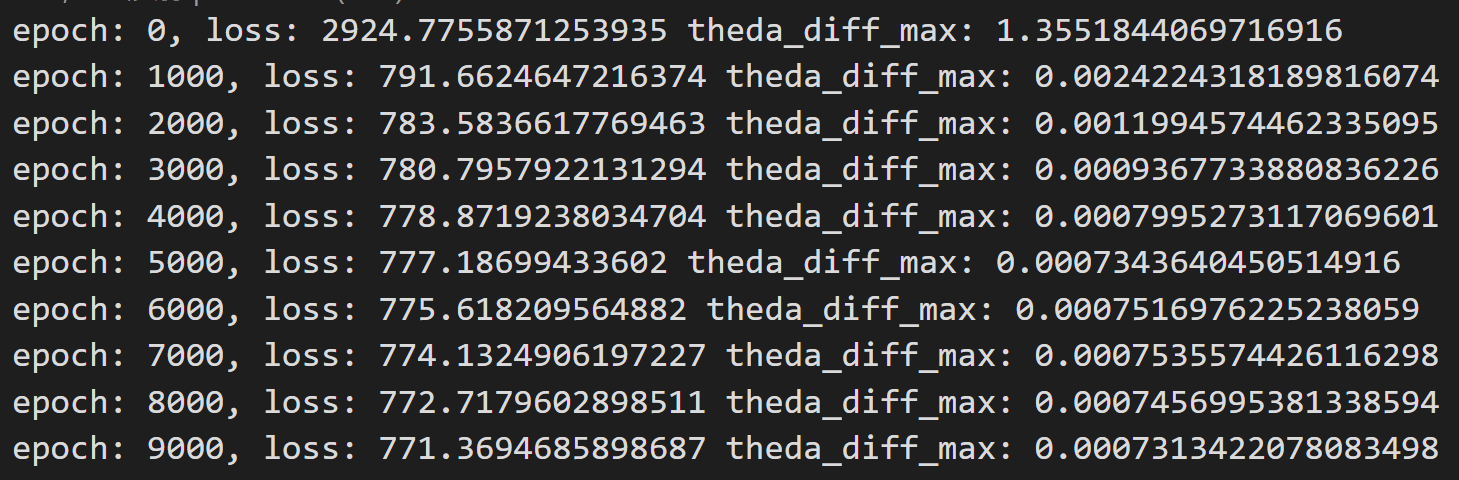
110505259 陳柏燊

**Accuracy and precision**



**The effect of different parameters**

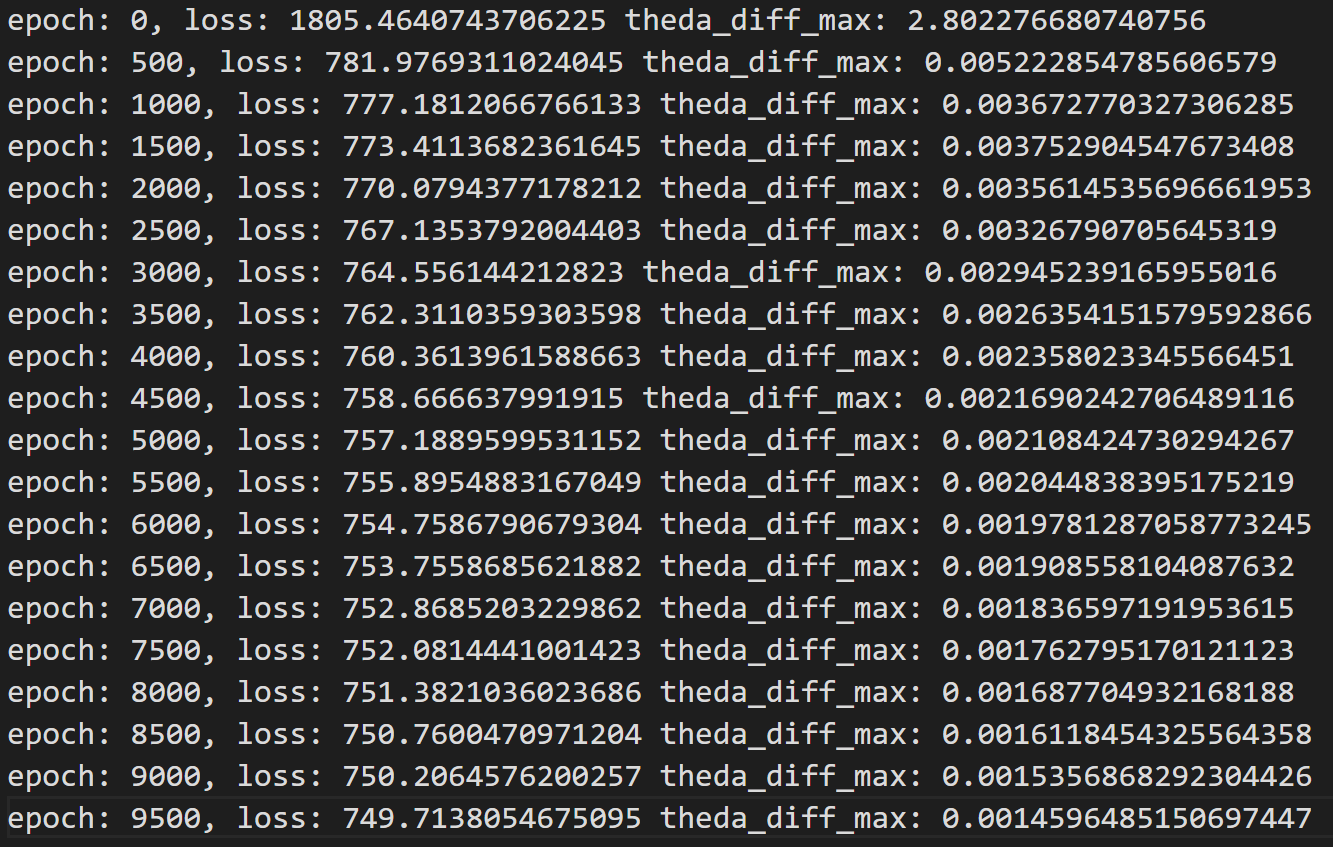
Alpha = 0.01, Max\_epochs = 10000, eps = 1e-4



We compare eps and epochs together.

We can find that eps and epochs both decide how many rounds should we update our Theda. We can see if we set eps to 8e-4, we will stop at near epoch 4000, and if we set epochs to 1000, we will stop at epochs 1000 stop. IF we stop earlier, we can see we will have higher los on train data, but if we encounter overfitting, stop earlier will help to solved it.

Alpha = 0.05, Max\_epochs = 10000, eps = 1e-4



We compare the 2 pictures above. There are same in Max\_epochs and eps but different in alpha. We first look at epoch 0. It shows that the loss with high alpha is lower than which with smaller alpha. I consider that alpha decide the size of steps. So if we move in a bigger step, we will got lower loss quicker than small steps.

**Discussion of the results**

We can find that the dataset is very incline to 0. We find that label with 1 with just 1639(9%) instances when the all data set is 17898 instances. So it more useful to use ROC curve that PR curve when PR curve would have lots of can’t reach area. I also find another excited thing. We find that since the value set of logReg is from 0 to 1, so the bias of the feature will always be zero, so I think it will cause it is useless to add 1 to all instances. Finally I don’t want to calculate AUC by myself so I all sklearn.mtrics.AUC to help me calculate that. I found that we should do some cleaning o our tpr and fpr to use the function. (sklearn AUC require for one fpr should only have one tpr and fpr should be increasing). That all my discussion.